AFAPL-TR-76-43 VOLUME VIII

12

FG

AIRCRAFT HYDRAULIC SYSTEMS DYNAMIC ANALYSIS

VOLUME VIII
TRANSIENT THERMAL ANALYSIS (HYTTHA)

COMPUTER PROGRAM TECHNICAL DESCRIPTION

MCDONNELL AIRCRAFT COMPANY
MCDONNELL DOUGLAS CORPORATION
ST. LOUIS, MISSOURI

February 1977

TECHNICAL REPORT AFAPL-TR-76-43, VOLUME VIII

AIR FORCE AERO PROPULSION LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

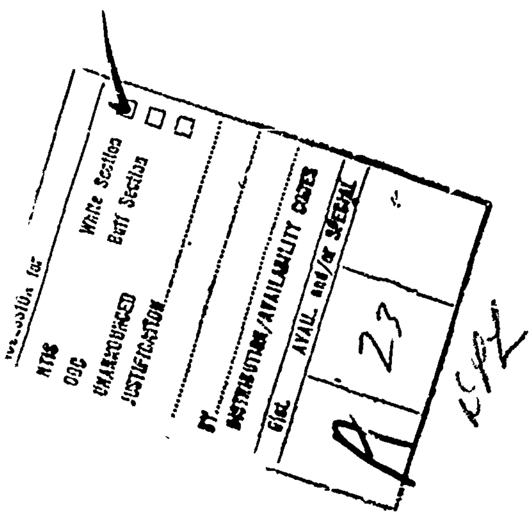
When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report was submitted by McDonnell Douglas Corporation, under contract F33615-74-C-2016.

The effort was sponsored by the Air Force Aero Propulsion Laboratory, Air Force Systems Command, Wright-Patterson A.F.B., Ohio, under Project No. 3145-30-18 with AFAPL/POP/, and was under the direction of Paul Lindquist and William Kinzig.

Neil Pierce and Gerry Amies of McDonnell Douglas Corporation were technically responsible for the work.

This report has been reviewed by the Information Office, (ASD/OIP) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.



Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAPL-TR-76-43 VOL. VIII	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) 6 AIRCRAFT HYDRAULIC SYSTEM DYNAMIC ANALYSIS VOLUME VIII. TRANSIENT THERMAL ANALYSIS (HYTTHA) COMPUTER PROGRAM TECHNICAL DESCRIPTION.		5. TYPE OF REPORT & PERIOD COVERED Interim Technical Report, 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Ray Levek Jeff Monroe		8. CONTRACT OR GRANT NUMBER(s) 15 F33615-74-C-2016
9. PERFORMING ORGANIZATION NAME AND ADDRESS McDonnell Douglas Corporation P. O. Box 516 St. Louis, Missouri 63166		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 3145-30-18 12 364p.
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Aero Propulsion Laboratory Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433		12. REPORT DATE 11 February 1977 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 16 3145 17 30		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

18 AFAPL 19 TR-76-43-Vol-8

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Program	Solenoid Valve
Hydraulic System	Reservoir
Transient Thermal Response	Piston Accumulator
Heat Exchanger	Actuator
Pressure Compensated Pump	User Manual

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Hydraulic Transient Thermal Analysis (HYTTHA) computer program has been developed to predict temperatures, and temperature gradients in hydraulic components and systems, due to changes in flow demands.

The steady state flows and pressures throughout the system are used by the thermal portion of the program to predict fluid temperatures, line wall temperatures, component temperatures, and other component variables.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403111

JB

The engineering input data to the program is normally available to a design engineer. Additional components, not covered here, may be added if necessary without much effort.

TABLE OF CONTENTS

PAGE

1.0 INTRODUCTION 1.0-1

2.0 TECHNICAL SUMMARY. 2.0-1

3.0 MAIN PROGRAM 3.0-1

3.1 THYTR 3.1-1

3.2 BLOCK DATA. 3.2-1

3.3 COMMON USAGE. 3.3-1

3.4 TFLUID. 3.4-1

4.0 STEADY STATE SUBROUTINES 4.0-1

4.1 TSSDATA 4.1-1

4.2 TCALC 4.2-1

4.3 TLEGCAL 4.3-1

5.0 LINE SUBROUTINES 5.1-1

5.1 TLINEA 5.1-1

6.0 COMPONENT SUBROUTINES. 6.0-1

6.1 TCOMPA. 6.1-1

6.11 TBRAN11 . . . . 6.11-1

6.21 TVALV21 . . . . 6.21-1

6.22 TVALV22 . . . . 6.22-1

6.31 TCVAL31 . . . . 6.31-1

6.41 TREST41 . . . . 6.41-1

6.51 TPUMP51 . . . . 6.51-1

6.61 TRSVR61 . . . . 6.61-1

6.62 TRSVR62 . . . . 6.62-1

6.69 THEX69. . . . . 6.69-1

TABLE OF CONTENTS (Continued)

PAGE

6.71 TACUM71 . . . . 6.71-1

6.81 TTILT81 . . . . 6.81-1

6.101 TACT101 . . . . 6.101-1

6.102 TACT102 . . . . 6.102-1

7.0 OUTPUT SUBROUTINES 7.0-1

7.1 TSTORE 7.1-1

7.2 TGRAPH 7.2-1

7.3 SCALED 7.3-1

8.0 UTILITY SUBROUTINES 8.0-1

8.1 INTERP 8.1-1

8.2 DISER1 . . . . 8.2-1

8.3 LAGRAN 8.3-1

8.4 SIMULT 8.4-1

8.5 TGAUSS 8.5-1

8.6 UFW 8.6-1

8.7 XLIMIT 8.7-1

8.8 FRIC 8.8-1

9.0 REFERENCES 9.0-1

LIST OF FIGURES

NUMBER	TITLE	PAGE
3.1-1	THYTR FLOW DIAGRAM.	3.1-2
4.2-1	TCALC GENERALIZED FLOW DIAGRAM.	4.2-2
4.2-2	TCALC EXAMPLE SYSTEM.	4.2-6
4.2-3	TCALC SUBROUTINE PHASE ONE OPERATION.	4.2-7
4.2-4	TCALC SUBROUTINE PHASE TWO OPERATION.	4.2-12
4.3-1	TLEGCAL ORGANIZATION.	4.3-2
5.1-1	LINE NODE REPRESENTATION.	5.1-1
5.1-2	LINE SEGMENT NODE REPRESENTATION.	5.1-2
5.1-3	LINE SEGMENT THERMAL MODEL.	5.1-5
6.1-1	COMPONENT SUBROUTINE ORGANIZATION	6.0-2
6.11-1	BRANCH CONFIGURATIONS	6.11-1
6.11-2	"CROSS" BRANCH AND LINE SEGMENT NODE REPRESENTATION	6.11-2
6.11-3	BRANCH THERMAL MODEL.	6.11-5
6.21-1	TYPE NO. 21 TWO-WAY VALVE	6.21-1
6.21-2	VALVE AND LINE NODE REPRESENTATION.	6.21-2
6.21-3	VALVE 21 THERMAL MODEL.	6.21-6
6.22-1	FOUR-WAY VALVE SCHEMATIC.	6.22-1
6.22-2	VALVE NODE REPRESENTATION	6.22-2
6.22-3	FLUID FLOW PATHS.	6.22-3
6.22-4	VALVE 22 THERMAL MODEL.	6.22-9
6.22-5	EFFECTIVE VALVE AREA CHARACTERISTICS.	6.22-11
6.31-1	TYPE NO. 31 CHECK VALVE	6.31-1
6.31-2	CHECK VALVE NODE REPRESENTATION	6.31-2
6.31-3	NODE REPRESENTATION FOR REVERSE FLOW.	6.31-5

LIST OF FIGURES (Continued)

NUMBER	TITLE	PAGE
6.31-4	CHECK VALVE THERMAL MODEL FOR FORWARD FLOW.	6.31-7
6.31-5	TCVAL31 STEADY STATE PRESSURE DROP CHARACTERISTICS.	6.31-8
6.41-1	TYPE NO. 41 ORIFICE RESTRICTOR.	6.41-1
6.41-2	RESTRICTOR AND CONNECTOR NODE REPRESENTATION.	6.41-2
6.41-3	RESTRICTOR THERMAL MODEL.	6.41-5
6.51-1	TYPE NO. 51 PRESSURE REGULATED VARIABLE DISPLACEMENT PUMP .	6.51-1
6.51-2	PRESSURE REGULATED VARIABLE DISPLACEMENT PUMP AND LINE NODE REPRESENTATION	6.51-2
6.51-3	PUMP THERMAL MODEL.	6.51-11
6.51-4	PUMP OUTLET FLOW VS OUTLET PRESSURE CURVE	6.51-12
6.61-1	TYPE NO. 61 CONSTANT PRESSURE RESERVOIR	6.61-1
6.61-2	RESERVOIR NODE REPRESENTATION FOR SAMPLE SYSTEM	6.61-2
6.62-1	TYPE NO. 62 BOOTSTRAP RESERVOIR	6.62-1
6.62-2	BOOTSTRAP RESERVOIR AND LINE NODE REPRESENTATION.	6.62-2
6.62-3	RESERVOIR THERMAL MODEL.	6.62-7
6.62-4	TRSVR62 STEADY STATE FLOW DIAGRAM	6.62-8
6.69-1	TYPE NO. 69 HEAT EXCHANGER.	6.69-1
6.69-2	HEAT EXCHANGER AND LINE NODE REPRESENTATION	6.69-2
6.69-3	HEAT EXCHANGER THERMAL MODEL.	6.69-7
6.71-1	TYPE NO. 71 FREE PISTON ACCUMULATOR	6.71-1
6.71-2	ACCUMULATOR AND CONNECTING LINE SEGMENT REPRESENTATION. . .	6.71-2
6.71-3	ACCUMULATOR THERMAL MODEL	6.71-8
6.81-1	TYPE NO. 81 F-4 TYPE IN-LINE FILTER	6.81-1
6.81-2	FILTER AND LINE SEGMENT NODE REPRESENTATION	6.81-2
6.81-3	FILTER THERMAL MODEL.	6.81-5

LIST OF FIGURES (Continued)

NUMBER	TITLE	PAGE
6.101-1	TYPE NO. 101 VALVE CONTROLLED ACTUATOR.	6.101-1
6.101-2	ACTUATOR, VALVE AND LINE SEGMENT NODE REPRESENTATION.	6.101-2
6.101-3	ACTUATOR WORKING SIMULATIONS	6.101-3
6.101-4	ACTUATOR THERMAL MODEL.	6.101-12
6.102-1	TYPE NO. 102 UTILITY ACTUATOR	6.102-1
6.102-2	ACTUATOR AND CONNECTOR NODE REPRESENTATION.	6.102-2
6.102-3	ACTUATOR THERMAL MODEL.	6.102-3

LIST OF TABLES

NUMBER	TITLE	PAGE
3.2-1	MATERIAL PROPERTY ARRAY.	3.2-3

LIST OF PAGES

Title Page
ii through ix
1.0-1 through 9.0-1

1.0 INTRODUCTION

The Hydraulic Transient Thermal Analysis (HYTTHA) computer program is intended for use by designers with an interest in the thermal effects on the performance of an aircraft hydraulic system.

An aircraft hydraulic system is basically a power source connected to several loads. The main power source is the pump, while the loads include components such as valves and actuators. The power is transmitted by hydraulic fluid in the lines which connect the components.

The system can also be considered as several thermal sources and sinks. The main thermal source is the pump while secondary sources include valves and restrictors. The sinks include the atmosphere and structure external to the sources. The HYTTHA program provides a tool for calculating the transient thermal response of the system when a thermal source, such as a valve, changes the flow demand in the system, thus changing system temperatures.

The program calculates flow rates, pressures, and temperatures throughout the system. Initially input data are used to calculate steady state pressures and flow rates as input for an initial temperature calculation for the first time increment. The steady state values are then used to calculate all line temperatures for the same time increment, and the line temperatures are used to calculate all component temperatures. The program continues to calculate steady state flow rates and pressures, and line and component temperatures for successive time increments.

To use the program, the designer inputs data describing lines, components, and system configurations. Since the simulated system is only as good as the data, care must be given to providing the best data possible.

In the steady state section of the program, the pressures and flow rates are balanced throughout the system (for the previous time step temperatures) and all state variables are calculated.

In the line section, the lines are divided into segments. The temperatures at the boundaries of each line are predicted from stored information and the temperatures of each successive line segment are calculated.

In the component section, the calculated line data is used to calculate the component fluid and wall temperatures.

After all calculations are completed, the output is printed and plotted. The designer selects the variables that are required as output tables or plots. The output is essentially a time history of the selected variables.

Since the program calculations advance in discrete time steps, results can be integrated into other simulations, if the cost of running can be tolerated.

This report is a technical description of the HYTTHA Program. Included are detailed listing of the main program and subroutines, and the theoretical basis and assumptions made in the calculations.

Volume VII of this report is a users manual which describes how the program can be used, the method of data input, and the interpretation of the output.

2.0 TECHNICAL SUMMARY

The HYTTHA program uses a building block approach which allows a designer to solve transient thermal problems by combining existing hydraulic line and component subroutines to thermally simulate hydraulic systems. This approach allows the user to add special component subroutines to the existing component subroutine library, as required.

For the analysis, the lines and components are represented by both wall and fluid nodes, equations are written for heat transfer to and from each node, and the equations are solved for successive time increments. The equations are defined in a backwards difference scheme and include modes of heat transfer such as conduction, convection, radiation, heat transfer due to mass transfer, and temperature rise due to a pressure drop. The line temperatures are calculated for one Δt and the results are used to calculate the component temperatures for the same Δt. The component results are then used to calculate new line temperatures and subsequently new component temperatures.

HYTTHA uses some basic assumptions and approximations throughout the program. The assumptions are:

1. The emissivity of the materials remain constant, (0.3).
2. The atmosphere and structure temperatures external to the lines and components are constant.
3. Each node is considered as a mass at one temperature.
4. The fluid exiting from a component is equal to the component fluid temperature calculated.
5. A pressure drop across an orifice results in a fluid temperature rise (DCAPT). A percentage (D(PERC)) of the heat goes directly into the fluid, and the remainder into the wall.
6. The interface conductance, between the lines and components, is infinite.

7. The pump heat rejection D(HTREJ) is constant.

The approximations are:

1. The radiative shape factor (SHAPF) is .96
2. A default value for the coefficient of heat transfer (UFWIL) to and from the external component wall to the atmosphere is 0.0069 WATTS/FT.^{2}-°F, which is that of still air.

The terms DCAPT, PERC, UFWIL, (convective heat transfer coefficient between the fluid and the wall), SHAPF, and HTREJ are explained below.

DCAPT - The temperature rise in a fluid due to a pressure drop across an orifice is a function of the fluid temperature at the orifice. The oil is essentially incompressible and energy extracted from the oil is assumed to be negligible. Because of incompressibility, the specific volume may be considered independent of pressure. A constant enthalpy process, which is insensitive to pressure variations is simulated. The temperature rise across an orifice is:

T = (1/density) (High Pressure - Low Pressure)/(CJ*Cp)

CJ = mechanical equivalent of heat

C_p = specific heat of the fluid

(DCAPT is equal to T)

D(PERC)- This term denotes how much of the heat, generated by a pressure drop, is added directly to the fluid. The remaining heat is added to the wall in contact with the fluid. Normally D(PERC) is equal to 1.0 which means that 100% of the heat is added directly to the fluid.

UFWIL - This term is a coefficient for heat transfer between a fluid and a wall. It is calculated by a separate subroutine, FUNCTION UFW.

SHAPF - The radiation shape factor is defined as the "fraction of diffusely distributed radiation leaving a surface Ai that reaches surface Aj" (Reference 9.1). Since components are completely enclosed by structure, the shape factor should be equal to 1.0, but this does not account for that part of the radiation from one component node which reaches another node of the same component. The value 0.96 is used for the shape factor of the components.

HTREJ - This term is heat rejection associated with the pump. This term includes the heat of compression of the fluid and heat due to friction in the moving parts in the pump. It is specified for many pumps, generally as a function of R.P.M. and volume flow rate. In the present pump model a constant value is input by the user. From this value, 32.3% of the heat is added to the exiting fluid, 25% is added to the pump walls, 18.7% is added to the piston mass, and the other 24% is added to the case fluid.

To use the program, the designer inputs:

1. Dimensions - such as lengths, areas, and volumes.
2. Material Properties - the user indicates type and the program uses tabulated values of Cp, ρ, and K at 100°F. The materials stored in the program are titanium, aluminum, steel, and teflon.
3. Initial Temperature - the initial temperatures of the lines and components are inputted.
4. Initial Flows - the initial flow quesses in the lines and components are inputted.
5. Heat Transfer Coefficients - Several may be needed. These are special for each subroutine.

The output is either a table or graph of the time history of a component or line wall and fluid temperatures.

It should be noted that the current maximum number of lines (MNLINE), components (MNEL), legs (MNLEG), nodes (MNODE), plots (MNPLOT), and line points (MNPTS) that can be input are limited in BLOCK DATA. Hence BLOCK DATA must be changed if any of these maximum values are exceeded when inputting a system. These are defined in Section 3.2 in the manual.

3.0 MAIN PROGRAM

The main or executive program section of HYTTHA is named THYTR. THYTR controls the flow of the program, and keeps track of the counters for the time variables. The block data and fluid subroutines are also included in this section.

The HYTTHA program is very similar in organization to the HYTRAN (Hydraulic Transient Analysis) computer program. Many of the subroutines in HYTTHA have HYTRAN counterparts and function in the same way.

Some cost savings can be made by the use of overlays or segments. The implementation of these devices is left to the individual user.

3.1 THYTR PROGRAM

THYTR is the main or executive program of HYTTHA. The program flow is directed from THYTR. The main program card is set up to read from a file called DATA. This may be changed to suit the user's own data inputting scheme. Extensive use is made of common and equivalences in the program, so care is required in modifying variables that are contained therein.

A flow diagram of the main program is shown in Figure 3.1-1. In the first section of THYTR, the general system data is input. This data is printed out and a call is made to the fluid subroutine. In TFLUID the values of bulk modulus, viscosity and density are tabulated for the inputted fluid type. Next the TLINEA subroutine is called to read the line data cards, and to initialize the appropriate variables. Likewise TCOMPA reads in all the component data cards and calls the component subroutines. The TSSDATA subroutine is then called to input all the steady state leg and node information.

In the next section IENTR is set to zero and a call is made to TCALC to calculate the steady state flows and pressures throughout the system. The TLINEA subroutine is called to initialize the line temperature based on the steady state flows and pressures. TSTORE reads all the output data requirements and stores the user selected output data just calculated at time = 0.0.

The third section advances the time step by DELT. TLINEA and TCOMPA are called to do the thermal transient calculations and the variables to be plotted are saved by TSTORE. If the sum of the time steps are less than the final time TSCALC is called to compute the latest steady state flows and pressures and this computation section is repeated. The program stops when the time exceeds the final time specified in the input.

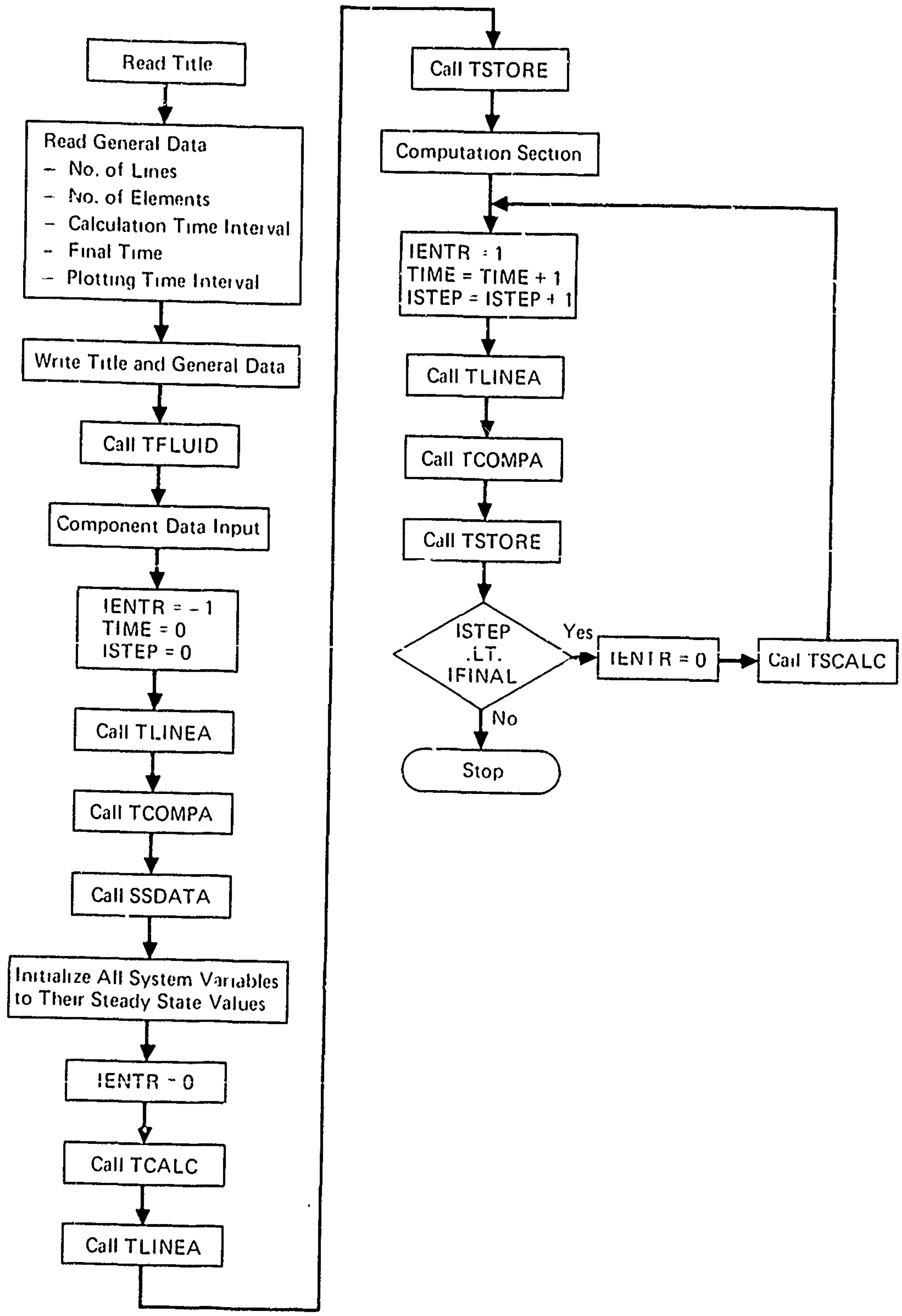


FIGURE 3.1-1
THYTR FLOW DIAGRAM

GP77-0065-1

3.1.1 Math Model. Not applicable.

3.1.2 Assumptions.

The basic assumptions in THYTR are as follows:

- Flow is one-dimensional, that is, the fluid properties are constant across any transverse cross section of the pipe.
- Pipes have circular cross sections.
- Stresses in pipes are always below the elastic limit.
- Pipe geometry is such that the "thin wall" case is valid.
- Pipe and liquid are perfectly elastic (all energy dissipation is due to shearing stresses at the walls).

3.1.3 Computation Methods. Not applicable.

3.1.4 Approximations.

THYTR approximations are those inherent in numerical analysis. They are kept small enough by error control to be of no practical influence.

3.1.5 Limitations.

THYTR currently has the following constraints:

- Temperature range ... -65° to 300°F
- Pressure range ... 0 psia to 5000 psia
- Maximum number of components ... 99
- Maximum number of lines ... 150
- Maximum number of legs ... 70
- Maximum number of nodes ... 55
- Maximum number of plots ... 60

3.1.6 Variable Names

Variable	Description	Dimension
I	Counter	--
IFINAL	Number of transient iterations	--
PRESS	Working Pressure	PSI
TEMP	Working Temperature	°F
Y	Dummy Variable	--

3.1.7 Main Program Listing

```
      PROGRAM THYTR(DATA,OUTPUT,DATA1,TAPE5=DATA,TAPE6=OUTPUT,
     + TAPE7=DATA1)
C *** REVISED AUGUST 5, 1976 ***
      COMMON DUM(3500),VSTORE(6000)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /LIMIT/MNLINE,MNEL,MNLEG,MNNODE,MNPLOT,MNLPTS,MDS
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      COMMON /PLOT/TITLE(20),PLTDEL,NPTS,IPOINT,ISTEP,TFINAL,NLPLT(61,3)
     + ,NABSQ,NTOPL,NTOLPL
C***
      READ(5,470)(TITLE(I),I=1,20)
      WRITE(6,480) TITLE
      ISTEP=0
      PI=3.1416
      TIME=0.0
C***
C     THIS READ STATEMENT INPUTS THE FOLLOWING DATA
C     NLINE =NUMBER OF LINES
C     NEL =NUMBER OF COMPONENTS
C     DELT  =DELTA TIME BETWEEN CALCULATIONS   SEC
C     TFINAL=FINAL TIME    SEC
C     PLTDEL=DELTA TIME BETWEEN PLOT POINTS   SEC
      READ(5,133)NLINE,NEL,DELT,TFINAL,PLTDEL
  133 FORMAT(2I5,3E10.0)
      WRITE(6,485) TFINAL,DELT,PLTDEL
      IF(DELT.EQ.0) GO TO 251
      NPTS=1.01 + TFINAL/PLTDEL
      IPOINT=0.5+PLTDEL/DELT
      IFINAL=0.5+TFINAL/DELT
      Y=TFLUID(TEMP,PRESS)
C
      IENTR=-1
C
      CALL TLINEA
C
      INEL=0
      CALL TCOMPA
C
      CALL TSSDATA
C
C *** THIS SECTION CALLS TLINEA AND TCOMPA TO INITIALIZE ALL THE
C     SYSTEM VARIABLES TO THEIR STEADY STATE VALUES
C
      IENTR=0
      INEL=0
C
      CALL TCALC
C
```

3.1.7 (Continued)

```
      CALL TLINEA
C
      CALL TSTORE
C
C     THERMAL TRANSIENT CALCULATION SECTION
C
 150  IENTR=0
      CALL TSCALC
C
      TIME=TIME+DELT
      ISTEP=ISTEP+1
C
      IENTR=1
C
      CALL TLINEA
C
C     DO ELEMENT CALCULATIONS
C
      INEL=0
      CALL  TCOMPA
C
      CALL TSTORE
C
      IF (ISTEP.LT.IFINAL) GO TO 150
C
      STOP
  251 CONTINUE
      STOP 3100
  470 FORMAT(20A4)
  480 FORMAT(1H1,25X,20A4,//)
 485  FORMAT(20X,52H THE THERMAL TRANSIENT RESPONSE IS FROM T=0.0 TO T=
  ,
     1 F7.3,35H SECONDS AT TIME INTERVALS OF DELT=    ,
     2 F7.5,//,30X, 48HWITH OUTPUT POINTS PLOTTED AT INTERVALS OF    ,
     3 F7.5,9H  SECONDS   , //)
      END
```

3.2 BLOCK DATA

Block data is used to initialize values in COMMON/LIMIT/and COMMON/COMD/. The maximum number of various input values in COMMON/LIMIT/ are established using the following data initialization statement

```
      DATA MNLINE,MNEL,MNLEG,MNNODE,MNPLOT,MNLPTS,MDS
     + /150,99,70,55,60,1500,4500/
```

Maximum and minimum values for each individual component are initialized in COMMON/COMPD/ as follows:

```
      DATA LT/100*0/
      DATA L11/11,4,0,4,0,0,4,2,0,0/
      DATA L1220/90*0/
      DATA L21/32,4,0,5,0,3,2,2,1,0/
      DATA L22/48,20,0,7,0,4,4,2,0,0/
      DATA L23/24,2,0,5,0,3,2,2,1,0/
      DATA L2430/8,12,0,12,0,0,8,8,0,0,60*0/
      DATA L31/24,12,0,3,0,1,2,2,1,0/
      DATA L32/6,5,0,3,0,0,3,3,1,0/
      DATA L3340/80*0/
      DATA L41/20,20,0,2,0,2,2,2,1,0/
      DATA L4250/90*0/
      DATA L51/36,30,0,4,0,5,3,3,1,0/
      DATA L52/10*0/
      DATA L5360/80*0/
      DATA L61/10,0,0,12,0,1,10,1,0,0/
      DATA L62/24,18,0,9,0,2,5,2,0,0/
      DATA L6368/60*0/
      DATA L69/24,6,0,4,0,3,2,2,1,0/
      DATA L70/10*0/
      DATA L71/24,18,0,10,0,1,2,1,0,0/
      DATA L7280/90*0/
      DATA L81/18,4,0,2,0,1,2,2,1,0/
      DATA L82/24,21,0,8,0,3,6,6,1,0/
      DATA L8390/80*0/
      DATA L91/0,3003,0,3.0,0,1,1,0,0/
      DATA L92100/90*0/
      DATA L101/56,27,0,7,0,2,2,2,1,0/
      DATA L102/32,21,0,6,0,2,2,2,1,0/
      DATA LEND/480*0/
      DATA PROP/140.4,144.,165.6,230.4,230.4,242.6,229.3,121.3
     1 ,118.8,115.2,115.2,115.2,295.34,.164,.16,.164,.101,.093,.098,.
     2 101,.286,.286,.282
     3 ,.28,.287,.0775,.0174,.105,.093,.226,.197,.210,.21,.35,.17,.
     4 23,.25,.21,.00354/
```

In BLOCK DATA the material properties of thirteen materials are stored in the PROP array in COMMON/FLUID/. The first column contains the specific heat of the material. Column two gives the material density and the conductivity is in column three. All the material properties are input for a temperature of 100°F. Since these properties did not vary greatly over the operating temperature range of the program (-65°F to 300°F), these values are not temperature compensated in the HYTTHA program. See Table 3.2-1 for a list of the material properties.

TABLE 3.2-1

MATERIAL PROPERTY ARRAY
PROP(X,Y)

X	MATERIAL TYPE	SPECIFIC HEAT (X,1)	DENSITY (X,2)	CONDUCTIVITY (X,3)
1	Titanium 6AL-25N-4ZR-2MO	140.4	.164	.1074
2	Titanium 6AL-4V	144	.16	.105
3	Titanium 6AL-6V-25N	165.6	.164	.093
4	Aluminum 2014	230.4	.101	.226
5	Aluminum 2024-T6	230.4	.098	.197
6	Aluminum 6061-T6	242.6	.098	.219
7	Aluminum 7075-T6	229.3	.101	.21
8	Steel 4130	121.3	.286	.35
9	Steel 301	118.8	.286	.17
10	Steel 304	115.2	.282	.23
11	Steel 17-4 PH	115.2	.28	.25
12	Steel A286	115.2	.287	.21
13	Teflon	295.34	.0775	.00354

3.3 COMMON USAGE

One blank and nine labeled common statements are used in the HYTTHA program. Their purpose is to share storage and pass arguments between the various subroutines.

1. Blank column is used to store output variables and pass steady state information.

```
COMMON DUM(3500),VSTORE(6000)
```

2. Common TRANS contains all the temperature, pressure and flow information to be used by the lines and components. Also the current time, calculation interval, PI, number of lines and number of elements are stored in this labeled common.

```
 COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
+ ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
```

3. Common LIMIT provides the maximum limits on the number of components, lines, nodes, legs, plots, line points and D variables.

```
COMMON /LIMIT/MNLINE,MNEL,MNLEG,MNNODL,MNPLOT,MNLPTS,MDS
```

4. Common LINE is used pass arguments concerning such line parameters as inside diameter and temperature of each fluid and wall segment.

```
COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
+ NLSEG(150)
```

5. Common COMP passes information on component types, numbers of active connections, current component numbers and leg number.

```
COMMON /COMP/LTYPL(99),NC(99),KTEMP(99),IND,IENTR,INEL
```

6. Common STEADY contains information used in the steady state portion of all the component and steady state subroutines.

```
COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
+ QA,QS,QI,PUP,PDOWN,NNODE,NLEG,NCPN,TERM,
+  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NELEM(90),
+ILEGAD(90),ILEG(1000)
```

7. Common ICC is used by the TCALC and TGAUSS subroutines in the steady state solution process to pass matrix information.

```
COMMON /ICC/ICOL(55,20),JCENT(55),JRENT(55)
```

8. Common FLUID contains atmospheric pressure, conductivity and specific heat of the fluid, the system default fluid temperature and physical properties of various materials.

```
COMMON /FLUID/APPRES,CF,CPFN,FTEMP,PROP(13,3)
```

9. Common PLOT is used to pass variables for the plotting subroutines.

```
COMMON /PLOT/TITLE(20),PLTDEL,NPTS,IPOINT,ISTEP,TFINAL,NLPLT(61,3)
+ ,NABSQ,NTOPL,NTOLPL
```

10. Common COMPD is used to pass variables used in component calculations.

```
 COMMON /COMPD/ D(4500),LP(100),L11(10),L1220(90),L21(10),L22(10),
+ L23(10),L2430(70),L31(10),L32(10),L3340(80),L41(10),L4250(90),
+ L51(10),L52(10),L5360(80),
+ L61(10),L62(10),L6368(60),L69(10),L70(10),L71(10),
+ L7280(90),L81(10),L82(10),L8390(80),L91(10),L92100(90),
+ L101(10),L102(10),LEND(480),LE(99,4)
```

The maximum input limits of the program are set in BLOCK DATA. In order to decrease any of the limits, the initialized data statement in LIMIT must be changed.

Note: The maximum number of lines that can be input is equal to the dimension of the P array divided by 2.

Array initialization for the components used in BLKDTA are as follows:

Array Location	Description
1	Number of real data points, D( )
2	Number of temporary variables, DT( )
3	Number of double precision variables, DD( )
4	Number of integer variables, L( )
5	Not used
6	Minimum number of data cards
7	Maximum number of connections
8	Minimum number of connections
9	Not used
10	Not used

3.3.1 Variable Names

Variable	Description	Dimension
ACF( )	Array of cross sectional areas of the fluid	IN^2
ACW( )	Array of cross sectional tube areas	IN^2
ATPRES	Atmospheric pressure	PSI
C ( )	Array of wall conductivity	WATTS/IN-°F
CF	Conductivity of the fluid	WATTS/IN-°F
CPFN	Specific heat of the fluid	WATTS-SEC/LB-°F
D( )	Component real data array	--
DELT	Program time step	SEC
DXF( )	Array of distances from fluid node to interface	IN.
FTEMP	Fluid temperature	°F
ICOL( )	Computational array indicating rows and column locations in a square matrix	--
ICON	Component connection number	--
IENTR	Subroutine entry point indicator	---
ILEG( )	Array containing component and line numbers identifying steady state legs	--
ILEGAD( )	Address of the start of each leg in the ILEG( ) array	--
IND	Number assigned to component by user	--
INEL	Current leg number in steady state computation	--
INV	Dummy variable	--
INX	Current element number in leg	--
INZ	Number of elements in a leg	--
IPOINT	Counter for number of points stored	--
ISTEP	Sum of time steps up to current time	--

Variable	Description	Dimension
JCENT( )	Computational Array indicating number of filled columns in a square matrix	--
JRENT( )	Computational array indicating number of filled rows in a square matrix	--
KTEMP( )	Dummy variable array	--
L( )	Component integer data array	--
LE(N)	Address of real data for component N	--
LEGN	Dummy variable	--
LEND( )	Dummy array	--
LSTART(N)	Address of first segment of line N	--
LT( )	Dummy array	--
LTYPE(N)	Component N type number	--
MDS	Maximum D( ) array size	--
MNEL	Maximum number of components	--
MNLEG	Maximum number of legs	--
MNLINE	Maximum number of lines	--
MNLPTS	Maximum number of line points	--
MNNODE	Maximum number of nodes	--
MNPLOT	Maximum number of plots	--
NABSQ	0 = normal plots 1 = plots magnitude	--
NC(N,I)	Line number attached to connection I of component N, and temporary storage area	--
NCPN	Number of constant pressure nodes	--
NDWN	Array of downstream node numbers	--
NEL	Number of components input	--

Variables	Description	Dimension
NELEM( )	Array of the number of legs and components in a leg	--
NLEG	Number of legs	--
NLINE	Number of lines input	--
NLPLT(,1)	Address of variable in P,Q,TC,TW or TF array	--
NLPLT(,2)	Line number	--
NLPLT(,3)	Coded input (1=P, 2 =Q, 3=TC, 4=TW, 5=TF)	--
NLSEG(N)	Number of segments in line N	--
NNODE	Number of steady state nodes	--
NPTS	Number of plot points	--
NTOLPL	Number of line plots	--
NTOPL	Total number of plots	--
NUP	Array of upstream node numbers	--
P( )	Array of line end pressures	PSI
PARM(,1)	Line length	IN.
PARM(,2)	Inside line diameter	IN.
PARM(,3)	Line equivalent length	IN.
PARM(,4)	Transition flow when multiplied by viscosity	CIS
PDLEG( )	Array of external pressure drops	PSI
PDOWN	Downstream leg pressure	PSI
PEX( )	Dummy array used in steady state section for external pressure calculations	--
PI	Constant 3.1416	--
PLTDEL	Plot time interval	SEC
PN( )	Array of node pressures	PSI
PROP(,1)	Material specific heat	WATTS-SEC/LB-°F

Variable	Description	Dimension
PROP(,2)	Material density	LB/IN^3
PROP(,3)	Material conductivity	WATTS/IN-°F
PUP	Upstream pressure	PSI
Q( )	Array of line end flows	CIS
QA	Magnitude of flow	CIS
QL( )	Array of leg flows	CIS
QN( )	External flow at a node	CIS
QS	Sign of the flow	--
Q1	Flow rate	--
TC( )	Array of component temperatures at line ends	°F
TERM	Dummy Variable	--
TF( )	Array of fluid temperatures at the line ends	°F
TFINAL	Final calculation time	SEC
TIME	Current main program calculation time	SEC
TITLE( )	Program run title array	--
TLF( )	Array of line segment fluid temperatures	°F
TLW( )	Array of line segment wall temperatures	°F
TW( )	Array of wall temperatures at the line ends	°F
VSTORE( )	Array for storage of line and component variable data required for plotting	--

3.4 TFLUID FUNCTION

Function TFLUID reads in the fluid parameters and computes tables of fluid density, adiabatic bulk modulus, and kinematic viscosity for the temperature range of -65°F to 300°F. Data for three types of fluid are currently included in TFLUID. They are MIL-H-5606B, MIL-H-83282 and SKYDROL 500B. TFLUID is dimensioned to accept data on three additional fluids. The data sources are contained in the TFLUID function subprogram.

3.4.1 MATH MODEL - Not applicable.

3.4.2 ASSUMPTIONS - Not applicable.

3.4.3 COMPUTATION METHOD - The arguments of TFLUID require that the temperature and pressure be input for any computation of density, viscosity or bulk modulus.

All the fluid parameters are dimensioned for nine input data points and six fluids. Data statements are used to input the name of each fluid, the nine temperature data points for each fluid, and the bulk modulus and viscosity data corresponding to the nine temperature points for each fluid. Only two points are used for density input data, since a straight line interpolation is used over the entire temperature range for density calculations. The values of specific heat and conductivity are stored for each fluid type in the DCF and DCPFN arrays.

Next the system fluid type, initial temperature vapor pressure and atmospheric pressure are read in. Default values are assigned when a initial temperature, vapor or atmospheric pressures are not assigned by the user.

Subroutine INTEPP is then called to estimate the fluid's viscosity and bulk modulus values from -65°F to 300°F in 2.5 degree increments. The values of the fluid properties are stored in DVISC( ) and DBULK( ). Viscosity is converted from centistokes to NEWTS in the process. A pressure coefficient of viscosity is also computed and stored in DCOEFF( ) for every 2.5 degree increment of temperature.

Finally TFLUID writes out the fluid type and vapor pressure before returning control to the main program.

ENTRY TFLUID

Whenever a value of bulk modulus viscosity or density is required by the component or line subroutines a call is made to the TFLUID function through the appropriate entry statement, with the current values of temperature and pressure. For the bulk modulus computation the temperature is converted to an array location in DBULK( ).

$$IV = (TEMP+65.)/2.5$$

DBULK(IV) gives the value of bulk modulus at the inputted temperature. The value is then pressure corrected and returned to the calling program. The process is similar for the viscosity calculation. Should the temperature exceed 300°F the fluid properties are given at 300°F.

In entry RHO the equation of a straight line drawn between the inputted density data points is solved to obtain the density value.

3.4.4 APPROXIMATIONS

1. The values of specific heat and conductivity are input at 100°F for the different fluid types and are not corrected for any fluid temperature rise during program execution.

2. The fluid property values of bulk modulus and viscosity are only accurate to 2.5 degrees because of the table look-up feature.

3. For any fluid temperatures greater than 300°F the bulk modulus and viscosity value will be given at 300°F.

3 4.5 VARIABLE NAMES

VARIABLE	DESCRIPTION	DIMENSION
A	Temperature Ratio	--
ABULK( )	Array for Ten Adiabatic Bulk Modulus Input Data Points for Six Fluids	PSI
ARHO( )	Array for Two Density Input Data Points for Six Fluids	$LB*SEC^2/IN^4$
ATEMP( )	Array for Ten Temperature Data Points for Six Fluids	TEMP
AVISC( )	Array for Ten Viscosity Input Data Points for Six Fluids	CENTISTOKES
B	Viscosity Correction Exponent	--
COEFF( )	Array of Viscosity Correction Factors	--
DBULK( )	Tabulated Array of Bulk Modulus Values for User Selected Fluid	--
DCF( )	Array of Fluid Conductivities	WATTS/IN-°F
DCOEFF( )	Array of Viscosity Pressure Correction Factors for User Select Fluid	--
DCPFN( )	Array of Fluid Specific Heats	WATTS-SEC/LB-°F
DVISC( )	Tabulated Array of Viscosity Values of User Selected Fluid	IN^2/SEC
I,IEK	Dummy Variables	--
IF	Fluid Type Identification Number	--
IFLUMN( )	Array for Fluid Names	--
IK( )	Number of Input Temperature Points	--
IV	Address of Tabulated Viscosity or Bulk Modulus Values	--
PRESS	Input Fluid Operating Pressure	PSI
PVAP	Fluid Vapor Pressure	PSI

VARIABLE	DESCRIPTION	DIMENSION
SLOPE	Slope of Density-Temperature Line	$LB\text{-}SEC^2/IN^4/°F$
TBULK	Dummy Variable	--
TEMP	Input Fluid Operating Temperature	°F
TFLUID,TJ,TVISC	Dummy Variables	--
Y1,Y2	Input Density Data Corrected for Operating Pressure	$LB\text{-}SEC^2/IN.^4$

3.4.6 Subroutine Listing

```
      FUNCTION TFLUID(TEMP,PRESS)
C**** REVISED MARCH 3, 1975 ****
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION ATEMP(10,6),AVISC(10,6),ABULK(10,6),ARHO(2,6),
     1COEFF(6),IK(6),IFLUNM(3,6),DCF(6),DCPFN(6)
      DIMENSION DVISC(148),DCOEFF(148),DBULK(148)
C
C     SECOND SUBSCPIPT REFERS TO FLUID TYPE (IF PARAMETER)
C
      DATA IFLUNM
     1/8H FOR MIL,8H-H-5606B,8H          ,
     28H FOR MIL,8H-H-83282,8H          ,
     38H FOR SKY,8HDROL 500,8H3         ,
     46*8H          ,
     58H         ,8H         ,8H         /
C
      DATA ATEMP /
     1-65.,-40.,0.,50.,100.,150.,200.,250.,300.,300.,
     2-65.,-40.,0.,50.,100.,150.,200.,250.,300.,300.,
     3-65.,-40.,0.,50.,100.,150.,200.,250.,300.,300.,
     430*10. /
C
C      RHO,BULK AND VISC DATA ARE FOR 0.0 PSIG
C
C     RHO DATA SOURCE:
C     1-MDC REPORT A2686 DATED 4/74
C     2-MDC REPORT A2686 DATED 4/74
C     3-MONSANTO DATA SHEET DATED 6/67(DOUGLAS HYD MANUAL)
      DATA ARHO /
     13.57E-5,7.63E-5,
     28.49E-5,7.3E-5,
     310.3E-5,8.9E-5,6*10./
C
C     BULK DATA SOURCE:
C     1-LETTER TO G.AMES FROM J.W.NOONAN DATED 11/70
C     2-LETTER TO G.AMES FROM J.W.NOONAN DATED 11/70
C     3-LETTER TO G.AMES FROM J.W.NOONAN DATED 11/70
      DATA ABULK /
     113.47E5,3.25E5,2.9E5,2.48E5,2.08E5,1.73E5,1.42E5,1.19E5,.98E5,
     A.98E5,
     213.47E5,3.25E5,2.9E5,2.48E5,2.08E5,1.73E5,1.42E5,1.19E5,.98E5,
     A.98E5,
     334.26E5,4.05E5,3.64E5,3.18E5,2.7E5,2.29E5,1.94E5,1.62E5,1.38E5,
     A1.38E5,30*10. /
C
C     VISC DATA SOURCE:
C     1-MDC REPORT A2686 DATED 4/74
C     2-MDC REPORT A2686 DATED 4/74
C     3-MONSANTO DATA SHEET DATED 6/67(DOUGLAS HYD MANUAL)
      DATA AVISC /
```

```
         11993.5,482.3,134.4,34.85,14.47,7.46,4.58,3.19,2.39,2.39,
         211446.9,2019.3,269.45,48.87,15.95,7.46,4.24,2.83,2.04,2.04,
         33485.5,598.07,104.18,27.9,11.7,6.5,4.18,2.89,2.15,2.15,
         A30*10./
C
      DATA IK/3*9,3*10/
      DATA COEFF/.335,.33,.42,3*10./
      DATA DCF/.0017,.0023,.0022,3*0.0/,
     + DCPFN/552.7,461.48,403.99,3*0.0/
C     INPUT FLUID TYPE AND INITIAL FLUID TEMP
      READ(5,333)IF,FTEMP,PVAP,ATPRES
  333 FORMAT(I5,3E10.0)
C     IF INDICATES THE FLUID TYPE
C     SET SYSTEM FLUID TEMP TO DEFAULT VALUE
      IF(FTEMP.EQ.0.0)FTEMP=100.
C     SET THE VAPOR PRESSURE TO ITS DEFAULT VALUE
      IF(PVAP.EQ.0.)PVAP=2.
C     SET THE ATMOSPHERIC PRESSURE TO ITS DEFAULT VALUE
      IF(ATPRES.EQ.0.)ATPRES=14.5
C
      CF=DCF(IF)
      CPFJ=DCPFN(IF)
      TJ=-65.
      DO 10 I=1,148
      CALL INTERP(TJ,ATEMP(1,IF),AVISC(1,IF),11
     +,IK(IF),TVISC,IER)
C     VISC IS CONVERTED FROM CENTISTOKES TO NEWTS
      DVISC(I)=TVISC*1.555E-3
      CALL INTERP(TJ,ATEMP(1,IF),AVISC(1,IF),12
     +,IK(IF),COEFF(IF),IER)
      DCOEFF(I)=COEFF(IF)
      CALL INTERP(TJ,ATEMP(1,IF),ABULK(1,IF),20
     +,IK(IF),TBULK,IER)
      DBULK(I)=TBULK
      TJ=TJ+2.5
   10 CONTINUE
      TFLUID=TJ
C     WRITE(6,12) (DVISC(I),I=1,148),(DCOEFF(I),I=1,148),
C    + (DBULK(I),I=1,148)
C  12 FORMAT(3X,10E12.5)
      GO TO 20
      ENTRY BULK
      IV=(TEMP+65.)/2.5
      IF(IV.GT.146)IV=146
      TFLUID=DBULK(IV)+12.*PRESS
      RETURN
      ENTRY RHO
C     WRITE(6,13) TEMP,PRESS
C  13 FORMAT(3X,5H*RHO*,3X,2E12.5)
      Y1=ARHO(1,IF)
```

3.4.6 (Continued)

```
      Y2=ARHO(2,IF)
      Y1=Y1*(1.+PRESS/2.5E5)
      Y2=Y2*(1.+PRESS/2.5E5)
      SLOPE=(Y2-Y1)/340.
      TFLUID=SLOPE*(TEMP+65.)+Y1
      RETURN
      ENTRY VISC
      IF(ABS(PRESS).GT.90000)PRESS=90000.
C     WRITE(6,19) TEMP,PRESS,IV
C  19 FORMAT(3X,6H*VISC*,3X,2E12.5,I10)
      IV=(TEMP+65.)/2.5
      IF(IV.GT.146)IV=146
      A=560./(TEMP+460.)
      B=((DCOEFF(IV))**A)*PRESS*2.3E-4
      TFLUID=DVISC(IV)*EXP(B)
      RETURN
   20 WRITE(6,600)(IFLUNM(I,IF),I=1,3),PVAP
  600 FORMAT(22X,15HFLUID DATA FOR ,3A8,25HWITH A VAPOR PRESSURE OF ,
     +F7.1,4H PSI,//)
      RETURN
      END
```

4.0 STEADY STATE SUBROUTINES

The steady state subroutines comprising TSSDATA, TCALC and LEGGAL provide the thermal transient section with the distribution of pressures and flows in the system.

The steady state programs need to know how each constant flow path is connected, where the flow splits and adds, and where there is a net displacement or overboard flow.

This data is input after the component information. The input data used gives great flexibility and is very easy to modify.

The steady state program can cope with system configurations that are very complex and it is particularly valuable with closed loop systems and intertwined flow paths.

4.1 SUBROUTINE TSSDATA

The TSSDATA subroutine reads the input data which specifies one system configuration.

TSSDATA is a simple input routine, with very little calculation. The data storage is divided into two sections; the basic leg data is contained in the NUP, NDWN and NELEM arrays, and the elements in a leg are stored in the ILEG array.

When all the data has been read in, it is written to the output so that a check can be made for errors in each data field.

4.1.1 Math Model - Not applicable.

4.1.2 Assumptions - Not applicable.

4.1.3 Computation Method -

The first set of input data to be read is the number of nodes, NNODE, and the number of legs, NLEG. The data storage arrays are then filled with the steady state leg and element information. The NUP and NDWN arrays contain the upstream and downstream node numbers. The number of elements in the leg is stored in the NELEM array.

The LEG element data is stored in ILEG( ) in data pairs. If the first value is equal to zero, the second is the line number. If the first value is nonzero, it is the component number and the second is the connection number. There are ILEGAD(I) pairs of data for each LEG #I with the first value stored at ILEG (ILEGAD(I)).

4.1.4 Approximations - Not applicable.

4.1.5 Limitations - The steady state data is essentially a restatement of previously inputed thermal transient data, in a form that can be followed during the steady state calculations. A sorting routine would eliminate the need for inputing steady state data, by generating it from the data inputed for the system components.

4.1.6 Variable Names

Name	Description
I	Do Loop Counter
J	Number of Elements in a LEG
JJ	Do Loop Counter
K	Address Counter
NCPN	Dummy Variable
NLEG	Number of LEGS
NNODE	Number of Nodes

4.1.7 Subroutine Listing

```
      SUBROUTINE TSSDATA
C**** REVISED AUGUST 5, 1975 ****
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,NNODE,NLEG,NCPN,TERM,
     +  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NELEM(90),
     +ILEGAD(90),ILEG(1000)
C
C **** READ THE NUMBER OF NODES,LEGS,AND CONSTANT PRESSURE
C        NODES.
C
      READ(5,69)NNODE,NLEG,NCPN
   69 FORMAT(3I5)
C
C ****  WRITE OUT THE INPUTED DATA
C
      WRITE(6,70) NNODE,NLEG,NCPN
      WRITE(6,71)
   70 FORMAT(1H1,55X,23HSTEADY STATE INPUT DATA,//,
     1  30X,17HNUMBER OF NODES =,I3,5X,16HNUMBER OF LEGS =,I3,
     2  5X,35HNUMBER OF CONSTANT PRESSURE NODES =,I3,//)
   71 FORMAT(52X,25HLEG CONNECTION INPUT DATA,
     4  //,10X,6HLEG NO,9X,12HUPST NODE NO,4X,12HDWST NODE NO,4X,
     5 14HNO OF ELEMENTS,5X,10HFLOW GUESS,5X,10HUPST PRESS,5X,
     6 10HDWST PRESS)
      K=0
C
C ***  READ IN DATA FOR EACH LEG
C
      DO 200 II=1,NLEG
      READ(5,76)I,NUP(I),NDWN(I),JJ,QL(I),PUP,PDOWN
   76 FORMAT(4I5,3E10.0)
      WRITE(6,80)I,NUP(I),NDWN(I),JJ,QL(I),PUP,PDOWN
   80 FORMAT(10X,I5,10X,I5,10X,I5,10X,I5,9X,5X,F10.5,
     1  5X,F10.5,5X,F10.5)
      NELEM(II)=JJ
      JJ=JJ*2
C
C     READ LEG ELEMENT DATA
C
      READ(5,199) (ILEG(K+J),J=1,JJ)
  199 FORMAT(16I5)
      ILEGAD(II)=K+1
  200 K=K+JJ
      WRITE(6,90)
  90  FORMAT(1H0,9X,30HLEG NO      ELEMENTS IN LEG----)
      DO 300 II=1,NLEG
      K=ILEGAD(II)
      JJ=NELEM(II)*2-1+K
      WRITE(6,152)II,(ILEG(J),J=K,JJ)
  300 CONTINUE
  152 FORMAT(10X,I3,7X,10(I3,3H --,I3,1H,),//,20X,10(10(I3,3H --,I3,1H,)
     +,//,20X))
      RETURN
      END
```

4.2 SUBROUTINE TCALC

The TCALC subroutine is responsible for the steady state calculations in the system. TCALC is called from the THYTR main program. The subroutine will compute the pressures at all the system nodes and flows in all the legs, using pressure drop data obtained from TLEGCAL. Figure 4.2-1 is a generalized flow diagram of TCALC.

On entry into TCALC the first phase performed by the subroutine will be to initialize the appropriate calculation arrays. After the initialization, the computation phase begins. All the legs will be assigned conductance values from the TLEGCAL subroutine. These conductance values, along with constant factors, will then be inserted into two matrices. The TGAUSS subroutine will be called to compute the new pressure values. These pressure values at the nodes are then used to calculate the new flow rates for the legs in the system. When all the flows pass the convergence test, the flows and pressures are written to labeled common arrays and program control is passed back to THYTR. If the number of iterations exceeds 50, the most recent calculated values of flow and pressure are returned to the labeled common arrays and an error message is printed.

4.2.1 Math Model - The development of the TCALC subroutine to analyze complex flow systems results from the assumption that all resistance factors in a line can temporarily be assumed linear. The net flow around any node can then be written as the sum of all the flows entering and leaving that node or $Q_{NET} = 0$.

If R_{12} is a resistance factor used to describe a resistance in a leg, then $R_{12} = \Delta P_{12}/Q_{12}$.

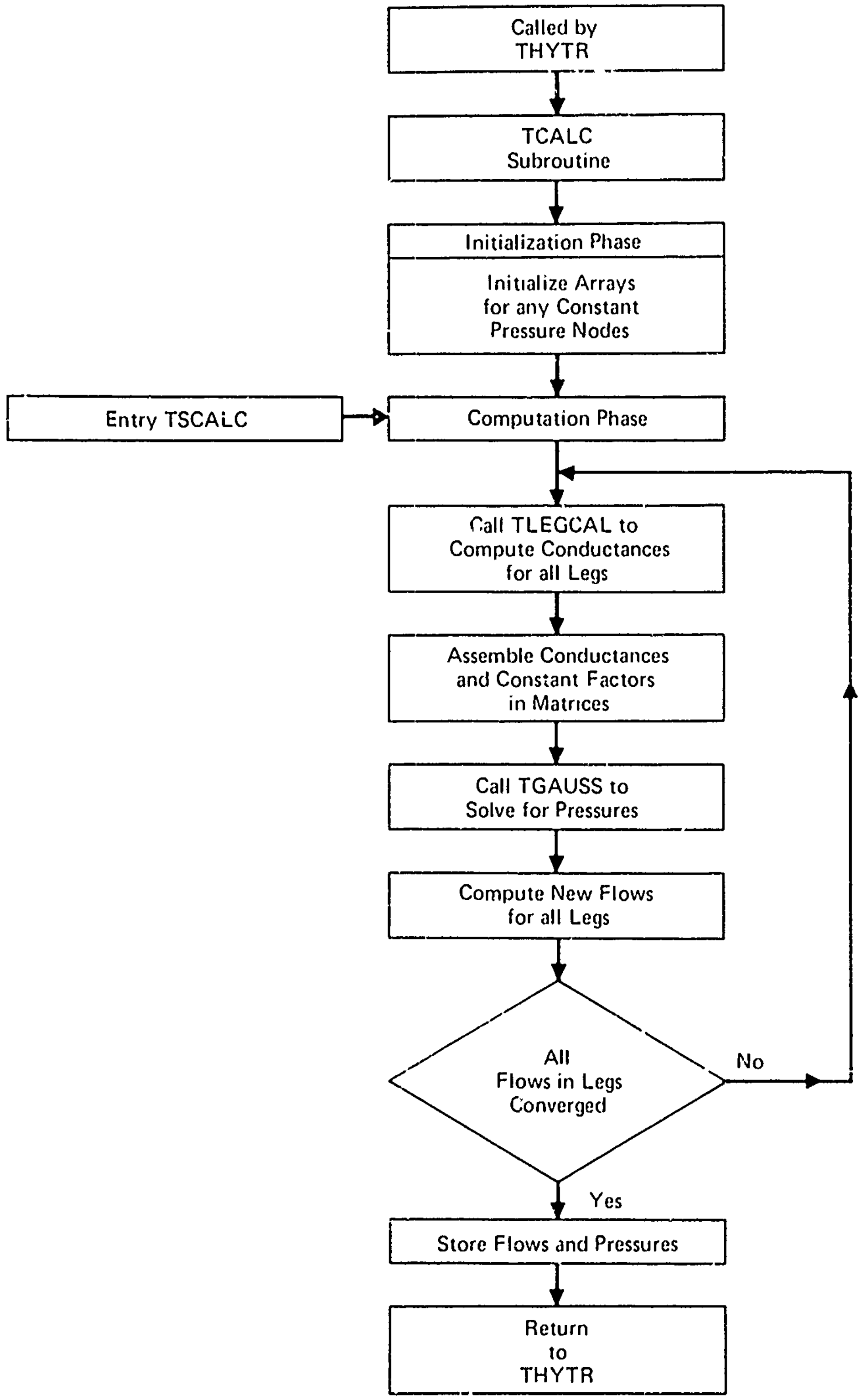


FIGURE 4.2-1
TCALC GENERALIZED FLOW DIAGRAM

GP77-0065-15

where:

R_{12} = Resistance from node 1 to node 2 of the leg

ΔP_{12} = Pressure drop from node 1 to node 2 of the leg

Q_{12} = Flow in the leg

Conductance is then defined as:

$$G_{12} = \frac{1}{R_{12}}$$

where:

G_{12} = Conductance from node 1 to node 2 of the leg

Then:

$$Q_{12} = G_{12} P_{12}$$

The net flow at any node (where three or more legs come together) must be zero.

Therefore, the flow requirement is satisfied if:

$$\Sigma_J G_{IJ} [P_I - P_J \pm \Delta P_{IJ}] - \Sigma_K \pm Q_{IK} = 0$$

Where:

P_I = pressure at node I

P_J = pressure at node J

ΔP_{IJ} = a pressure rise or loss (from a pump or actuator) in leg IJ

Q_{IK} = fixed flow in leg IK connected to node I

Equations of the above form are input to a matrix for solution of pressures at nodes. These matrix solution pressures are used in conjunction with the calculated conductance (G) to calculate a new flow guess in each leg. When two successive flow guesses for all legs in the system are within a specific tolerance such as .001 CIS, the solution has converged. Refer to Appendix A SSFAN Technical Manual, AFAPL-TR-76-43, Vol. VI, for a more detailed mathematical development.

4.2.2 TCALC Subroutine Description - The TCALC subroutine is divided into two phases. The first phase deals directly with the input data for establishing the system pressure node identification arrays. Six arrays are generated which are used in the calculation of node pressures and leg flows in phase two. Specifically, these arrays are:

JCOL:

1) Dimension(M,M)

2) The final JCOL array (in compressed form) identifies the columns in a square CALC1 array which are filled with non-zero terms. The rows of JCOL correspond to the rows of CALC1, and the elements in each row of JCOL correspond to the column number in each row of CALC1.

3) Note: JCOL describes a square CALC1 array in order to be compatible with the solution technique in TGAUSS.

IDIAG:

1) Dimension(M)

2) The IDIAG array identifies which columns of CALC1 contain the positive flow values. IDIAG(1) corresponds to the column in which the positive element is located in the first row of the CALC1 array. IDIAG(2) corresponds to the column in which the positive element is located in the second row of the CALC1 array.

3) Note: IDIAG describes a compressed CALC array.

JNEG:

1) Dimension(ML)

2) The JNEG array identifies which column in CALC1 contains the first appearance (in a row-by-row search) of the leg numbers used as G-subscripts for negative elements in the CALC1 array. For example, JNEG(4) represents the leg to be used as a G-subscript. If JNEG(4)=3, then the first time a $-G_4$ appears is in column 3 of the CALC1 array (the row number is already known).

3) Note: JNEG describes a compressed CALC1 array.

INEG:

1) Dimension(ML)

2) The INEG array differs from the JNEG array in only one respect, that being the INEG array stores the second appearance of the leg numbers used as G-subscripts for negative elements in the CALC1 array.

3) Note: The INEG array describes a compressed CALC1 array.

JRENT:

1) Dimension(M)

2) The JRENT array identifies the number of non-zero entries in each row of CALC1. IRENT is a duplicate of JRENT, however, IRENT is passed to TGAUSS to be used in the solution process while JRENT remains permanent. For this reason, IRENT is easily built from JRENT for each iteration.

3) Note: JRENT describes either a square or compressed CALC1 array.

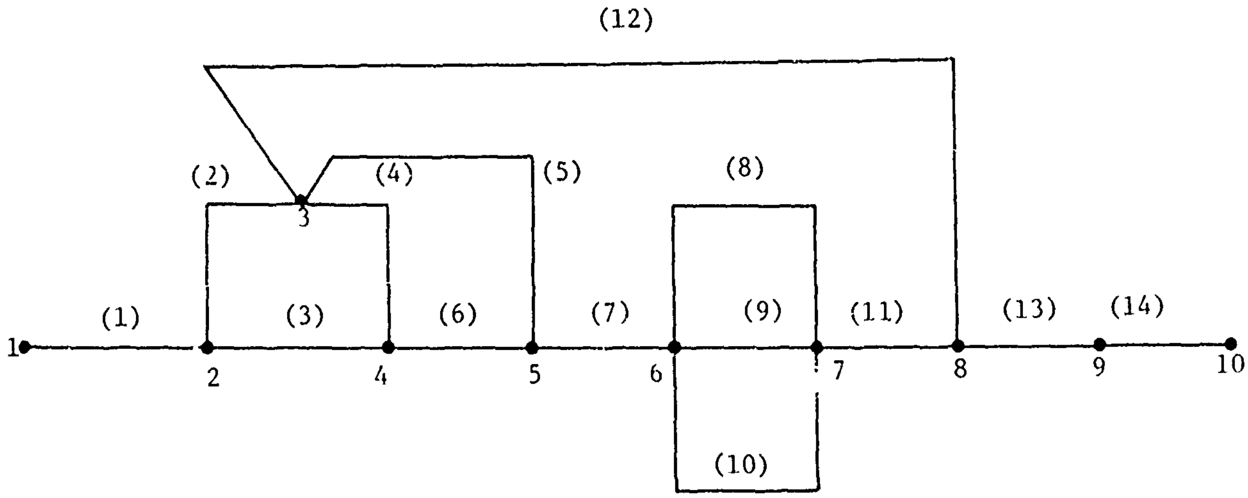
JCENT:

1) Dimension(M)

2) The JCENT array identifies the number of non-zero entries in each column of CALC1. ICENT is built from JCENT for every iteration.

3) Note: JCENT describes a square CALC1.

To understand how Phase I works, the example system in Figure 4.2-2 is developed below. A simplified flow diagram of Phase I is shown in Figure 4.2-3.



1	1	2
2	2	3
3	2	4
4	3	4
5	3	5
6	4	5
7	5	6
8	6	7
9	6	7
10	6	7
11	7	8
12	3	8
13	8	9
14	9	10

ILEP ARRAY

NUMBER OF PRESSURE NODES M = 10

NUMBER OF LEGS ML = 14

Figure 4.2-2

TCALC EXAMPLE SYSTEM

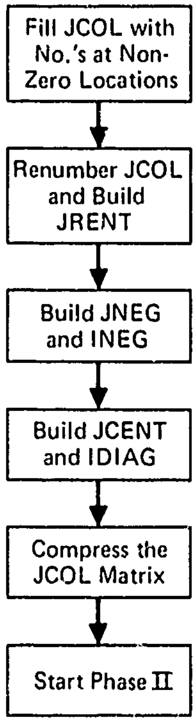


FIGURE 4.2-3
TCALC SUBROUTINE PHASE ONE OPERATION

GP77-0065-18

I. JCOL is initially filled with non-zero terms to indicate the positions of non-zero terms in a square CALC1.

```
DO 10 K=1, ML
I = ILEP(K,2)
J = ILEP(K,3)
JCOL(I,J)=I
JCOL(J,I)=I
JCOL(I,I)=I
JCOL(J,J)=I
```

JCOL =

	1	2	3	4	5	6	7	8	9	10
1	1	1								
2	1	3	2	3						
3		2	12	4	5			12		
4		3	4	6	6					
5			5	6	7	7				
6					7	10	10			
7						10	11	11		
8			12				11	13	13	
9								13	14	14
10									14	14

II. JCOL is renumbered to provide for easy construction of JRENT, JCENT, JNEG, INEG, and IDIAG.

```
C-----RENUMBER JCOL AND BUILD JRENT
      DO 20 I=1,M
      KOUNT=0
      DO 35, J=1,M
      JJ=JCOL(I,J)
      IF(JJ.EQ. ) GO TO 35
      KOUNT=KOUNT+1
      JCOL (I,J)=KOUNT
 35   CONTINUE
      JRENT(I)=KOUNT
 20   CONTINUE
```

JCOL =

	1	2	3	4	5	6	7	8	9	10
1	1	2								
2	1	2	3	4						
3		1	2	3	4			5		
4		1	2	3	4					
5			1	2	3	4				
6					1	2	3			
7						1	2	3		
8			1				2	3	4	
9								1	2	3
10									1	2
	1	2	3	4	5	6	7	8	9	10

JRENT =

2	4	5	4	4	3	3	4	3	2

III. JNEG records the CALC1 column containing the downstream appearance of the leg number as a negative element. INEG records its upstream occurrence.

```
C ------ LOCATE ALL OF OFF-DIAGONAL ELEMENTS
         DO 45 K=1,ML
         I=ILEP(K,2)
         J=ILEP(K,3)
         JNEG(K)=JCOL(I,J)
   45    INEG(K)=JCOL(J,I)
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
JNEG =	2	3	4	3	4	4	4	3	3	3	3	5	4	3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
INEG =	1	1	1	2	1	2	1	1	1	1	2	1	1	1

IV.

```
C ------- BUILD JCENT AND IDIAG
         DO 65 I=1,M
         IDIAG(K)=JCOL(K,K)
         KOUNT=0
         DO 67 J=1,M
         IF(JCOL(J,I).EQ.0)GO TO 67
         KOUNT=KOUNT+1
   67    CONTINUE
   65    JCENT(I)=KOUNT
```

	1	2	3	4	5	6	7	8	9	10
JCENT =	2	4	5	4	4	3	3	4	3	2

	1	2	3	4	5	6	7	8	9	10
IDIAG =	1	2	2	3	3	2	2	3	2	2

V. The JCOL elements are all left-justified, and their previous positions are set equal to zero by the statement JCOL(I,J)=0. This statement must precede JCOL(1,K)=J so that, in the event that J=K, the compressed JCOL matrix contains its non-zero elements in the proper location. ICOL can now be copied from JCOL and be passed to GAUSS for use in the solution process.

```
C ------ COMPRESS THE JCOL MATRIX
         DO 70 I=1,M
         NN =JRENT(I)
         J=0
         DO 70 K=1,NN
   75    J=J+1
         K1=JCOL(I,J)
         IF(K1.EQ.0)GO TO 75
         JCOL(I,J)=0
         JCOL(I,K)=J
   70    CONTINUE
```

JCOL =

	1	2	3	4	5
1	1	2			
2	1	2	3	4	
3	2	3	4	5	8
4	2	3	4	5	
5	3	4	5	6	
6	5	6	7		
7	6	7	8		
8	3	7	8	9	
9	8	9	10		
10	9	10			

Phase two operation of the CALC subroutine begins with initializing the conductance array - CALC1, and the constant array - CALC2, to zero values. (See Figure 4.2-4 for a flow diagram of the phase two operation.) A call is now made to the subroutine TLEGCAL for each leg in the system. TLEGCAL will return the value of conductance to the G array in the unlabeled common.

After all the conductance values are calculated for each leg, they must be entered into the compressed CALC1 array. For the example system the CALC1 array contains:

```
C ------ BUILD CALC1 MATRIX
         DO 9099 K=1,ML
         I=ILEP(K,2)
         J=ILEP(K,3)
         L=IDIAG(I)
         LM=IDIAG(J)
         CALC1(I,L)=CALC1(I,L)+G(K)
         CALC1(J,LM)=CALC1(J,LM)+G(K)
         L=JNEG(K)
         LM=INEG(K)
         CALC1(I,L)=CALC1(I,L)-G(K)
         CALC1(J,LM)=CALC1(J,LM)-G(K)
9099     CONTINUE
```

	1	2	3	4	5
1	G_1	$-G_1$	0	0	0
2	$-G_1$	$G_1+G_2+G_3$	$-G_2$	$-G_3$	0
3	$-G_2$	$G_2+G_4+G_5+G_{12}$	$-G_4$	$-G_5$	$-G_{12}$
4	$-G_3$	$-G_4$	$G_3+G_4+G_6$	$-G_6$	0
5	$-G_5$	$-G_6$	$G_5+G_6+G_7$	$-G_7$	0
6	$-G_7$	$G_7+G_8+G_9+G_{10}$	$G_8-G_9-G_{10}$	0	0
7	$-G_8-G_9-G_{10}$	$G_8+G_9+G_{10}+G_{11}$	$-G_{11}$	0	0
8	$-G_{12}$	$-G_{11}$	$G_{11}+G_{12}+G_{13}$	$-G_{13}$	0
9	$-G_{13}$	$G_{13}+G_{14}$	$-G_{14}$	0	0
10	$-G_{14}$	G_{14}	0	0	0
	1	2	3	4	5

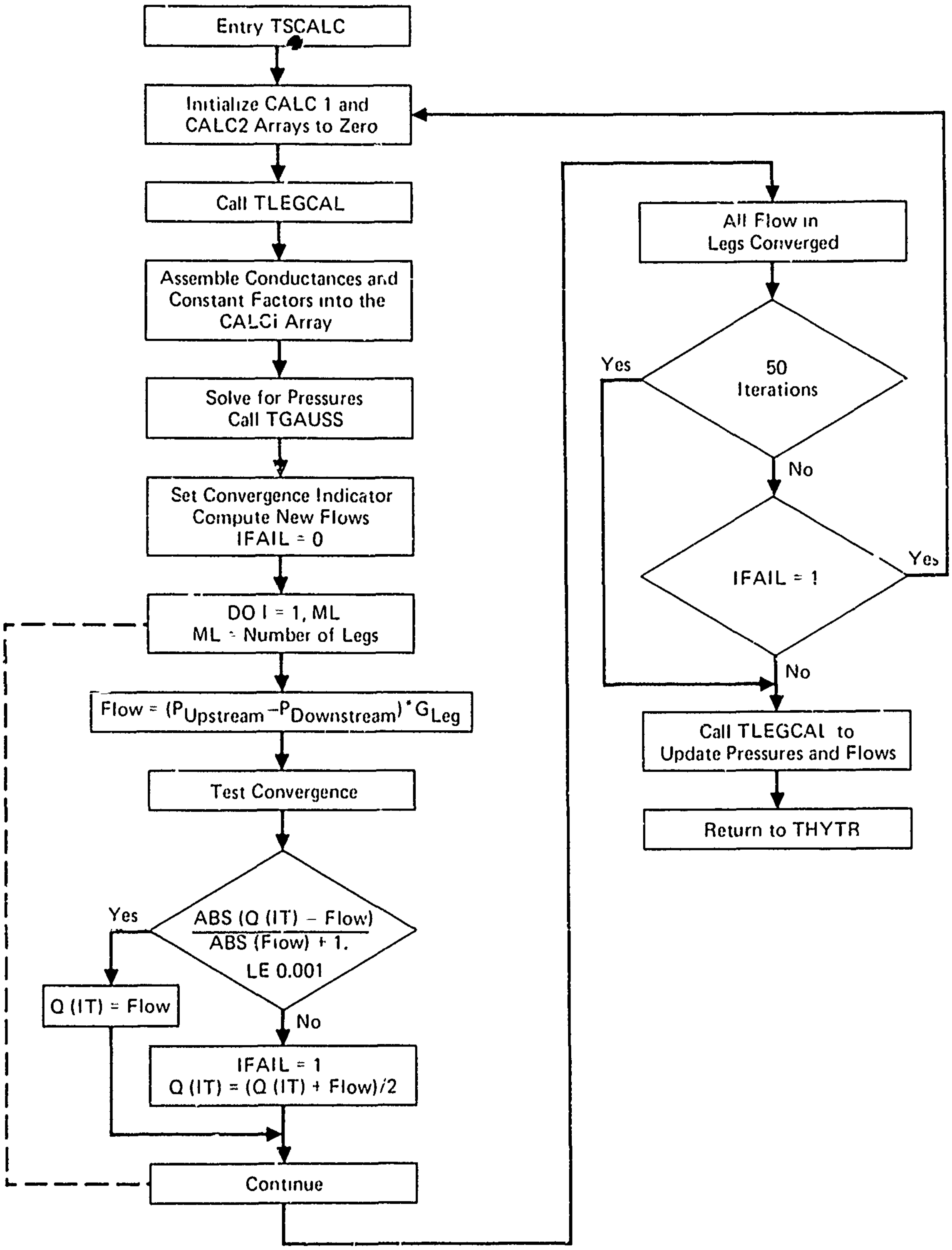


FIGURE 4.2-4
CALC SUBROUTINE PHASE TWO OPERATION

GP77-0065-17

CALC1 is built in this manner for each iteration. This compressed form speeds the solution process.

The CALC2 array contains the constant terms of the system of linear equations that describe the model. Constant pressure drops in legs, external flows and constant pressure sources are all inserted into this array. Any constant pressure source or pressure drop is multiplied by the conductance of the leg it is associated with. If leg (6) has a pressure drop term - PDLEG(6), then PDLEG(6) will be multiplied by the conductance for leg (6) which is G(6), making the resulting term a flow. Thus, all external flows have no multiplication factor.

With both CALC1 and CALC2 filled, the TGAUSS subroutine is called to solve for pressures in the system. The answers are returned through the CALC1 array and then put into the PN array which contains all the system node pressures. Now a new flow is calculated for each leg in the system based on the recent calculation of the pressures. The new flow is equal to the difference of pressures between the nodes of the leg plus any constant pressure drops all multiplied by the conductance of the leg.

The solution for flows in all the legs are final when all the previous flows (Q) and the latest calculated flows (FLOW) are within a specified tolerance. For all flows if

$$\text{ABS}\ \frac{(\text{FLOW}-\text{Q(IT)})}{\text{ABS(FLOW)}+1} \leq .001 \tag{1}$$

then the flows have converted.

If equation (1) is not satisfied in each leg of the system a new value of flow will be computed in each leg by the following equation:

$$Q(IT) = \frac{Q(IT) + FLOW}{2} \tag{2}$$

These new flows will then be given to TLEGCAL for computation of new conductance values for another iteration. If all the legs do not converge after fifty iterations, the cycle will stop and all the current values will be used as the steady-state variables. Before transfer is made back to THYTR a last call is made to TLEGCAL to distribute pressure drops and flows for the steady state conditions.

4.2.3 Computations. The only direct computation made in the solution of the steady state values in TCALC is the calculation of FLOW. The purpose of this of course, is to establish an error tolerance in flows that is reduced through iterations to meet the convergence criteria as discussed in the previous section. The majority of the TCALC subroutine handles the bookkeeping necessary to manipulate the leg and node numbers to compute system pressures and flows.

4.2.4 Approximations. The coefficients of the CALC1 array are linearily approximated to represent the system conductances. Inherent approximations exist in some of the constant data in CALC2.

4.2.5 Limitations. Most limitations exist in the areas of physical discontinuities. TCALC was written to solve a flow balance in a system. Any flow discontinuities that occur, such as in a simple unbalanced actuator, must have mathematical formula to describe what happens to the flow. TCALC also requires the leg pressure drops to be continuous over a specified flow range. When this does not occur, as in a check valve, the proper input from the check valve subroutine must be fed to TCALC so it may respond to the changed conditions. Refer to Appendix D SSFAN Technical Manual for a more thorough discussion on the limitations of TCALC.

4.2.6 Variable Names

Variables	Description	Dimensions
CALC1( )	Array of conductances	--
CALC2( )	Array of constants	--
FLOW	Latest value of leg flow	CIS
I	DO loop counter	--
G	Array of conductances	CIS/PSI
IFAIL,IFLAG	Indicators	--
IL,IM	Dummy variables	--
INEG( ),JNEG( )	Arrays containing location of off diagonal conductance values	--
JFCOL( )	Computational array	--
ITER	Iteration counter	--
IU,IV,J,JJ,JL, JX,JY,K,KOUNT, K1,2,LM,L1	Dummy variables	--
M	Number of nodes	--
ML	Total number of legs	--
PN( )	Array of node pressures	PSI
PDLEG( )	Location of pressure drops or increases	PSI
PEX	Array of external pressure constants	PSI
QL( )	Array of leg flows	CIS
QN( )	Flow gain or loss at a pressure node changed to an M matrix of constants	CIS

4.2.7 Subroutine Listing

```
      SUBROUTINE TCALC
**** REVISED JULY 07,1976 ****
      DOUBLE PRECISION CALC1,CALC2
      COMMON G(90),CALC2(55),JPCOL(55,20),CALC1(55,20)
      COMMON/ICC/ICOL(55,20),JRENT(55),JCENT(55)
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,M,ML,NCPN,TERM,
     +  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NLLEM(90),
     +ILEGAD(90),ILEG(1000)
      DIMENSION IDIAG(55),JCOL(55,55),JNEG(90),INEG(90)
      EQUIVALENCE(JPCOL(1,1),JCOL(1,1))
      WRITE(6,900)
  900 FORMAT(1H1,50X,30HSTEADY STATE CALCULATION DATA )
      DO 5 I=1,M
      QN(I)=0.0
      PEX(I)=0.0
 5    PN(I)=0.0
      DO 6 I=1,ML
    6 PDLEG(I)=0.0
      DO 80 I=1,55
      DO 80 J=1,55
 80   JCOL(I,J)=0
      DO 10 K=1,ML
      I=NUP(K)
      J=NDWN(K)
      JCOL(I,J)=I
      JCOL(J,I)=I
      JCOL(I,I)=I
      JCOL(J,J)=I
 10   CONTINUE
C-----RENUMBER JCOL AND BUILD JRENT
      DO 20 I=1,M
      KOUNT=0
      DO 35 J=1,M
      JJ=JCOL(I,J)
      IF(JJ.EQ.0) GO TO 35
      KOUNT=KOUNT+1
      JCOL(I,J)=KOUNT
 35   CONTINUE
      JRENT(I)=KOUNT
 20   CONTINUE
C-----LOCATE ALL THE OFF-DIAGONAL ELEMENTS
      DO 45 K=1,ML
      I=NUP(K)
      J=NDWN(K)
      JNEG(K)=JCOL(I,J)
 45   INEG(K)=JCOL(J,I)
C-----BUILD JCENT AND IDIAG
      DO 65 I=1,M
      IDIAG(I)=JCOL(I,I)
      KOUNT=0
      DO 67 J=1,M
```

```
      IF(JCOL(J,I).EQ.0) GO TO 67
      KOUNT=KOUNT+1
 67   CONTINUE
 65   JCENT(I)=KOUNT
C-----COMPRESS THE JCOL MATRIX
      DO 70 I=1,N
      NN=JRENT(I)
      J=0
      DO 70 K=1,NN
 75   J=J+1
      K1=JCOL(I,J)
      IF(K1.EQ.0) GO TO 75
      JCOL(I,J)=0
      JPCOL(I,K)=J
 70   CONTINUE
C     INITIALIZE CALC1 AND CALC2 ARRAYS TO ZERO
      ENTRY TSCALC
      ITER=1
      WRITE(6,910)
 910  FORMAT(//,24X,10HFLOW OULSS,4X,13HPRESSURE DROP,6X,9HLEG DELTP,
     + 9X,3HPUP,12X,5HPDOWN,9X,11HCONDUCTANCE,/)
  200 DO 220 L1=1,M
      DO 210 K1=1,20
      ICOL(L1,K1)=0
  210 CALC1(L1,K1)=0.
      PEX(L1)=0.0
      QK(L1)=0.0
  220 CALC2(L1)=0.
      DO 221 L1=1,NL
  221 PDLEG(L1)=0.0
C     COMPUTE G*S FOR CALC ARRAYS
      CALL TLEGCAL
      DO 9099 K=1,NL
      I=NUP(K)
      J=NDWN(K)
      L=IDIAG(I)
      L4=IDIAG(J)
      CALC1(I,L)=CALC1(I,L)+G(K)
      CALC1(J,L4)=CALC1(J,L4)+G(K)
      L=JNEG(K)
      L4=INEG(K)
      CALC1(I,L)=CALC1(I,L)-G(K)
      CALC1(J,L4)=CALC1(J,L4)-G(K)
 9099 CONTINUE
      DO 700 IL=1,55
      DO 700 JL=1,20
 700  ICOL(IL,JL)=JPCOL(IL,JL)
      DO 400 JX=1,NL
      IF(PDLEG(JX).EQ.0.)GO TO 400
      JY=NUP(JX)
      CALC2(JY)=CALC2(JY)-PDLEG(JX)*G(JX)
      JY=NDWN(JX)
```

4.2.7 (Continued)

```
      CALC2(JY)=CALC2(JY)+PDLEG(JX)*G(JX)
  400 CONTINUE
      DO 60 I=1,M
      J=IDIAG(I)
      CALC2(I)=CALC2(I)+QN(I)
 60   CALC1(I,J)=CALC1(I,J)+PLX(I)
C     WRITE(6,2005)((CALC1(I,J),J=1,20),I=1,M)
C     WRITE(6,2005)(CALC2(I),I=1,M)
C     WRITE(6,2004)((ICOL(I,J),J=1,20),I=1,M)
 2004 FORMAT(1X,20I6)
 2005 FORMAT(1X,10E12.5)
C     WRITE(6,2005)(PLX(I),I=1,M)
C     WRITE(6,2005)(PDLEG(I),I=1,ML)
C     WRITE(6,2005)(QN(I),I=1,M)
      CALL TGAUSS(M,ITER)
      DO 410 IM=1,M
 410  PN(IM)=CALC1(IM,1)
      WRITE(6,9000)(PN(I),I=1,M)
      WRITE(6,9001)
 9000 FORMAT(1H0,(5X,14HNODE PRESSURES,2X,8F12.3,/))
 9001 FORMAT(1H0)
      IFAIL=0
C     CALCULATE NEW FLOW RATES
      DO 435 IT=1,ML
      IU=NUP(IT)
      IV=NDWN(IT)
      QOLD=QL(IT)
      FLOW=((PN(IU)+PDLEG(IT)-PN(IV))*G(IT))
C     TEST NEW FLOW RATES
      IF(ABS(FLOW-QOLD)/(ABS(FLOW)+1.).GT.0.0001)GO TO 436
C     RECALCULATE FLOW RATES
      QL(IT)=FLOW
      GO TO 435
  436 QL(IT)=(QOLD+FLOW)/2.0
      IFAIL=1
 435  CONTINUE
      IF(IFAIL.EQ.0)GO TO 520
      IF(ITER.EQ.50)WRITE(6,999)
  999 FORMAT(10X,44H**** WARNING EXCEEDED 50 ITERATIONS IN TCALC,
     + 19H-PROGRAM CONTINUING,//)
      IF(ITER.EQ.50)GO TO 520
      ITER=ITER+1
      GO TO 200
  520 CONTINUE
C     MAKE A LAST CALL TO ALL LEGS TO DISTRIBUTE PRESSURE
C     DROPS AND FLOWS CALCULATED FOR STEADY STATE CONDITIONS
C     DO 521 I=1,90
C521  PDLEG(I)=0.0
      CALL TLEGCAL
      RETURN
      END
```

4.3 SUBROUTINE TLEGCAL

TLEGCAL is called by the TCALC to obtain a leg conductance and fixed pressure drops for a given flow guess for all the system.

The constant pressure drop such as that across a check valve, relief valve, actuator, or pump, is passed via PDLEG(NLEG) in common/steady. A positive PDLEG(NLEG) is a pressure rise such as at a pump, a negative is a drop such as across a check valve. The leg conductance is passed via G(NLEG) in common.

TLEGCAL obtains the line and component pressure drops by calling all the elements in the leg. The leg conductance is computed by dividing the leg flow by the leg pressure drop.

A flowchart of TLEGCAL's organization is shown in Figure 4.3-1.

4.3.1 Theory

LEGCAL calls the elements in a leg to determine the pressure drop for a given flow.

The leg conductance (inverse of resistance) is calculated from the leg pressure drop, including the constant pressure drop value

G(NLEG) = QA/ABS(DELTP)

where

DELTP = PN(NUP(NLEG))-PUP+PDLEG(NLEG)

The conductance is always positive. Using this formula the conductance value is flow dependent. It has to be updated whenever the flow guess is changed.

4.3.2 Assumptions

The assumption that the pressure drop can be described using the leg pressure drop is generally valid. If for some reason an element in a leg cannot be described in this manner then a pseudo description can

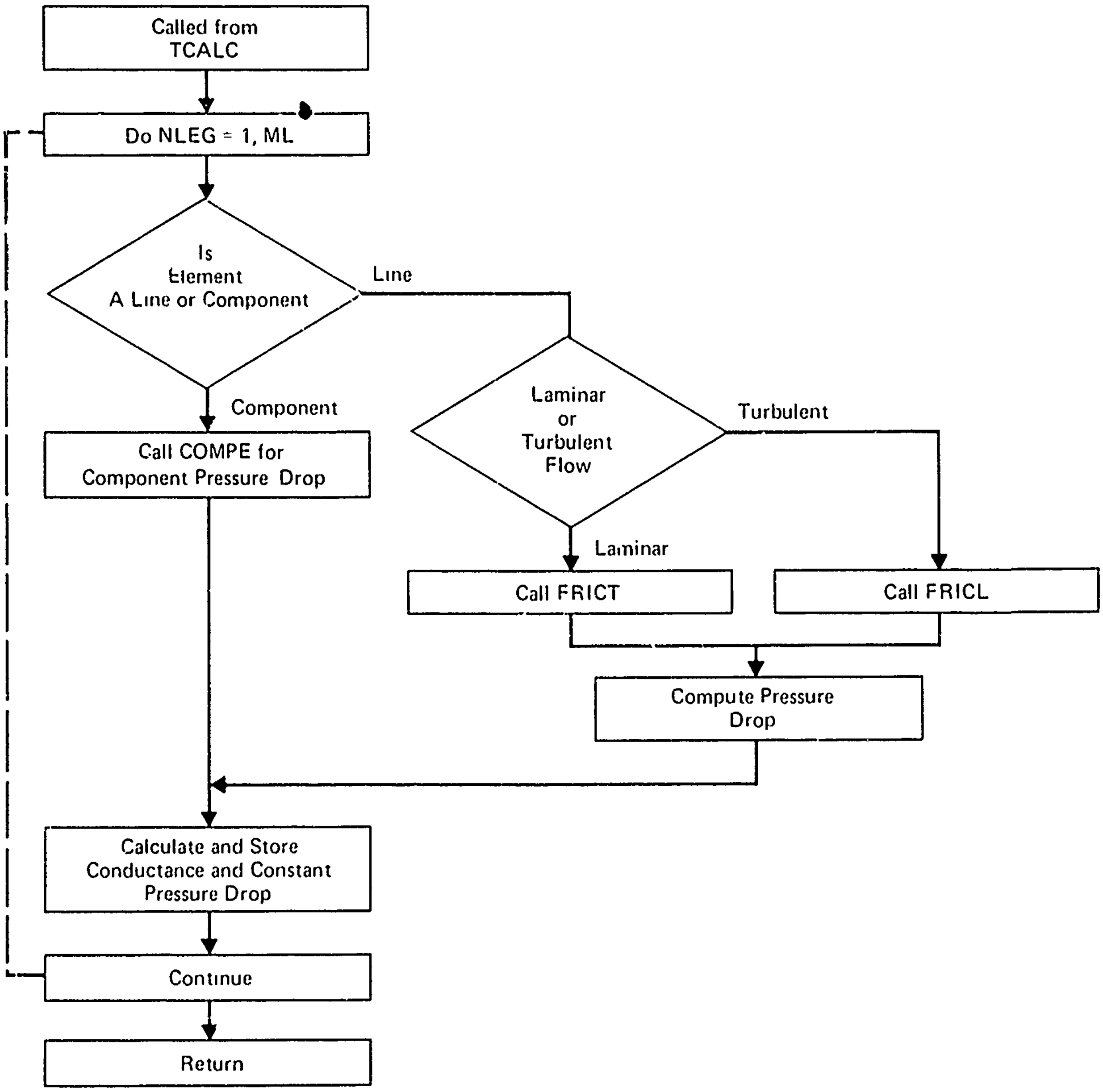


FIGURE 4.3-1
TLEGCAL ORGANIZATION

GP77-0065-18

be used without loss of accuracy. This could involve generation a formula of the form

$$\Delta P = K_1 + K_2Q + K_3Q^{1.75} + K_4Q^2$$

where Q = leg flow

The line and component subroutines would then provide the values to the K_1, K_2, K_3 and K_4 constants.

4.3.3 Computation Method

The variable Q1, the new flow guess, is first split into its absolute value and its sign, ± 1.0. The up and downstream node pressures for the leg are taken from the PN( ) array.

Each element in a leg is called and the pressure drop through the line or component is calculated and subtracted from the upstream pressure, PUP. Once the entire leg pressure drop has been determined the new conductance value is computed. The variables IND, and KNEL are the component number and the connection number respectively.

The common variables INZ and INX are set equal to the number of elements in the leg and the actual element that is being calculated respectively. This allows particular component subroutines to determine which end of the leg they are connected to, and hence which node is located at the component.

4.3.4 Approximations

The use of a formula requires some approximations but these are usually related to approximations in the component model and are an integral part of the component model. In general this method is good but it could be easily extended to a higher order approximation if it was found desirable.

4.3.5 Limitations

So far we have not found any limitations to the technique used in TLEGCAL itself.

However, some of the component subroutines called by TLEGCAL such as the bootstrap reservoir,pump and actuators, are complicated by the interaction between the flow guesses, flow direction and node pressures.

Some of these subroutines use calculations which, though conforming to the basic calculation technique, do not fall into any simple category and have to be treated individually.

4.3.6 Variable Names

Variable	Description	Dimensions
DELTP	Line Pressure Drop	PSI
INZ	Number of Elements in Leg	--
I	Address of Leg Data in ILEG	--
K	Ith Element in a Leg	--
KNEL	Component Connection Number	--
IND	Component or Line Number	--
ML	Total Number of Legs	--
PUP	Flow Dependent Leg Pressure Drop	PSI
QA	ABS Value of Leg Flow	CIS
QT	Leg Transition Flow	CIS
Q1	Leg Flow Guess	--
QS	Flow Sign CIS	--

For variables in common refer to Paragraph 3.3.

4.3.7 Subroutine Listing

```
      SUBROUTINE TLEGCAL
C**** REVISED AUGUST 5, 1976 ****
      COMMON G(90)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
     + NLSEG(150)
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDWN,M,ML,NCPN,TERM,
     +  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NELEM(90),
     +ILEGAD(90),ILEG(1000)
C     FIND THE SIGN OF THE FLOW GUESS AND ITS ABSOLUTE VALUE
      DO 200 NLEG=1,ML
      TERM=0.0
      INV=1
      Q1=QL(NLEG)
      QA=ABS(Q1)
      IF(QA.LE..00001)QA=.00001
      QS=SIGN(1.0,Q1)
C     CALCULATE THE FORMULAE FOR THE LEG PRESSURE DROP
      INEL=NLEG
      INZ=NELEM(NLEG)
C     INZ - NO OF ELEMENTS IN A LEG
C     INEL - LEG NUMBER
      PUP=PN(NUP(NLEG))
      PDWN=PN(NDWN(NLEG))
C     WRITE(6,900)NLEG,INZ,NUP(NLEG),NDWN(NLEG)
  900 FORMAT(10X,5I10)
C     WRITE(6,910)PUP,PDWN,Q1
  910 FORMAT(10X,5E12.5)
C     CALL EACH ELEMENT IN THE LEG
C
      I=ILEGAD(NLEG)
      DO 600 K=1,INZ
C     INX - CURRENT ELEMENT NO. IN LEG
      INX=K
      IND=ILEG(I)
      KNEL=ILEG(I+1)
      I=I+2
      ICON=KNEL
C      ICON - CONNECTION NO.
C    IF THE ELEMENT IS A LINE GO TO 500
      IF(IND.EQ.0) GO TO 500
      CALL COMPE
      GO TO 600
C *** THIS SECTION ADDS THE VALUES INTO THE FORMULAE FOR THE LINES
C
  500 CONTINUE
      LOC=KNEL*2-1
```

```
      LOCD=KNEL*2
C     WRITE(6,503) LOC,TF(LOC),PUP
C 503 FORMAT(3X,1I10,2E12.5)
      QT=PARM(KNEL,4)*VISC(TF(LOC),PUP)
      IF(QA.GT.QT)GO TO 505
      DELTP=QA*FRICL(KNEL,TF(LOC),PUP)
      GO TO 593
 505  DELTP=FRICT(KNEL,TF(LOC),PUP)*QA**1.75
  593 CONTINUE
      P(LOC)=PUP
      IF(TERM.GT.0.0)P(LOC)=TERM
      PUP=P(LOC)-DELTP
      P(LOCD)=PUP
      Q(LOC)=-Q1
      Q(LOCD)=Q1
      GO TO 600
  600 CONTINUE
      DELTP=PN(NUP(NLEG))-PUP+TERM
      G(NLEG)=QA/ABS(DELTP)
      WRITE(6,50)NLEG,Q1,PDLEG(NLEG),DELTP,PN(NUP(NLEG)),
     + PN(NDWN(NLEG)),G(NLEG)
      IF(INV.EQ.0)GO TO 200
      PDLEG(NLEG)=0.0
   50 FORMAT(13H       LEG NO ,I3,5F16.5,E20.8)
  200 CONTINUE
      RETURN
      END
```

5.0 LINE SUBROUTINE

5.1 SUBROUTINE TLINEA

TLINEA simulates a line or pipe connected to two components. It divides each line into segments, the length of each segment being not less than the volume flow rate times the time step divided by the cross sectional area of the fluid. The subroutine calculates the wall and fluid temperatures of each line segment.

The values of the flows and pressures in the line are calculated by the steady state subroutines.

TLINEA is called by the main program at the beginning of each new time step and uses the latest values of pressures and flows, to compute line segment temperatures.

5.1.1 Math Model - The line is represented by n number of segments. Each segment consists of two nodes, one fluid and one wall, and the end segments are connected to components as shown in Figure 5.1-1.

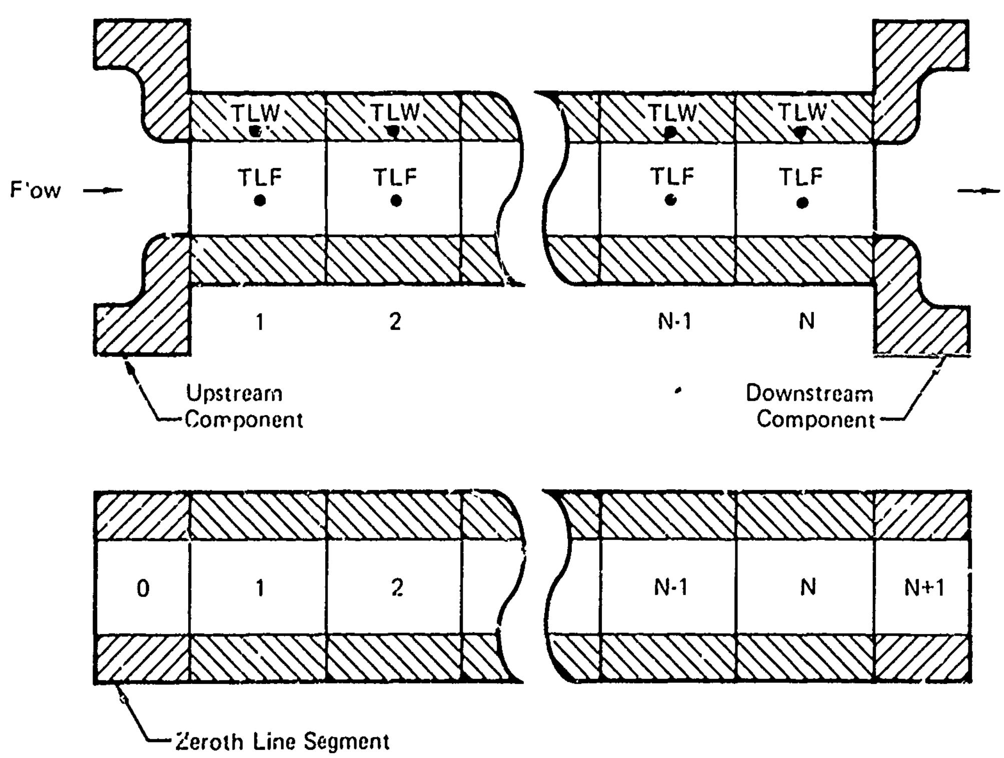


FIGURE 5.1-1
LINE NODE REPRESENTATION

GP77-0065-5

The line calculations are conducted as follows:

1. The component wall and fluid temperatures are predicted for the present time step. TC and TF equal the component wall and fluid temperatures of the last time step and TCO and TFO are equal to the component wall and fluid temperatures of the 2nd to last time step. The predicted wall and fluid temperatures are computed as:

   TCP = TC*2 - TCO

   TFP = TF*2 - TFO

2. A zeroth line segment is assumed to exist (upstream of the first line segment) having wall and fluid temperatures equal to the predicted component temperatures.

3. As shown in Figure 5.1-2, the predicted temperatures of the zeroth segment and the previous time step calculated values of the second segment are used to calculate the new temperatures of the first segment.

4. The new temperatures of the first segment and the previous time step temperatures of the third segment are used to calculate the new temperatures of the second segment. This continues until the n-1 segment temperatures have been calculated.

5. The n+1 line segment is assumed to exist (downstream of the nth line segment) having wall and fluid temperatures equal to the previous time step calculated values of the component.

6. The new temperatures of the n-1 line segment and the previous temperatures of the n+1 line segment are used to calculate the new temperatures of the nth segment.

The math model for the line includes heat transfer to and from components attached to each line end as well as to and from individual segments of the line. For the calculation six nodes are considered: three fluid nodes and three wall nodes as shown in Figure 5.1-2. The temperatures of the J-1 wall and fluid nodes

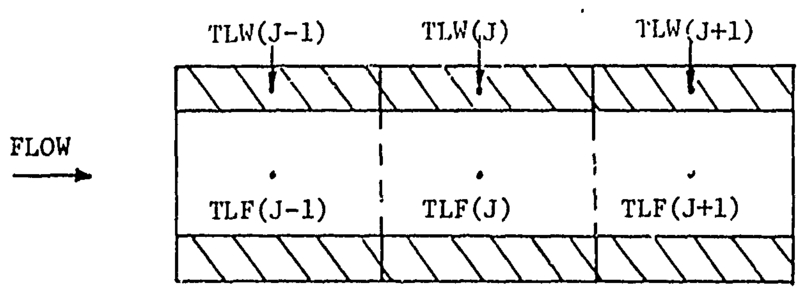


FIGURE 5.1-2

LINE SEGMENT NODE REPRESENTATION

are TLW(J-1) and TLF(J-1), the temperatures of the J wall and fluid nodes are TLW(J) and TLF(J), and the temperatures of the J+1 wall and fluid nodes are TLW(J+1) and TLF(J+1). Two heat transfer equations are written to solve for TLW(J) and TLF(J), using the line material properties and dimensions, the atmosphere and structure temperatures external to the line, and TLW(J-1) TLW(J=L), and TLF(J-1). (Note: TLF(J+1) = TLF(J), see assumptions). One equation is for heat transferred to and from the J fluid node. The second equation is for heat transferred to and from the J wall node.

The first equation represents three modes of heat transfer with the J fluid node:

1. conduction to and from the J-1 fluid node

   R1*(TLF(J-1)-TLF(J))

   where R1 is the conduction coefficient and is equal to CF/(2.*DxF(INO)/ACF(INO)+RMF*DELT/(ACF(INO)**2*RHOIL))

2. convection to and from the J wall node

   B9*(TLW(J) - TLF(J))

   where B9 is the convection coefficient and is equal to UFWIL*ASFW(INO)

3. heat transfer due to mass transfer into the J segment from the J-1 segment.

   $\dot{M}Cp$*(TLF(J-1)-TLF(J))

   where $\dot{M}Cp$ is the flow rate and is equal to Q(L1)*RHOIL*CPFN

These heat transfer modes are combined to produce the equation for heat transferred to and from the J fluid node:

$$\frac{MCp}{DELT}*(TLF(J)-TLF(J)_{OLD})=R1*(TLF(J-1)-TLF(J))+B9*(TLW(J)-TLW(J))+$$
$$Q(IL)*RHOIL*CPFN*(TLF(J-1)-TLF(J)) \qquad (1)$$

where MCp is equal to FNM(INO)*CPFN

The second equation represents three modes of heat transfer with the J wall node:

1. conduction to and from the J-1 and J+1 wall nodes respectively

   R3*(TLW(J-1)-TLW(J))

   R4*(TLW(J+1)-TLW(J))

   where R3 and R4 are the conduction coefficients equal to CW(INO)/2.0*DXF(INO)/ACW(INO))

2. a. convection to and from the J fluid node

   B9*(TLF(J)-TLW(J))

   where B9 was defined previously

2. b. convection to and from the external atmosphere

   C3*(TA(INO)-TLW(J))

   where C3 is the convection coefficient and is equal to UAW(INO)*ASAW(INO)

3. radiation exchange with the surrounding structure

   $CIP*(TST(INO)-(TLW(J)+460)^4)$

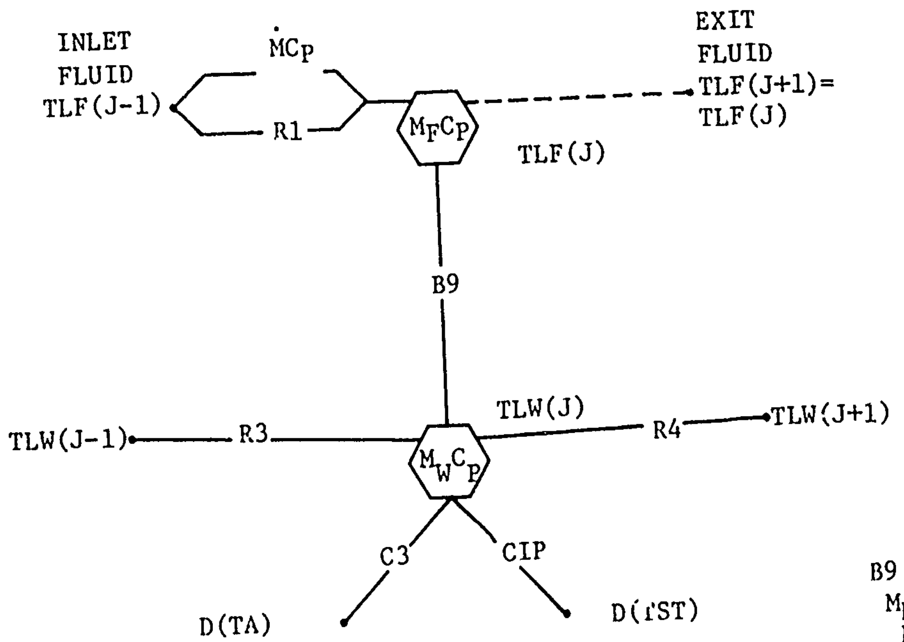
   where CIP is the radiation coefficient and is equal to SIGMA*SHAPF*EPSION*ASAW(INO).

These heat transfer modes are combined to produce the equation for heat transferred to and from the J wall node:

$$
\begin{aligned}
\frac{MCp}{DELT} *(TLW(J)-TLW(J)) = {} & R3*(TLW(J-1)-TLW(J))+ \\
& R4*(TLW(J+1)-TLW(J))+ \\
& B9*(TLF(J)-TLW(J))+ \\
& C3*(TA(INO)-TLW(J))+ \\
& CIP*(TST(INO)-(TLW(J)+450)^4)
\end{aligned}
\qquad (2)
$$

where MCp is equal to WNM(INO)*CPWN

A thermal model of the above heat transfer terms for three line segments is shown in Figure 5.1-3.



B9,C3 = CONVECTION
M_NCp = STORAGE
$\dot{M}Cp$ = FLOW
R_N = CONDUCTION
CIP = RADIATION

FIGURE 5.1-3

THERMAL MODEL

5.1.2 Assumptions

1. Atmosphere and structure temperatures remain constant.
2. Predicted temperatures are based upon the previous two calculated values.
3. Temperature of the fluid that leaves each node is the temperature calculated for this node, TLF(J).
4. The interface conductance between the line and components is infinite.
5. The wall node is all at the same temperature.
6. The emissivity of the walls is a constant (.3 is used for steel).
7. Transition from laminar to turbulent flow is assumed to occur at a Reynolds number of 1200.
8. Friction factors used are based on circular cross-section, smooth ID, drawn tubing.

5.1.3 Computational Methods

Section 1000

Each line data card is read in and the equivalent line lengths for bends and fittings is calculated

```
EQUIVL=DIA*(N45ELB*12.+N90ELB*57.+
        NALT90/(45.*4.65)+NAGT90/(90.*7.5)
        +PLENGTH(N)*DIA)/100.
```

The equivalent line length is then added to the actual line length and is used to calculate the laminar and turbulent flow constants.

If the line segment length DELTAX(INO) is left blank a value of 36 inches is assigned and this number of line segments is computed. Line temperatures and input physical parameters are assigned to the appropriate arrays and the data is printed. The next set of line data cards is then read in. The above is repeated until the data for all the lines are read in.

Section 2000

The segment length for each line is recalculated based on the fluid flow in the line. If the segment length is larger than the read in value, the appropriate arrays in line are initialized to reflect the different segment size and the new segment length is printed out. Otherwise this program continues to check the next line.

After each line has been interrogated the fluid and wall temperatures for each line segment node is initialized. The heat transfer coefficient for the fluid to the line wall is calculated as

UFWIL=UFW(AAA, DDD, ABS(Q(INAU)), TF(INAU), P(INAU))

The old predicted temperatures are also calculated

TFO(INAU)=TF(INAU)

TCO(INAD)=TC(INAD)

Section 3000

All the constants dependent on the current values of the line volume flow rates and pressures are computed.

If there is only one segment then the fluid and wall temperatures are calculated with upstream and downstream nodes being the end components.

If there is more than one segment then the first segments fluid and wall temperatures are calculated with the upstream node being the upstream component. The other line segments are calculated as discussed in the math model until the last segment is reached.

The last segment is calculated with its downstream nodes being the downstream components. The two temperatures of each segment (wall and fluid) are stored to be used by the line subroutine at next time step. The four end segment wall and fluid temperatures are stored in COMMON/TRANS/arrays to be transferred to the other subroutines.

5.1.4 Approximations

(1) The predicted temperatures at the beginning of each line are approximated on the past time history of the component temperatures.

5.1.5 Limitations

See technical summary.

5.1.6 Variable Listing

Variable	Description	Dimension
A( )	dummy computational array	
AAA	dummy variable	
ACF	cross sectional area of the fluid	IN^2
ACW	cross sectional area of the wall	IN^2
ASAW	external surface area of the wall segment	IN^2
ASFW	internal surface area of the wall segment	IN^2
A2,A9,B9	dummy variables	
B( )	dummy computational array	
C,CW	thermal conductivity of the walls	WATTS/IN-°F
CF	thermal conductivity of the fluid	WATTS/IN-°F
CID1	dummy variable	-
CID2	dummy variable	-
CIP	radiation coefficient	WATTS/°F
CPFN	specific heat of the fluid	WATTS-sec/LBm-°F
CPWN	specific heat of the walls	WATTS-sec/LBm-°F
C1,C12,C3, C9	dummy variables	-
DDD	dummy variable	-
DELTAX	distance of each line segment	IN
DIA	outside diameter of the line or wall	IN
DIAINS	outside diameter of the line	IN
DXF	distance from the line segment node to interface with next segment	IN

Variable	Description	Dimension
EPSION	emissivity factor for radiation	-
EQUIVL	equivalent line length	IN
FNM	fluid mass of each node	LB_m
FTEMP	dummy variable	-
FLTEMP	input fluid temperature	°F
FWTEMP	dummy variable	-
INAD	downstream leg address	-
INAU	upstream leg address	-
INO	line number	-
J,JJ,JM1	dummy variables	-
MTYPE	material type of the line wall	-
NAGT90	number of bends greater than 90°	-
NALT90	number of bends less than 90°	-
NPFRIC	percentage increase in line friction	-
N45ELB	number of 45° elbows	-
N90ELB	number of 90° elbows	-
PLENGTH	total line length	IN
REN	Reynolds number	-
RHOIL	fluid density	LBm/IN^3
RHOW	wall density	LBm/IN^3
RMF	dummy variable	-
R1,R3,R4	dummy variable	-
SHAPF	radiation shape factor	-
SIGMA	Stefan-Boltzmann radiation constant	$WATTS/IN^2$-°R^4
SPMAF	dummy variable	-
SPMAW	dummy variable	-
TA	surrounding atmospheric temperature	°F
TC	storage variable (temperature of the component)	°F

Variable	Description	Dimension
TCO	storage variable (old temperature of the component)	°F
TCP	predicted component temperature	°F
TF	storage variable	-
TFO	storage variable (old temperature of the component fluid)	°F
TFP	predicted fluid temperature	°F
TLF	temperature of the segment fluid	°F
TLW	temperature of the segment wall	°F
TST	surrounding structure temperature	°F
TW	storage variable	
UAW	external heat transfer coefficient of the wall	WATTS/IN^2-°F
UFWIL	internal heat transfer coefficient of the wall	WATTS/IN^2-°F
WNM	mass of each line wall segment	LBm
WTHICK	wall thickness of the line	IN

5.1.7 Subroutine Listing

BEST AV...

```
      SUBROUTINE TLINLA
C *** REVISED MAY 1, 1976 ***
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NLL
      COMMON /LIMIT/MNLINE,MNLL,MNLEG,MNNODE,MNPLOT,MNLPTS,MDS
      COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
     + NLSEG(150)
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION TA(300),TST(300),UAW(300),TFO(300),DELTAX(150),
     +TCO(300),SPMAF(150),SPMAW(150),PLENGTH(150),A(2,2),B(2)
      DIMENSION LC(300),ASAW(150),ASPM(150),FNM(150),CW(150)
     + ,WNM(150),FWTEMP(300)
      EQUIVALENCE (C(1),LC(1)),(PARM(1,1),PLENGTH(1))
      DATA SIGMA/.349E-11/,SHAPF/.96/,EPSION/.3/
      REN=1200.
      IF(IENTR)1000,2000,3000
 1000 CONTINUE
C       INO      =INDIVIDUAL LINE NUMBER
C       LINET=LINE TYPE
C       NPFRIC=PERCENTAGE INCREASE IN LINE FRICTION
C       NPWT  =PERCENTAGE INCREASE IN WEIGHT
C       N45ELB=NUMBER OF 45 DEG ELBOWS
C       N90ELB=NUMBER OF 90 DEG ELBOWS
C       NALT90=TOTAL OF BEND ANGLES .LT. 90 DEG    DEG
C       NAGT90=TOTAL OF BEND ANGLES .GE. 90 DEG    DEG
C       LINETH(N)=LINE LENGTH
C       DIA   =OUTSIDE DIAMETER
C       WTHICK=WALL THICKNESS
C       MTYPE =MATERIAL TYPE
C       DELTAX(INO)=DISTANCE OF SEGMENTS
C       UAW(N)=HEAT TRANSFER COEFF. AMBIENT TO WALL
C       TA(N) =TEMP. OF AMBIENT, DEG. F
C       TLF(N) =TEMP. OF FLUID, DEG. F
C       TST(N)=TEMP. OF STRUCTURE, DEG. R
C
C       PARM(N,1) = LINE LENGTH
C       PARM(N,2) = INSIDE LINE DIA-DIAINS
C       PARM(N,3) = EQUIVL
C       PARM(N,4) = TRANSISTION FLOW
      LSTART(1)=1
      WRITE(6,400)
  400 FORMAT(//11H   LINE DATA,/10H   LINE NO.,5X,6HLENGTH,5X,
     + 8HINTERNAL,7X,4HWALL,8X,6HDELTAX,8X,7HAMBIENT,4X,9HSTRUCTURE,
     + 6X,5HFLUID,7X,8HMATERIAL,/29X,3HDIA,7X,9HTHICKNESS,21X,
     +4HTEMP,8X,4HTEMP,9X,4HTEMP,10X,4HTYPE)
      DO 1100 INO=1,NLINE
      READ(5,433) N, MTYPE,NPFRIC,NPWT,N45ELB,N90ELB,NALT90,NAGT90
  433 FORMAT(8I5)
      READ(5,439) PLENGTH(INO),DIA,WTHICK,DELTAX(INO),UAW(INO),
```

```
     + TST(INO),TA(INO),FLTEMP
  439 FORMAT(8E10.0)
      IF(INO.NE.N) WRITE(6,430) N
  430 FORMAT(1X,43H THE LINE CARDS ARE OUT OF ORDER AT NUMBER ,I5)
      M=N*2
      LC(M)=1
      LC(M-1)=1
      IF(LINET.LT.10) GO TO 65
      LC(M)=-1
      LINET=LINET-10
   65 LQUIVL=DIA*(N45ELB*12.+N90LLB*57.+NALT90/(45.*4.65)+
     + NAGT90/(90.*7.5)+PLENGTH(N)*DIA)/100.
C     CALCULATE NUMBER OF SEGMENTS
      IF(UAW(INO).EQ.0.0) UAW(INO)=0.0056
      IF(DELTAX(INO).EQ.0.0) DELTAX(INO)=36.
      IF(FLTEMP.EQ.0.0) FLTEMP=FTEMP
      NLSEG(INO)=PLENGTH(N)/DELTAX(INO)
      LSTART(INO+1)=LSTART(INO)+NLSEG(INO)
      IND=INO
      INAD=INO*2
      INAU=INO*2-1
      RHOW=PROP(MTYPE,2)
      CPWN=PROP(MTYPE,1)
      FWTEMP(INAD)=FLTEMP
      FWTEMP(INAU)=FLTEMP
      DIAINS=DIA-2.0*WTHICK
      PARM(N,2) = DIAINS
      PARM(N,3) = EQUIVL
      PARM(N,4) = .7854*REN*DIAINS
      CV(INO)=PROP(MTYPE,3)
      ACW(IND)=PI*(DIA**2-DIAINS**2)/4.0
      ACW(INAD)=ACW(IND)
      ACW(INAU)=ACW(IND)
      WLM(IND)=ACW(IND)*RHOW*DELTAX(INO)
      SPHAW(IND)=WLM(IND)*CPWN
      ASAW(IND)=PI*DIA*DELTAX(INO)
      ACF(IND)=PI*DIAINS**2/4.0
      ACF(INAU)=ACF(IND)
      ACF(INAD)=ACF(IND)
      ASFW(IND)=PI*DIAINS*DELTAX(INO)
      TF(INAU)=FLTEMP
      TF(INAD)=FLTEMP
      TW(INAU)=FLTEMP
      TW(INAD)=FLTEMP
      WRITE(6,410)N,PLENGTH(N),DIAINS,WTHICK,DELTAX(N),TA(N),
     + TST(N),FLTEMP,MTYPE
  410 FORMAT(/1X,I5,8X,F8.4,4X,F8.4,5X,F8.4,5X,F8.4,7X,F8.4,4X,F8.4
     + ,5X,F8.4,6X,I4)
      TST(IND)=(TST(IND)+460.)**4
 1100 CONTINUE
```

5.1.7 (Continued)

```
      IF(NLINE.GT.NNLINE-1) GO TO 252
      RETURN
  252 WRITE(6,475) NLINE,LSTART(NLINE),NLSEG(NLINE)
  475 FORMAT(5X,25HERROR IN SUBROUTINE TLINE,3I10)
      STOP 5101
 2000 CONTINUE
C     INITALIZING ALL TEMP. IN LINE
      DO 2012 I=1,NLINE
      XDEL=ABS(Q(I*2-1))*DELT/(PARM(N,2)**2*PI/4.)
      IF(XDEL.LE.DELTAX(I))GO TO 2012
      DELTAX(I)=XDEL
      IF(DELTAX(I).GT.PLENGTH(I))DELTAX(I)=PLENGTH(I)
      WRITE(6,900)I,DELTAX(I)
  900 FORMAT(10X,13HTHE DELTAX IN LINE,I5,22H HAS BEEN CORRECTED TO,
     + F15.5,7H INCHES,//)
      NLSEG(I)=PLENGTH(I)/DELTAX(I)
      ASFW(I)=PI*PARM(I,2)*DELTAX(I)
      WNM(I)=ACW(I)*RHOW*DELTAX(I)
      SPMAW(I)=WNM(I)*CPWH
 2012 ASAW(I)=PI*PARM(I,2)*DELTAX(I)
 2013 INO=1
 2010 J=LSTART(INO)
      INAD=INO*2
      INAU=INO*2-1
      AAN=ACF(INO)
      DDD=PARM(INO,2)
      UFWIL=UFW(AAN,DDD,ABS(Q(INAU)),TF(INAU),P(INAU))
      TLW(J)=FWTEMP(INAU)
 2020 TLF(J)=FWTEMP(INAU)
      JM1=J-1
      IF(JM1.EQ.0) JM1=1
      TLW(J)=TLF(J)
C     TLW(J)=(UAW(INO)*ASAW(INO)*TA(INO)-UFWIL*ASFW(INO)*TLF(J)
C    1 +SIGMA*EPSION*SHAPF*ASAW(INO)*(TST(INO)))/(UAW(INO)*ASAW
C    2 (INO)-UFWIL*ASFW(INO)+SIGMA*EPSION*SHAPF*ASAW(INO)*TLW(JM1)**3)
      J=J+1
      IF(J.LE.(LSTART(INO)+NLSEG(INO)-1)) GO TO 2020
      C(INAU)=CW(INO)
      C(INAD)=CW(INO)
      TFO(INAU)=TF(INAU)
      TFO(INAD)=TF(INAD)
      TCO(INAU)=TC(INAU)
      TCO(INAD)=TC(INAD)
      DXF(INO)=0.5*DELTAX(INO)
      DXF(INAD)=DXF(INO)
      DXF(INAU)=DXF(INO)
      LSTART(INO+1)=LSTART(INO)+NLSEG(INO)
      INO=INO+1
      IF (INO.LE.NLINE) GO TO 2010
      RETURN
```

5.1.7 (Continued)


```
 3000 CONTINUE
      DO 3550 INO=1,NLINE
      IL=LSTART(INO)
      AAA=ACF(INO)
      DDD=PARM(INO,2)
      RHOIL=386.4*RHO(TLF(IL),P(INO))
      UFWIL=UFW(AAA,DDD,ABS(Q(INO)),TLF(IL),P(INO))
      FNM(INO)=ACF(INO)*RHOIL*DELTAX(INO)
      SPANF(INO)=FNM(INO)*CPFN
      N=NLSEG(INO)
      M=LSTART(INO)
      JJ=M+N-1
C     JJ IS THE LAST NODE
      J=M
C     J IS THE FIRST NODE
      IS=1
      IL=INO*2-1
      IF(Q(IL).LT.0.0) GO TO 3100
      IL=IL+1
      IS=-IS
      J=JJ
C     J IS THE LAST NODE
      JJ=M
      M=J
 3100 CONTINUE
      C1=ABS(Q(IL))*RHOIL*CPFN
      C3=UAW(INO)*ASAW(INO)
      CIP=SIGMA*EPSION*SHAPE*ASAW(INO)
      C9=UFWIL*ASFW(INO)
      RMF=ABS(Q(IL))*RHOIL
C     BEGINNING CALCULATION OF LINE TEMPERATURES
      IF(Q(IL).LT.0.0) GO TO 3033
      CID1=TC(IL+IS)
      CID2=TF(IL+IS)
      TC(IL+IS)=TC(IL)
      TC(IL)=CID1
      TF(IL+IS)=TF(IL)
      TF(IL)=CID2
 3033 TFP=TF(IL)*2.0-TFO(IL)
      TCP=TC(IL)*2.0-TCO(IL)
      TCO(IL)=TC(IL)
      TFO(IL)=TF(IL)
      TCO(IL+IS)=TC(IL+IS)
      TFO(IL+IS)=TF(IL+IS)
      IF(N.GT.1)GO TO 3200
C     THERE IS ONLY ONE SEG.
 3050 R1=CF/(2.0*DXF(INO)/ACF(INO)+RMF*DELT/(ACF(INO)
     +   **2*RHOIL))
      R3=(CW(INO))/(2.0*DXF(INO)/ACW(INO))
      R4=R3
```

5.1.7 (Continued)

```
      A2=SPMAW(INO)/DELT+R3+B9+C3+R4
      A9=SPMAF(INO)/DELT+C1+R1+B9
      A(1,1)=A9
      A(1,2)=-B9
      A(2,1)=-B9
      A(2,2)=A2
      B(1)=SPMAF(INO)*TLF(J)/DELT+(R1+C1)*TFP
      B(2)=SPMAW(INO)*TLW(J)/DELT+R3*TCP+R4*TC(IL+IS)
     + +C3*TA(INO)+CIP*TST(INO)-CIP*(TLW(J)+460.)**4
      CALL SIMULT(A,B,2,IERROR)
      C9=B(1)
      C12=B(2)
      TLF(J)=B(1)
      TLW(J)=B(2)
      GO TO 3500
 3200 CONTINUE
C     FIRST LINE SEG.
      R1=CF/(2.0*DXF(INO)/ACF(INO)+RMF*DELT/(ACF(INO)
     + **2*RHOIL))
      R3=1.0/(2.0*DXF(INO)/(ACW(INO)*CW(INO)))
      R4=R3
      A9=SPMAF(INO)/DELT+R1+B9+C1
      A2=SPMAW(INO)/DELT+R3+R4+B9+C3
      A(1,1)=A9
      A(1,2)=-B9
      A(2,1)=-B9
      A(2,2)=A2
      B(1)=SPMAF(INO)*TLF(J)/DELT+(R1+C1)*TFP
      B(2)=SPMAW(INO)*TLW(J)/DELT+R3*TCP+R4*TLW(J+IS)
     ++C3*TA(INO)+CIP*TST(INO)-CIP*(TLW(J)+460.)**4
      CALL SIMULT(A,B,2,IERROR)
      TLF(J)=B(1)
      TLW(J)=B(2)
      C9=B(1)
      C12=B(2)
 3300 IF(N.EQ.2)GO TO 3400
      J=J+IS
C     CALCULATING INNER SEG.
      A(1,1)=A9
      A(1,2)=-B9
      A(2,1)=-B9
      A(2,2)=A2
      B(1)=SPMAF(INO)*TLF(J)/DELT+(R1+C1)*TLF(J-IS)
      B(2)=SPMAW(INO)*TLW(J)/DELT+R3*TLW(J-IS)+R4*TLW(J+IS)
     + +C3*TA(INO)+CIP*TST(INO)-CIP*(TLW(J)+460.)**4
      CALL SIMULT(A,B,2,IERROR)
      TLF(J)=B(1)
      TLW(J)=B(2)
      N=N-1
      GO TO 3300
```

5.1.7 (Continued)

```
C      CALCULATING LAST NODE
 3400 CONTINUE
      J=J+IS
      R4=1.0/(2.0*DXF(INO)/(ACW(INO)*CW(INO)))
      A(1,1)=A9
      A(2,1)=-B9
      A(1,2)=-B9
      A(2,2)=A2
      B(1)=SPMAF(INO)*TLF(J)/DELT+(R1+C1)*TLF(J-IS)
      B(2)=SPMAW(INO)*TLW(J)/DELT+R3*TLW(J-IS)+R4*TC(IL+IS)
     + +C3*TA(INO)+CIP*TST(INO)-CIP*(TLW(J)+460.)**4
      CALL SIMULT(A,B,2,IERROR)
      TLF(J)=B(1)
      TLW(J)=B(2)
C      TLF(J)=(SPMAF(INO)*TLF(J)/DELT+(R1+C1)*TLF(J-IS)
C     +  +B*((SPMAW(INO)*TLW(J)/DELT+R3*TLW(J-IS)+R4*TC(IL+IS)
C     + +C3*TA(INO)-CIP*((TLW(J)+460.)**4)+CIP*TST(INO))/(A2
C     + )))/(A-B*B/A2)
C      TLW(J)=(SPMAW(INO)*TLW(J)/DELT+R3*TLW(J-IS)+R4*TC(IL+IS)
C     + +C3*TA(INO)+CIP*TST(INO)-CIP*((TLW(J)+460.)**4)+B*TLF(J))/A2
      IF(D(IL).LT.0.0) GO TO 3500
C      TF(IL+IS)=TLF(J)
      TF(IL)=TLF(J)
C      TW(IL+IS)=TLW(J)
      TW(IL)=TLW(J)
      TF(IL+IS)=C9
      TW(IL+IS)=C12
      GO TO 3599
 3500 TF(IL+IS)=TLF(J)
C      TF(IL)=TLF(J)
      TW(IL+IS)=TLW(J)
C      TW(IL)=TLW(J)
      TF(IL)=C9
      TW(IL)=C12
 3599 CONTINUE
 3550 CONTINUE
      RETURN
      END
```

6.0 COMPONENT SUBROUTINES

The components modeled vary from the simple restrictor to a hydraulic pump. Each model is broken down into its most basic equations of motion, flow, and heat transfer. The sum total of all these equations can be very complex but individually they are usually simple.

New subroutines can be added without difficulty and, if the computer system can tolerate unsatisfied external references that are not called during execution then component subroutines not in use, can be omitted from the input deck or file when not required. Figure 6.1-1 shows the basic subroutine organization.

In working with the program it is necessary to get a good grasp of the fundamentals involved in the simulation, even the most carefully checked routine can have traps built in which are not always found until the output data is carefully examined by someone who knows what it should look like.

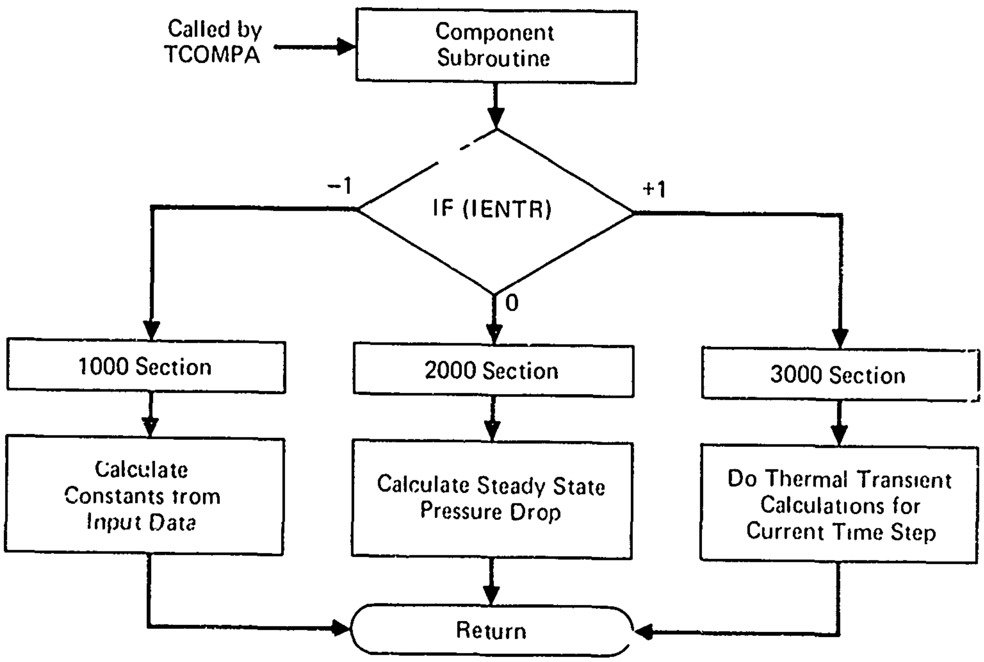


FIGURE 6.1-1
COMPONENT SUBROUTINE ORGANIZATION

GP77 0065 12

6.1 SUBROUTINE TCOMPA

Subroutine TCOMPA which is called by THYTR, reads and prints all component input data, sorts out connection data for all components, and calls each individual component in the order in which it was input.

6.1.1 Math Model

Not applicable.

6.1.2 Assumptions

Not applicable.

6.1.3 Computation Methods

Section 1000

A component Integer card is read and its data is printed. Connection data for the component is sorted out and stored. The real data cards for the component are then read and printed, if any exists. Next, the addresses of the component's real data, temporary, double precision and integer variables are established. Finally, the individual component subroutine is called passing as arguments its starting address in the real data, temporary, double precision and integer arrays. This process is repeated until all input components have been called.

Section 2000

This section consists of a DO loop that ranges from 1 to the number of input components. Within the loop a component group type is isolated by taking its type number, dividing by 10 and forcing truncation (due to the use of integers). This truncated value is then used in a computed GO TO statement to direct control to a statement or section that calls the specific component. If there is more than one component of that group type, specific component isolation is accomplished by subtracting the group component type number from the individual component type number and using the resulting value in a computed GO TO statement. This GO TO statement then directs control to a statement that calls the component. ENTRY COMPE isolates and calls each component in a simple and straightforward manner aiding in the running of the overall program since every component has to be called each time step in the thermal transient calculations and each iteration in the steady state solution.

6.1.4 Approximations

No applicable.

6.1.5 Limitations

No applicable.

6.1.6 Variable Names

Variable	Description	Units
I	Counter	
J	Counter	
KK	Dummy Variable for L4	
KTYPE	Dummy Variable for LTYPE( )	
LDATAC	Number of Element Real Data Cards	
L1	Data value 1	
L4	Data value 1	
N	Counter	
NCI	Number of Last Active Connection	
NDATAC	LDATAC/1000	
NLIM	Number of Real Data Fields	
NN	Dummy Variable Representing Maximum Number of Component Connections	
NTYPE	Group Type Number	
NX	Data Value 2	
N1	Dummy Variable Representing Max Number of Real Data Fields for a Given Component	
N2	Dummy Variable Representing Max Number of Temporary Variables for a Given Component	
N3	Dummy Variable Representing Max Number of Double Precision Variables for a Given Component	
N4	Dummy Variable Representing Max Number of Integer Variables for a Given Component	

6.1.7 Subroutine Listing

```
      SUBROUTINE TCOMPA
C *** REVISED AUGUST 5, 1976 ***
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /LIMIT/MNLINE,MNEL,MNLEG,MNNODE,MNPLOT,MNLPTS,MDS
      COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
     + NLSEG(150)
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,MEL
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + QA,QS,QI,PUP,PDOWN,NNODE,NLEG,NCFN,TERM,
     +  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NLLEG(90),
     +ILEGAD(90),ILEG(1000)
      COMMON /COMPD/D(4500),L(1500),LL(99,4)
      DIMENSION DD(1400),LT(10,150),ND(150,10),LC(1)
      EQUIVALENCE (L(1),LT(1,1)),(D(1),DD(1)),(C(1),LC(1))
      DATA L1,L4,NX/1,1,2/
      IND=1
      IF(IENTR) 1000,2000,2000
 1000 CONTINUE
      DO 1001 I=1,10
      DO 1002 J=1,150
 1002 ND(J,I)=LT(I,J)
 1001 CONTINUE
 1003 CONTINUE
      NCI=0
C     THIS READ STATEMENT INPUTS THE FOLLOWING DATA
C     I      =INDIVIDUAL COMPONENT NUMBER
C     LTYPE =COMPONENT TYPE
      READ(5,1009) I,LTYPE(I),LDATAC,(L(L4-1+J),J=1,12)
 1009 FORMAT(16I5)
      WRITE(6,500) I,I,LTYPE(I),LDATAC,(L(L4-1+J),J=1,12)
      NDATAC=LDATAC/1000
      LDATAC=LDATAC-NDATAC*1000
      NLIM=LDATAC*8
C***  LDATAC EQUALS THE NUMBER OF ELEMENT REAL DATA CARDS
      KTYPE=LTYPE(I)
      NC(I)=ND(KTYPE,7)
      IF(NC(I).EQ.0) GO TO 7
      KK=L4
      NN=L4+NC(I)-1
      DO 4 N=KK,NN
      IF(L(N)) 1,2,3
    1 L(N)=-L(N)*2-1
      LC(L(N))=0
      NCI=NCI+1
      GO TO 4
    2 L(N)=MNLINE*2-ND(KTYPE,9)
      GO TO 4
    3 L(N)=L(N)*2
      LC(L(N))=0
```

6.1.7 (Continued)

```
      NCI=NCI+1
    4 CONTINUE
    5 IF(L(NN).LT.NNLINL*2-1) GO TO 6
      NC(I)=NC(I)-1
      NN=NN-1
      GO TO 5
    6 CONTINUE
      INV=NCI
      IF(NCI.NE.NC(I).AND.ND(KTYPE,10).NE.0)GO TO 420
      IF(NC(I).LT.ND(KTYPE,8)) NC(I)=ND(KTYPE,8)
    7 N1=ND(KTYPE,1)
      N2=ND(KTYPE,2)
      N3=ND(KTYPE,3)
      N4=ND(KTYPE,4)
      IF(NLIN.EQ.0) GO TO 15
      READ(5,9) (D(L1+N-1),N=1,NLIN)
    9 FORMAT(8E10.0)
      IF(IND.NE.I) GO TO 410
      NN=L1
      DO 10 KK=1,LDATAC
      WRITE(6,510)KK,(D(NN+N-1),N=1,8)
      NN=NN+8
   10 CONTINUE
   15 INX=0
      IF (NDATAC.NE.0) GO TO 20
      IF (LDATAC.GT.ND(KTYPE,6)) N1=NLIN
      LL(I,1)=L1
      LL(I,2)=L1+N1
      GO TO 30
   20 CONTINUE
      LL(I,1)=LL(NDATAC,1)
      LL(I,2)=L1
      INX=1
   30 IF(NX*N3.LE.1) GO TO 40
      LL(I,3)=(LL(I,2)+N2+3)/2
      L1=(LL(I,3)+N3)*2+1
      GO TO 50
   40 LL(I,3)=LL(I,2)+N2
      L1=LL(I,3)+N3
   50 LL(I,4)=L4
      L4=L4+N4
      ENTRY COMPL
 2000 CONTINUE
      KTYPE=LTYPE(IND)
      NTYPE =KTYPE/10
      N1=LL(IND,1)
      N2=LL(IND,2)
      N3=LL(IND,3)
      N4=LL(IND,4)
      GO TO (210,220,230,240,250,260,270,280,290,300),NTYPE
```

6.1.7 (Continued)

```
C 360 GO TO 400
  210 CONTINUE
      CALL T3RAN11 (D(N1),D(N2),DD(N3),L(N4))
      GO TO  400
  220 CONTINUE
      KTYPL=KTYPL-20
      GO TO (221,222),KTYPL
 221  CALL TVALV21 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
 222  CALL TVALV22 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  230 CONTINUE
      KTYPL=KTYPL-30
      GO TO (231),KTYPL
  231 CALL TCVAL31 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  240 CONTINUE
      CALL TRLST41 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  250 CONTINUE
      KTYPL=KTYPL-50
      GO TO (251),KTYPL
  251 CALL TPUMP51 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  260 CONTINUE
      KTYPL=KTYPL-60
      GO TO (261,262,400,400,400,400,400,400,269),KTYPL
 261  CALL TRSVK61 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
 262  CALL TRSVK62 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  269 CALL THLX69 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  270 KTYPL=KTYPL-70
      GO TO (271,400,400),KTYPL
  271 CALL TACUM71 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  280 CONTINUE
      KTYPL=KTYPL-80
      GO TO (281),KTYPL
  281 CALL TFILT81 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  290 CONTINUE
      KTYPL=KTYPL-90
      GO TO (291),KTYPL
 291  CALL TTEST91 (D(N1),D(N2),DD(N3),L(N4))
      GO TO 400
  300 CONTINUE
      KTYPL=KTYPL-100
      GO TO (301,302),KTYPL
```

6.1.7 (Continued)

```
301 CALL TACT101 (D(N1),D(N2),DD(N3),L(N4))
    GO TO 400
302 CALL TACT102 (D(N1),D(N2),DD(N3),L(N4))
400 CONTINUE
    IF(INEL.NE.0.OR.IND.GE.NEL) RETURN
    IND=IND+1
    IF(IENTR) 1003,2000,2000
410 WRITE (6,180)
    STOP 6001
420 WRITE(6,190)IND
190 FORMAT(5X,42HTHERE ARE MISSING CONNECTIONS IN COMP NO    ,I5)
    STOP 6001
130 FORMAT ( 35H THE ELEMENT CARDS ARE OUT OF ORDER  )
500 FORMAT(1H0,5X,6HCOMP#,I5,2X,12HINTEGER DATA,2X,16I5,)
510 FORMAT(/,5X,16HREAL DATA CARD # ,I5,2X,8E12.4)
    END
```

6.11 SUBROUTINE TBRAN11

TBRAN11 simulates a frictionless branch with two, three, or four connecting lines. As sketched in Figure 6.11-1 with two lines, TBRAN11 represents a "junction", with three lines it represents a "Y", and with four lines, it represents a "cross". This subroutine calculates the temperature of the fluid within the branch and the temperature of the branch wall.

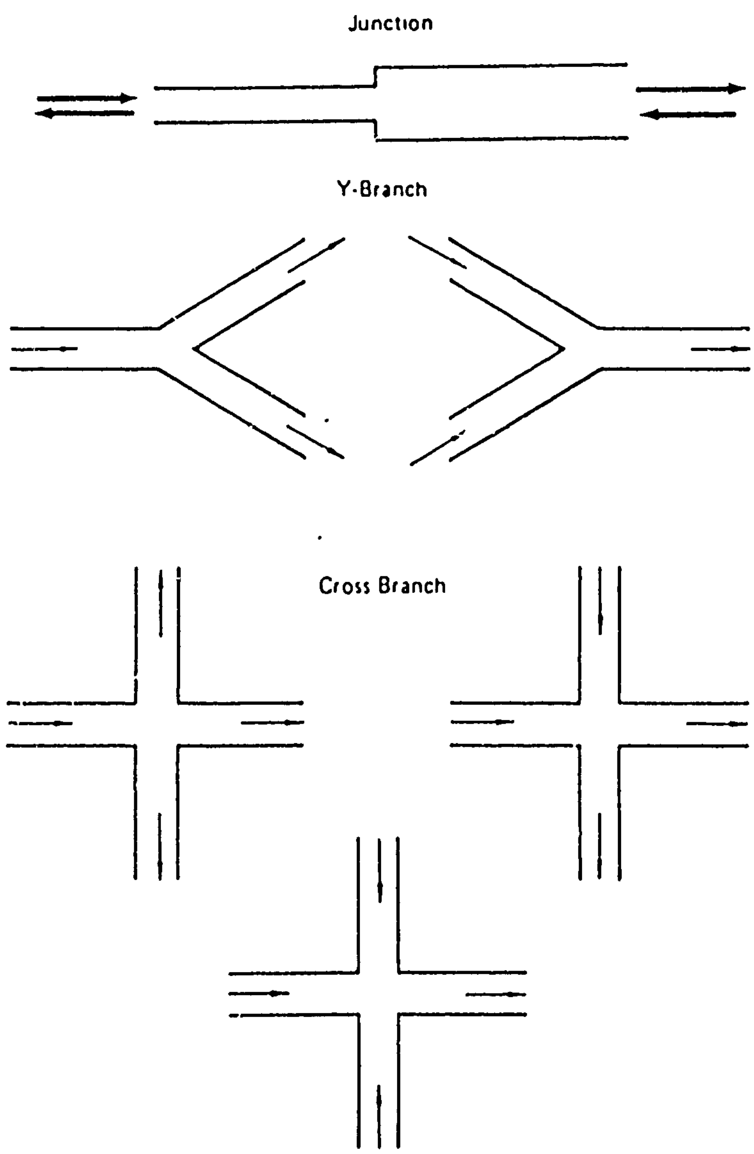


FIGURE 6.11-1
BRANCH CONFIGURATIONS

6.11.1 Math Model

The thermal math model for the branch includes heat transfer to and from either two, three, or four line segments. At least one is an upstream (inlet) line while at least one is a downstream (outlet) line. To familiarize the reader with the branch subroutine, one of the most complex branches "the cross" (with one upstream line segment and three downstream line segments) will be discussed at this time. For this branch ten nodes are considered: five fluid nodes and five wall nodes. The nodal representation is shown in Figure 6.11-2.

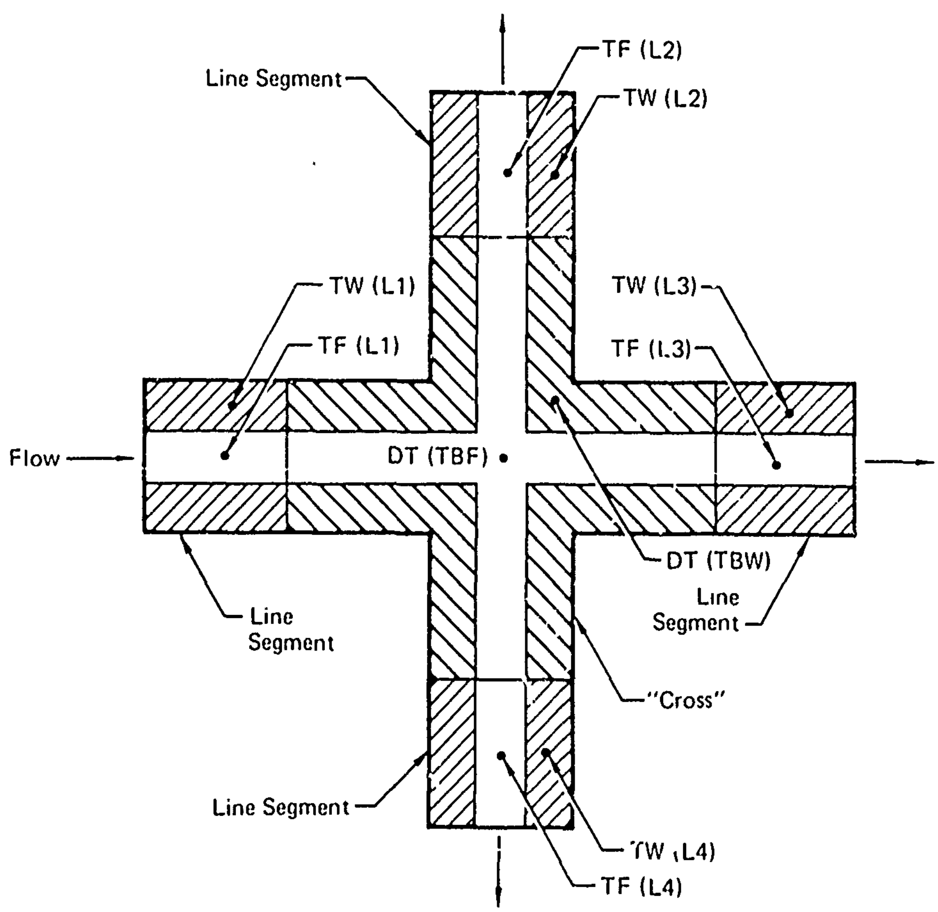


FIGURE 6.11-2
"CROSS" BRANCH AND LINE SEGMENT NODE REPRESENTATION

GP77-0065-7

The temperatures of the upstream line segment wall and fluid nodes are TW(L1) and TF(L1), the temperatures of the branch wall and fluid nodes are DT(TBW) and DT(TBF), and the temperatures of the downstream line segment wall and fluid nodes are TW(L2) and TF(L2), TW(L3) and TF(L3), and TW(L4) and TF(L4). (The branch consists of two nodes, regardless of the number of line connections). Two heat transfer equations are written to solve for DT(TBF) and DT(TBW), using the branch and line segment material properties and dimensions, the atmosphere and structure temperatures external to the branch, TW(L1) through TW(L4) and TF(L1) through TF(L4). One equation is for heat transferred to and from the branch fluid node. The other equation is for heat transferred to and from the branch wall node.

The first equation represents four modes of heat transfer wit' the branch fluid node:

1. conduction with the upstream line segment fluid node

   R1(L1)*(TF(Li)-DT(TBF))

   where R1(L1) is the conduction coefficient between the fluids and is equal to CF/(DXF(L1)/ACF(L1)+DXFB/ACFB+RMT(L1)*DELT/(ACFB**2*RHOIL))

2. convection with the branch node

   B1*(DT(TBW)-DT(TBF))

   where B1 is the convection coefficient and is equal to UFWIL*D(ASFW)

3. heat transfer due to mass transfer into the branch from upstream of the branch

   $\dot{M}$Cp*(TF(L1)-DT(TBF))

   where $\dot{M}$Cp is the flow rate and is equal to Q(L1)*RHOIL*CPFN

These heat transfer terms are combined to produce the equation for heat balance for the branch fluid.

$$\begin{aligned} MCp(DT(TBF)-DT(TBF)_{OLD}) &= R1(L1)*(TF(L1)-DT(TBF)) \\ &+ B1*(DT(TBW)-DT(TBF)) \\ &+ \dot{M}Cp*(TF(L1)-DT(TBF)) \end{aligned} \qquad (1)$$

where MCp is equal to FMASS*CPFN

The second equation represents three modes of heat transfer relative to the branch wall node:

1. Conduction to and from the upstream and downstream line segment walls

   R(LI*(TW(LJ)-DT(TBW))

   where R(LI) is the conduction coefficient and is equal to 1.0/(DXF(LI)/(ACW(LI)*C(LI))+ DXW/(ACBW*CW)) and I is the line number

2a. convection to and from the branch fluid

   B1*(DT(TBF)-DT(TBW))

   where B1 was defined previously

2b. convection to and from the external atmosphere

   B2*(D(TA)-DT(TBW))

   where B2 is the convection coefficient and is equal to D(UAW)*D(ASAW).

3. radiation exchange with the surrounding structure

   $C1P*(D(TST)-(DT(TBW)+460.)^4)$

   where C1P is the radiation coefficient and is equal to SIGMA* EPSION*SHAPF*D(ASAW).

These equations combine to produce the equation for heat balance for the branch wall node:

$$\frac{MCp}{DELT}\ (DT(TBW)-DT(TBW)_{OLD}) = \sum_{J=1}^{4} R\ (LI)*(TW(LI)-DT(TBW))$$

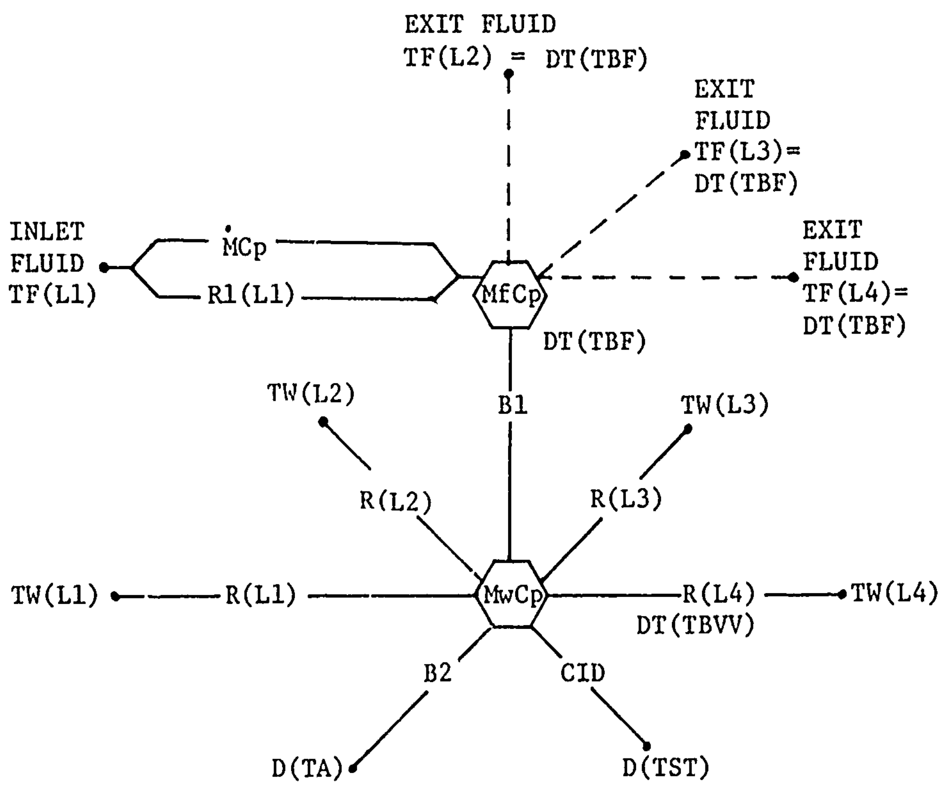
$$+ B1(DT(TBF)-DT(TBW))\qquad (2)$$

$$+ B2(D(TA)-DT(TBW))$$

$$+ C1P*(D(TST)-(DT(TBW)+460.)^4)$$

where MCp is equal to D(BMASS)*CPWN

A thermal model of the above heat transfer terms is shown in Figure 6.11-3. Equations (1) and (2) are combined to solve for DT(TBW) and DT(TBF).



BN = CONVECTION
MCp = STORAGE
$\dot{M}$Cp = FLOW
RN = CONDUCTION
CID - RADIATION

FIGURE 6.11-3

THERMAL MODEL

6.11.2 Assumptions

The following assumptions are made to produce the equations discussed in the previous section.

1. The temperature of the fluid leaving the branch is equal to the branch fluid node temperature, DT(TBF)

2. The branch wall and fluid are each represented by one node only, the entire node is at the same temperature.

3. The temperatures of the atmosphere and structure surrounding the branch are constant.

4. The emissivity of the wall material is constant, .3 for steel

5. The interface conductance between the branch walls and the line walls is infinite.

6. The math model does not incorporate any of the losses which normally occur at junctions which have changes in diameter, flow direction, or flow division.

7. Complete fluid mixing occurs in the fluid volume.

6.11.3 Computational Methods

SECTION 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

SECTION 2000

No thermal or hydraulic inputs or calculation accomplished here.

SECTION 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connections tow, three and four, (L2), (L3) and (L4). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 2x2 matrix is loaded and the mathematical equations are solved for DT(TBF) and DT(TBW) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions assigned to special arrays (TC and TF) in /TRANS/.

6.11.4 Approximations

1. DELTAX is the average value of all possible paths through the branch.

6.11.5 Limitations

The limitations of this subroutine are due to the pressure drop errors. Additional losses can be simulated by adding a pseudo 90 degree elbow or bend to the appropriate line.

6.11.6 Variable Names

Variable	Description	Dimension
A( )	Computational array	--
AAA	Dummy variable	--
ACBW	Cross sectional area of the branch wall	IN^2
ACFB	Cross sectional area of the branch fluid	IN^2
D(ASAW)	External surface area of the branch wall	IN^2
D(ASFW)	Internal surface area of the branch wall	IN^2
B( )	Computational array	--

Variable	Description	Dimension
D(BMASS)	Branch Mass	LB_m
C1P	Radiation coefficient	WATTS/°R^4
CJ	Mechanical Equivalent of Heat	FT-LB_m/WATTS-SEC
CPWN	Specific heat of the branch material	WATTS-SEC/LB_m-°F
CW	Thermal conductivity of the branch material	WATTS/IN-°F
DDD	Dummy variable	--
D(DELTAX)	Average distance fluid travels through branch	IN
DXW	Distance from branch node to interface	IN.
EPSION	Emissivity factor for the branch wall	--
FMASS	Fluid mass in branch	LB_m
I,IERROR	Dummy variables	--
D(ITC)	Initial wall temperature	°F
D(ITF)	Initial fluid temperature	°F
L1,L2	Leg connection addresses	--
R( ),R1( )	Computational array	--
RHOIL	Fluid density	LB_m/IN^3
RMT( )	Mass flow rate of fluid array	LB_m/SEC
SHAPF	Shape factor (wall to surrounding structure)	--
SID	Dummy variable	--
SIGMA	Stifan-Boltzmann constant for radiation	WATTS/IN^2-°R^4
T( )	Computational array	--
D(TA)	Temperature of the surrounding atmosphere	°F
DT(TBF)	Fluid temperature	°F
DT(TBW)	Wall temperature	°F
D(TST)	Temperature of the surrounding structure	°F

Variable	Description	Dimension
D(UAW)	Heat transfer coefficient (surrounding atmosphere to walls)	WATTS/IN^2-°F
UFWIL	Heat transfer coefficient (fluid to walls)	WATTS/IN^2-°F
D(VOLUME)	Volume of fluid inside branch	IN^3

6.11.7 Subroutine Listing

```
      SUBROUTINE TBRAN11 (D,DT,DD,L)
C *** REVISED AUGUST 5, 1975 ***
      DIMENSION D(1),DT(3),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TA(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,DLTIME,REL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,IREL
      COMMON /STEADY/PN(90),QN(90),PLX(90),POLLG(90),QL(90),QA,
     + PUP,PDOWN,NRODE,NLEG,NCPN,TERM,LEGN,ICON,INV,INY,INZ,
     + NUP(99),NDWN(90),NLEGN(90),ILLOAD(30),ILEG(1000)
      COMMON /FLUID/APPLE,CF,CPFN,FTEMP,PROP(13,3)
      INTEGER HAW,ASAW,ASFW,DELTAX,VOLUME,TBW,TBF,
     + TST,TA,BMASS,NCI
      DIMENSION RAT(4),PI(4),A(2,2),B(2),R(4),T(4)
C       D ARRAY VARIABLES
      DATA MTYPE/1/,BMASS/2/,VOLUME/3/,DELTAX/4/,ASAW/5/,ASFW/6/
     + ,HAW/7/,TST/8/,TA/9/,ITF/10/,ITC/11/
C       DT ARRAY VARIABLES
      DATA TBF/1/,TBW/2/
      DATA SIGMA/.349E-11/,SHAPE/.93/,EPSION/.8/,CJ/3.85/
      IF(IENTR)1000,2000,3000
C       VOLUME =TOTAL VOLUME INSIDE BRANCH
C       DELTAX =AVG. OF ALL POSSIBLE ROUTES THROUGH BRANCH
C       ASFW   =SURFACE AREA FLUID TO WALL INSIDE BRANCH
C       ASAW   =SURFACE AREA WALL TO AMBIENT
C       HAW    =HEAT TRANSFER COEFFICIENT WALL TO AMBIENT
C       BMASS    =MASS OF BRANCH WALL(LBS.)
C       MTYPE  =MATERIAL TYPE OF WALL
C       NCI    =NUMBER OF LINE CONNECTIONS
 1000 CONTINUE
      NCI=NC(IND)
      DO 1001 I=1,NCI
      N=L(I)
      TF(N)=D(ITF)
 1001 TC(N)=D(ITC)
      DT(TBF)=D(ITF)
      DT(TBW)=D(ITC)
 1003 D(TST)=(D(TST)+460.)**4
      IF(D(HAW).EQ.0.0) D(HAW)=.0069
      RETURN
 2000 CONTINUE
      RETURN
 3000 L1=L(1)
      NCI=NC(IND)
      KTYPE=D(MTYPE)+.001
      CPWN=PROP(KTYPE,1)
      CW=PROP(KTYPE,3)
      RHOIL=385.4*RHO(TF(L1),P(L1))
      ACFB=D(VOLUME)/D(DELTAX)
      ACBW=D(BMASS)/(RHOIL*D(DELTAX))
      CIP=SIGMA*SHAPE*EPSION*D(ASAW)
```

6.11.7 (Continued)

```
      DXW=D(DELTAX)/2.0
      DXFB=DXW
      L2=L(2)
      SIO=Q(L1)
      AAA=D(VOLUME)/(D(DELTAX))
      DDD=SQRT(AAA*4./PI)
      IF(Q(L1).LT.Q(L2)) Q(L1)=Q(L2)
      UFWIL=UFW(AAA,DDD,ABS(Q(L1)),(TF(L(1))
     ++TF(L(2)))/2.0,(P(L(1))+P(L(2)))/2.0)
C
      TEMP1=TC(L1)
      Q(L1)=SIO
      FMASS=D(VOLUME)*RHOIL
      B1=UFWIL*D(ASFW)
      B2=D(UAW)*D(ASAW)
 3003 NCI=NC(IND)
      DO 3009 I=1,NCI
      DO 3009 J=1,NCI
      A(I,J)=0.0
 3009 B(I)=0.0
      DO 3500 I=1,NCI
      N=L(I)
      RHOIL=386.4*RHO(TF(N),P(N))
      RMT(I)=ABS(Q(N))*RHOIL
      IF(Q(N).LE.0.0) RMT(I)=0.0
      R1(I)=CF/(DXF(N)/ACF(N)+DXFB/ACFB+RMT(I)*DELT/(ACFB**2*RHOIL))
      IF(Q(N).LE.0.0)R1(I)=0.0
      R(I)=1./(DXF(N)/(ACW(N)*C(N))+DXW/(ACBW*CW))
 3450 A(1,1)=A(1,1)+RMT(I)*CPFN+R1(I)
      B(1)=B(1)+(RMT(I)*CPFN+R1(I))*TF(N)
      A(2,2)=A(2,2)+R(I)
      B(2)=B(2)+R(I)*TW(N)
 3500 CONTINUE
      A(1,1)=A(1,1)+FMASS*CPFN/DELT+B1
      A(2,2)=A(2,2)+D(BWMASS)*CPWN/DELT+B1+B2
      B(1)=B(1)+FMASS*CPFN*DT(TBF)/DELT
      B(2)=B(2)+D(BWMASS)*CPWN*DT(TBW)/DELT+B2*
     + D(TA)+CIP*D(TST)-CIP*(DT(TBW)+460.)**4
      A(1,2)=-B1
      A(2,1)=-B1
      CALL SIMULT(A,B,2,IERROR)
      DO 3600 I=1,NCI
      N=L(I)
      T(I)=TF(N)
      TF(N)=B(1)
      IF(Q(N).GE.0.0) TF(N)=T(I)
      TC(N)=B(2)
 3600 CONTINUE
      DT(TBF)=B(1)
      DT(TBW)=B(2)
      RETURN
      END
```

6.21 SUBROUTINE TVALV

TVALV21 simulates a simple two-way valve with a valve position vs time input. A typical valve is sketched in Figure 6.21-1. This subroutine calculates the valve wall, piston, and fluid temperatures.

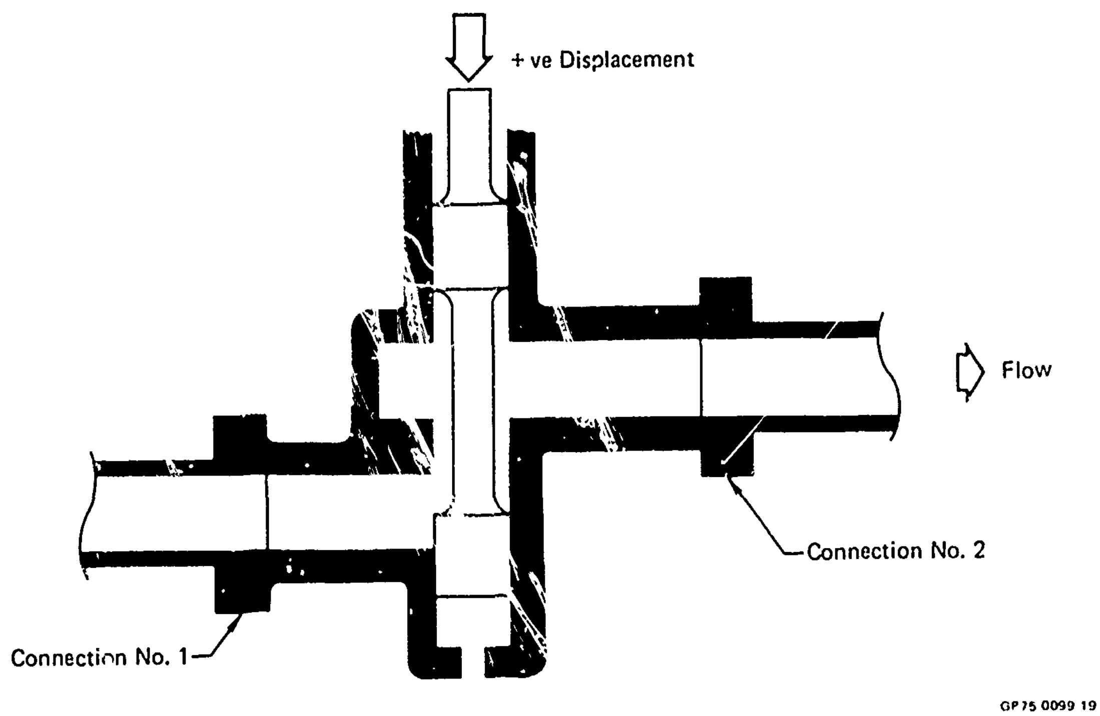


FIGURE 6.21-1
TYPE NO. 21 TWO-WAY VALVE

6.21.1 MATH MODEL

The thermal math model for the two-way valve includes heat transfer to and from two connecting line segments, one upstream and one downstream. Seven nodes are considered: three fluid nodes, three wall nodes, and one piston node (as shown in Figure 6.21-2). The temperatures of the upstream line segment wall and fluid nodes are TW(L1) and TF(L1).

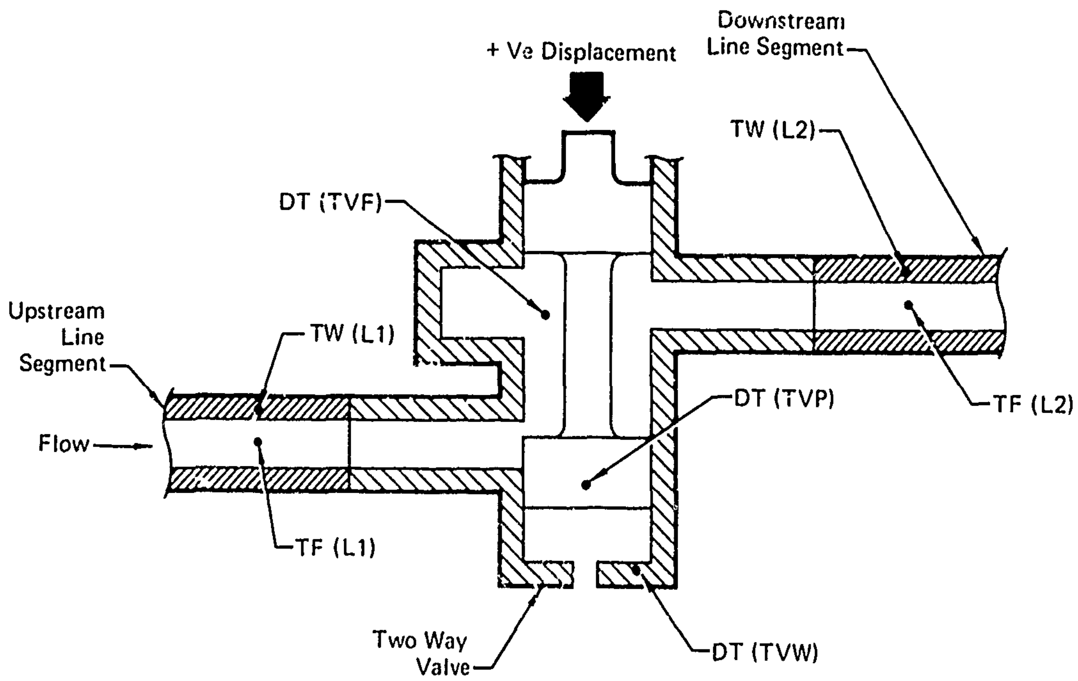


FIGURE 6.21-2
VALVE AND LINE SEGMENT NODE REPRESENTATION

GP77-0065-9

The temperatures of the valve wall and fluid nodes are DT(TVW) and DT(TVF). The temperatures of the downstream segment line wall and fluid nodes are TW(L2) and TF(L2), and the temperature of the piston is denoted by DT(TVP). Three heat balance equations are written to solve for DT(TVF), DT(TVP), and DT(TVP), using the valve and line segment material properties and dimensions, the atmosphere and structure temperatures external to the valve, and TW(L1), TW(L2), and TF(L1) (Note: TF(L2)=DT(TVF), see assumptions). One equation is a heat balance for the valve fluid node. The second equation is a heat balance for the valve wall node. The third equation is a heat balance for the valve piston.

The first equation represents four modes of heat transfer relative to the valve fluid node.

1. Conduction to or from the upstream line segment fluid node

   R3*(TF(L1)-DT(TVF))

   where R3 is the conduction coefficient and is equal to CF/ (DXV/D(ACVF)+DXF(L1)/ACF(L1)+RMFL1*DELT/(D(ACVF)**2*RHOIL)), and RMFL1 is equal to Q(L1)*RHOIL.

2. a. convection to or from the valve walls

B1*(DT(TVW)-DT(TVF))

where B1 is the convection coefficient and is equal to UFWIL*D(ASFV).

b. convection to or from the piston

B2*(DT(TVP)-DT(TVF))

where B2 is equal to USWIL*D(ASFP)

3. Heat transfer due to mass transfer into the valve node from the upstream line fluid segment.

$\dot{M}$Cp*(TF(L1)-DT(TVF))

where $\dot{M}$Cp is the flow rate coefficient, and is equal to Q(L1)*RHOIL*CPFN

4. Heat addition due to a pressure drop across the valve

$\dot{M}$Cp*DCAPT1

where DCAPT1 was described in the technical summary and is equal to

(1.0/RHOIL)*(P(L1)-P(L2))/(CJ*CPFN)

The heat transfer modes are combined to produce an equation for the heat balance for the valve fluid node:

$$\frac{MCp}{DELT}(DT(TVF)-DT(TVF)_{OLD})=R3*(TF(L1)-DT(TVF)) \\ +B1*(DT(TVW)-DT(TVF)) \\ +B2*(DT(TVP)-DT(TVF) \\ +\dot{M}Cp*(TF(L1)-DT(TVF)) \\ +\dot{M}Cp*DCAPT1 \qquad (1)$$

where MCp is equal to FMASS*CPEN

The second equation represents three modes of heat transfer relative to the valve wall node.

1. Conduction to or from the upstream and downstream line segment wall nodes

$$RI*(TW(LI)-DT(TVW))$$

where RI=1.0/(DXF(LI)/(ACW(LI)*C(LI))+DXV/(ACVW*CV)), and I=1 for the upstream line segment and 2 for downstream line segment

2. a. convection to or from the fluid in the valve

$$B1*(DT(TVF)-DT(TVW))$$

where B1 was defined previously

b. convection to or from the external atmosphere

$$B3*(D(TA)-D(TVW))$$

where B3 is the convection coefficient, and is equal to D(UAV)*D(ASAV)

3. radiation exchange with the surrounding structure

$$CIP*(D(TST)-(DT(TVW)+460.)**4)$$

where CIP is the radiation coefficient, and is equal to SIGMA*EPSION*SHARF*D(ASAV)

These heat transfer modes are combined to produce an equation for the heat balance for the valve wall node:

$$\frac{MCp}{DELT}*(DT(TVW)-DT(TVW)_{OLD}) = \begin{aligned}&R1*(TW(L1)-DT(TVW))\\&+R2*(TW(L2)-DT(TVW))\\&+B1*(DT(TVF)-DT(TVW))\\&+B3*(D(TA)-DT(TVW))\\&+CIP*(D(TST)\\&-CIP*((DT(TVW)+460.)**4)\end{aligned} \qquad (2)$$

where MCp is equal to D(VMASS)*CPVW

The third equation represents one mode of heat transfer relative to the valve piston node.

1. convection to or from the valve fluid node

$$B2*(DT(TVF)-DT(TVP))$$

where B2 was defined previously

The third equation for the heat balance for the piston node is:

$$\frac{MCp}{DELT}*(DT(TVP)-DT(TVP)_{OLD}) = B2*(DT(TVF)-DT(TVP)) \qquad (3)$$

where MCp is equal to D(PMASS)*CPPN

A thermal model of the above heat transfer terms for the two-way valve is shown in Figure 6.21-3. Equations (1) thru (3) are solved for the appropriated temperatures.

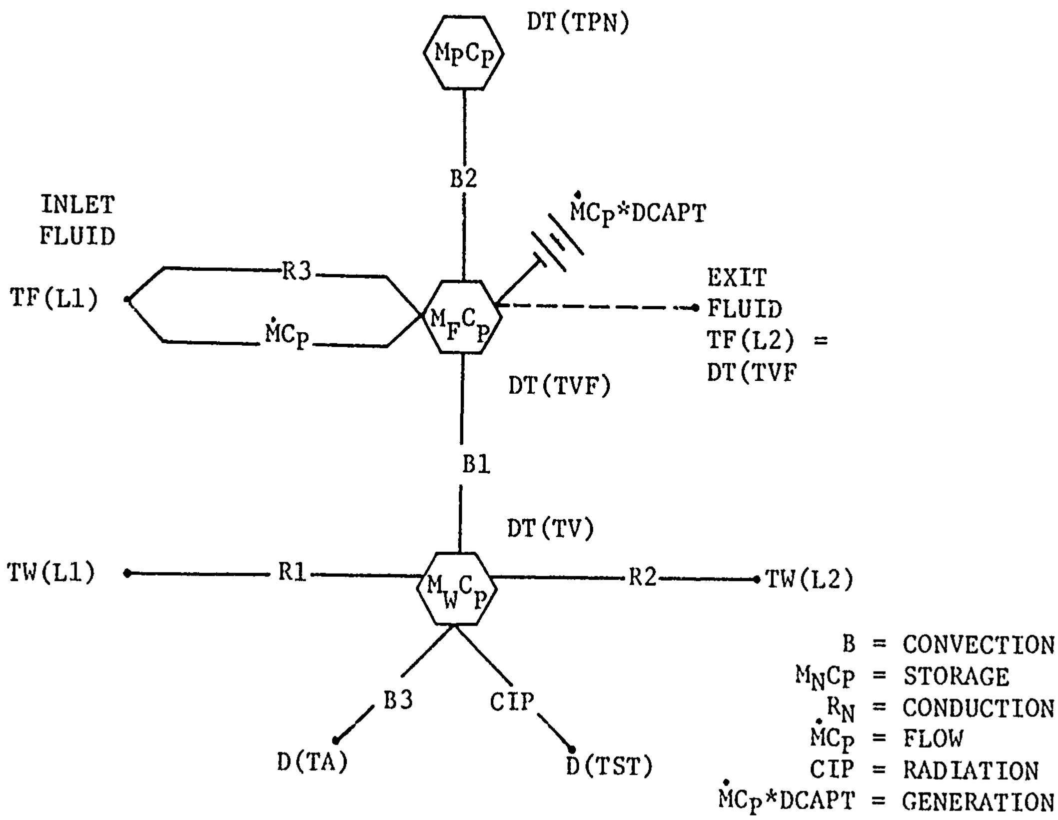


FIGURE 6.21-3

THERMAL MODEL

In the hydraulic model the valve is considered to be a simple orifice.

The basic equation for flow through an orifice is used to define flow through the two-way valve.

$$Q=AREA*CD*(2*(P1-P2)/RHO)^{1/2}$$

where AREA=area of valve orifice (IN^2)

Cd=discharge coefficient

RHO=fluid density $LB\text{-}SEC^2/IN^4$

Q=flow (CIS)

P1=inlet pressure (PSI)

P2=outlet pressure (PSI)

The pressure drop due to [illegible] given by Equation (5)

PUP=PUP-QA*Q1*(COEF*RHO(TF(L(1)),PUP)

where

PUP = upstream pressure (PSI)

QA = magnitude of flow (CIS)

Q1 = flow rate (CIS)

COEF = constant coefficient $(\frac{1}{IN4})$

RHO( ) = fluid density $(LB\text{-}SEC^2/_{IN}{}^4)$

The constant coefficient is made up of the valve opening area and discharge coefficient. When the valve is closed COEF equals zero.

6.21.2 ASSUMPTIONS

1. The temperature of the fluid leaving the valve is equal to the valve fluid node temperature, DT(TVF).
2. The pressure drop across the valve raises the temperature of the fluid in the valve.
3. The temperatures of the atmosphere and structure surrounding the valve remain constant.
4. The emissivity of the wall material is a constant, .3 for steel.
5. The interface conductance between the valve and line walls is infinite.
6. The hydraulic math model assumes a square law characteristic and a constant discharge coefficient for the complete flow range, which in practice is not correct. At very low flows the pressure drop tends toward a linear characteristic, and the discharge coefficient varies.
7. The pressure drop due to the fittings is assumed to be much smaller than the valve pressure drop and they are ignored in the computation.
8. Complete mixing of the fluids is assumed.

6.21.3 COMPUTATIONAL METHODS

Section 1000 - The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

Section 2000 - A call to INTERP is made to derive the valve opening from the input data which includes a table of valve position versus time. A first order interpolation is used in this derivation, second or higher order interpolations will cause unintended valve motion.

Once the valve opening is established, the valve area is calculated, and the valve pressure drop is determined using Equation (5).

Section 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connection line two (L2). During the calculation, the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 3 x 3 matrix is loaded and the mathematical equations are solved for DT(TVF), DT(TVW) and DT(TVP) and stored in the B computational array. The calculated values are assigned to their proper storage locations and boundary conditions assigned to special arrays (TF and TC) in COMMON/TRANS/ and distributed throughout the entire program.

6.21.4 APPROXIMATIONS

1. The shape factor is 0.96 (described in the Technical Summary).
2. Distances from nodes to interfaces are approximated.
3. The coefficient of heat transfer between the wall and external atmosphere is approximated.

6.21.5 LIMITATIONS - The computation is limited to a linear valve area versus position relationship. This apparent limitation can be overcome by inputting a nonlinear valve position versus time relationship which can produce any desired area versus time.

The constant discharge coefficient is also a limitation but since the changes in discharge coefficients depend on the particular valve configuration; this limitation is not easily overcome.

If the valve slot width and discharge coefficient are input as one, then the valve position table becomes a table of the product of valve area times the discharge coefficient versus time. The combined effects of area and discharge coefficient can then be inputted.

6.21.6 VARIABLE LISTING

Variable	Description	Dimension
AAA	Dummy variable	-
D(ACVF)	Cross sectional area of the valve fluid	IN^2
ACVW	Cross sectional area of the valve walls	IN^2
D(ASAV)	Outer surface area of valve	IN^2
D(ASFP)	Surface area of the piston	IN^2
D(ASFV)	Internal surface area of the valve walls	IN^2
A1,A2,A3,A4	Dummy variables	-
B( )	Computational array	-
CID	Dummy variable	-
CIP	Radiation coefficient	WATTS/°R^4
CJ	Mechanical equivalent of heat	FT-LB_m/WATTS-SEC
COEFF	Dummy variable	1/IN^4
CPPN	Specific heat of the valve piston	WATTS-SEC/°F
CPVW	Specific heat of the valve walls	WATTS-SEC/°F
CV	Thermal conductivity of the valve walls	WATTS/IN-°F
DCAPT1	Heat added to fluid due to a pressure change	°F

Variable	Description	Dimension
DCOEF	Discharge coefficient	-
DDD	Dummy variable	-
D(DELTAX)	Distance fluid travels through valve	IN
DXV	Distance from valve wall node to interface with line segment	IN
EPSION	Emissivity of the valve walls	-
FMASS	Mass of the fluid in the valve	LB_m
IERR,IERROR	Dummy variables	-
D(ITF)	Initial temperature of the fluid	°F
D(ITV)	Initial temperature of the valve walls	°F
L1,L2	Leg connection addresses	-
D(MTYPE)	Valve material type	-
D(PERC)	Percent of heat added to fluid due to pressure drop	-
D(PMASS)	Mass of the piston	LB_m
D(PTYPE)	Piston material type	-
RHOIL	Fluid density	LB_m/IN^3
RHOV	Density of the valve walls	LB_m/IN^3
RMFL1	Entering fluid mass flow	LB_m/SEC
RMFL2	Exiting fluid mass flow	LB_m/SEC
R1,R2,R3,R4	Dummy variables	-
SHAPF	Shape factor valve walls to surroundings	-
SIGMA	Stefan-Boltzman radiation constant	$WATTS/IN^2\text{-}°R^4$
D(TA)	Temperature of the surrounding ambient atmosphere	°F
D(TST)	Temperature of the surrounding structure	°F
D(TFV)	Fluid temperature in the valve	°F

Variable	Description	Dimension
DT(TVP)	Piston temperature	°F
DT(TVW)	Wall temperature of the valve	°F
D(UAV)	Heat transfer coefficient (surrounding ambient to valve)	WATTS/IN^2-°F
UFWIL	Heat transfer coefficient (fluid to valve walls)	WATTS/IN^2-°F
VALY	Valve piston position	IN
D(VMASS)	Mass of the valve walls	LB_m
D(VOLUME)	Internal volume of the valve	IN3
D(WIDTH)	Valve slot width	IN

For variables in common refer to Paragraph 3.3.

6.21.7 Subroutine Listing

```
      SUBROUTINE TVALV21 (D,DT,DD,L)
C *** REVISED AUGUST 20,1976 ***
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INL
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,NNODE,ILEG,ICPW,TERM,
     +  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NLEGN(90),
     +ILEGAP(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFW,FTEMP,PROP(13,3)
      DIMENSION A(3,3),B(3)
      INTEGER UAV,ASAV,TST,TA,VMASS,DELTAX,TVW,TVF,PERC,ASFV,WIDTH,
     + DCOLF,ACVF,PTYPE,PMASS,ASFP,TVP,VOLUME
C     D ARRAY VARIABLES
      DATA MTYPE/1/,VMASS/3/,DELTAX/5/,ASAV/7/,ASFV/8/,
     +UAV/11/,PERC/12/,TST/13/,TA/14/,ITF/15/,ITV/16/,VOLUME/6/,
     + WIDTH/17/,DCOLF/18/,ACVF/10/,PTYPE/2/,PMASS/4/,ASFP/9/
C      DT ARRAY VARIABLES
      DATA TVW/1/,TVF/2/,TVP/3/
      DATA SIGMA/.349E-11/,SHAPF/.96/,EPSION/.8/,CJ/8.85/
      IF(IENTR) 1000,2000,3000
 1000 CONTINUE
      KTYPE=D(MTYPE)+.001
      NTYPE=D(PTYPE)+.001
      RHOV=PROP(KTYPE,2)
      CPPN=PROP(NTYPE,1)
      CPVW=PROP(KTYPE,1)
      CV=PROP(KTYPE,3)
C     DELTAX =DISTANCE FLUID TRAVELS THROUGH VALVE
C     VMASS     =MASS OF THE VALVE(#)
C     ASAV    =SURFACE AREA VALVE TO AMBIENT
C     ACVW      =CROSS SECTIONAL AREA OF VALVE WALL
C     ASFV  =WETTED SURFACE AREA, FLUID TO VALVE,
      D(TST)=(D(TST)+460.)**4
      L1=L(1)
      L2=L(2)
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
      TC(L1)=D(ITV)
      TC(L2)=D(ITV)
      DT(TVW)=D(ITV)
      DT(TVP)=D(ITF)
      DT(TVF)=D(ITF)
      IF(D(UAV).EQ.0.0) D(UAV)=.0069
C     L(3)= NO. OF X VARIABLES
C     L(4) = START OF Y VARIABLES
      LL=(L(3)+7)/8
      L(4)=25+LL*8
      RETURN
```

6.21.7 (Continued)

```
 2000 CONTINUE
      IF(ICON.NE.1) RETURN
      CALL INTERP(TIME,D(25),D(L(4)),10,L(3),VALY,IERR)
      IF(VALY.LE.0.0)GO TO 2010
      COEF=.5/(D(DCOEF)*VALY*D(WIDTH))**2
      PUP=PUP-QA*Q1*COEF*RHO(TF(L(1)),PUP)
      RETURN
 2010 PUP=PUP-10E6*Q1
      RETURN
 3000 CONTINUE
      L1=L(1)
      L2=L(2)
      AAA=D(ACVF)
      DDD=SQRT(AAA*4./PI)
      UFOIL=UFM(AAA,DDD,ABS(Q(L1)),TF(L1),P(L1))
      TEMP1=TC(L1)
      RHOIL=336.4*RHO(TF(L1),P(L1))
      FMASS=D(VOLUME)*RHOIL
      ACVW=D(VMASS)/(RHOV*D(DELTAX))
      RMFL1=ABS(Q(L1))*RHOIL
      RMFL2=ABS(Q(L2))*RHOIL
      A3=RMFL1*CPFN
      B2=D(ASFP)*UFOIL
      A5=RMFL2*CPFN
      CIP=SIGMA*EPSTOT*SHAPE*D(ASAV)
      DXV=D(DELTAX)/2.3
      B1=UFOIL*D(ASAV)
      B3=D(UAV)*D(ASAV)
      DCAPT1=((1.0/RHOIL)*ABS(P(L1)-P(L2)))/(CJ*CPFN)
      R1=1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXV/(ACVW*CV))
      R2=1.0/(DXF(L2)/(ACW(L2)*C(L2))+DXV/(ACVW*CV))
      R3=CF/(DXV/D(ACVF)+DXF(L1)/ACF(L1)+RMFL1*
     + DELT/(D(ACVF)**2*RHOIL))
      R4=CF/(DXV/D(ACVF)+DXF(L2)/ACF(L2)+RMFL2*
     + DELT/(D(ACVF)**2*RHOIL))
      IF(Q(L1).GT..001) GO TO 3003
      IF(ABS(Q(L1)).LT..001) GO TO 3006
      L2=L(1)
      L1=L(2)
      R3=0.0
      A3=0.0
      CID=R1
      R1=R2
      R2=CID
      R42=0.0
      GO TO 3009
 3006 R3=D(ACVF)*D(UAV)
      R42=R3
      R4=0.0
      GO TO 3009
```

6.21.7 (Continued)

```
 3003 R4=0.0
      A5=0.0
      R42=0.0
 3009 CONTINUE
      A(1,1)=B1+A3+B2+R3+R4+FMASS*CPFN/DELT+R42+A5
      A(1,2)=-B1
      A(1,3)=-B2
      A(2,1)=-B1
      A(2,2)=D(VMASS)*CPVW/DELT+R1+R2+B1+B3
      A(2,3)=0.
      A(3,1)=-B2
      A(3,2)=0.
      A(3,3)=D(PMASS)*CPPN/DELT+B2
      B(1)=(A3+R4+R3+A5)*TF(L1)+RMFL1*CPFN*D(PLRC)*DCAPT1
     + +R42*TF(L2)+FMASS*CPFN*DT(TVF)/DELT
      B(2)=D(VMASS)*CPVW*DT(TVW)/DELT+R1*TW(L1)+R2*
     + TW(L2)+B3*D(TA)+CIP*D(TST)-CIP*(DT(TVW)+460.)
     + **4+RMFL1*CPFN*(1.-D(PLRC))*DCAPT1
      B(3)=D(PMASS)*CPPN*DT(TVP)/DELT
      CALL SIMULT(A,B,3,IERROR)
      TF(L2)=B(1)
      TC(L1)=B(2)
      TC(L2)=B(2)
      DT(TVF)=B(1)
      DT(TVW)=B(2)
      DT(TVP)=B(3)
      RETURN
      END
```

6.22 SUBROUTINE TVALV22

Subroutine TVALV22 describes a generalized four-way valve which can be a segment of a servo actuator (connected to it by lines) or control any servo or utility type function, as shown in Figure 6.22-1.

The valve position is derived from input data, tabulated versus time. The actual position is obtained using linear interpolation between the nearest two data inputs.

The valve orifice areas are derived using a variable law which can effectively describe leakage, open center, underlap and overlap conditions, with various pressure gains.

A fluid volume in the valve is associated with each connecting line. The subroutine calculates the temperatures of the four fluid volumes and the temperature of the valve wall.

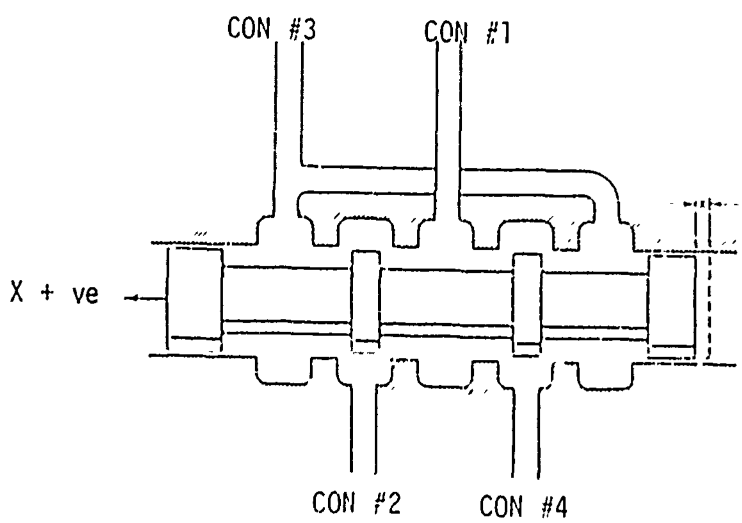


FIGURE 6.22-1

TYPE NO. 22 FOUR-WAY VALVE

6.22.1 Math Model

The thermal math model includes heat transfer to and from four connecting line segments, as shown in Figure 6.22-2. The valve is

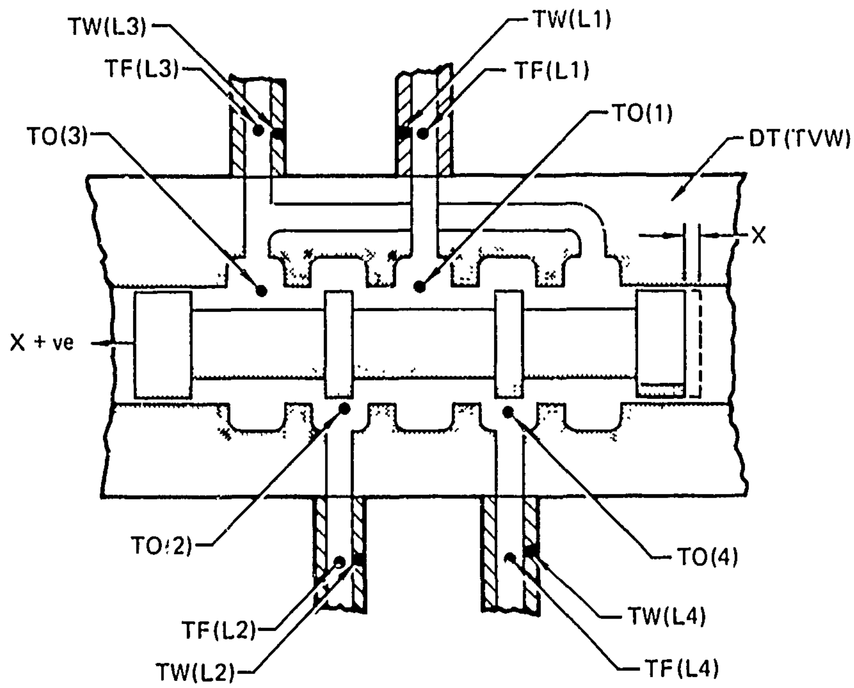


FIGURE 6.22-2
VALVE NODE REPRESENTATION

GP77-0065-14

represented by four fluid nodes and one wall node, and each line is represented by one fluid and one wall node, DT(TVW). The temperatures of the valve fluid volume downstream or upstream of each connecting line segment are TO(1), TO(2), TO(3), and TO(4). The temperatures of the fluid and wall nodes of each connecting line segment are TF(L1) through TF(L4) and TW(L1) through TW(L4). In a four way valve the fluid can enter two lines and can leave two lines. To understand the math model let us consider the case where the fluid enters connection 1, into volume 1, flows to volume 2, leaves connection 2, flows through part of the system, reenters connection 3, into volume 3, flows to volume 4, and leaves connection 4. These paths are shown in Figure 6.22-3. The fluid in the four nodes are affected by the fluid in two other nodes. As shown, the fluid in volume 1 is affected due to losses to volume 3 and gain from volume 4. The other fluid volumes are affected similarly.

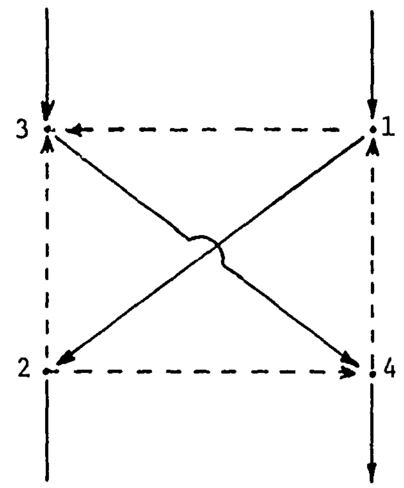


FIGURE 6.22-3

FLUID FLOW PATHS

Equations can be written for fluid node 1 through 4, and for the valve wall node.

The first four equations represents four modes of heat transfer to and from the volume fluid nodes.

1. Heat transfer due to mass transfer into the volume from upstream of the volume

RMF(1) * CPFN (TF(L1) - TO(1)) for volume 1

RMF(3) * CPFN (TF(L1) - TO(3)) for volume 3

zero for volumes 2 and 4

where RMF(1) is equal to Q(L1) * RHO1l, etc.

2. Conduction to and from the fluid node in the connecting line segment.

R(1) * (TF (L(1) - TO(1)) for volume 1

R(3) * (TF (L(3) - TL(3)) for volume 3

where R( ) is the conduction coefficient

and is equal to CF/(D X F (L(I) / ACF (L(I)) +

D X VF / D (ACVF) + RMF (I) * DELT/ (D(ACVF)**

2 * RHOIL)) where I = 1 for line 1 and 3 for line 3

This term is zero for nodes 2 and 4

1b. Heat transfer due to mass transfer into the volume from the other three fluid volumes.

RMD(4) * CPFN (TO(4) - TO(1)) for volume 1

RMD(1) * CPFN (TO(1) - TO(2)) for volume 2

RMD(5) * CPFN (TO(1) - TO(3))
RMD(2) * CPFN (TO(2) - TO(3)) for volume 3

RMD(6) * CPFN (TO(2) - TO(4))
RMD(3) * CPFN (TO(3) - TO(4)) for volume 4

where RMD(1) * CPFN is equal to QI(1) * CPFN * RHOIL etc.

QI(5) and QI(6) are the leakage flows between volumes one and three, and two and four, respectively, and are equal to zero

4. Convection with the wall

```
U(4) * A(1) * (DT(TVW) - TO(1)) for volume 1

U(1) * A(2) * (DT(TVW) - TO(2)) for volume 2

U(5) * A(3) * (DT(TVW) - TO(3))    for volume 3
U(2) * A(3) * (DT(TVW) - TO(3))

U(3) * A(4) * (DT(TVW) - TO(4))    for volume 4
U(6) * A(4) * (DT(TVW) - TO(4))
```

Where U(3) is a heat transfer coefficient for volume flow rate RMD(3) and A(4) is the surface area of volume four and the valve walls equal to D(ASFV) (or really D(ASFV)/4.0).

5. Heat addition due to a pressure drop experienced by the fluid.

```
DCAPT (1) * RMD(1) * CPFN   for volume 2

DCAPT (2) * RMD(2) * CPFN   for volume 3
DCAPT (5) * RMD(5) * CPFN

DCAPT (3) * RMD(3) * CPFN   for volume 4
DCAPT (6) * RMD(6) * CPFN

DCAPT (4) * RMD(4) * CPFN   for volume 1
```

Where DCAPT(2) is the heat add to volume two due to the pressure drop between line one and line two equal to (1.0/RHOIL) * ABS (P(L1) - P(L2)) / (CJ * CPFN)

The heat transfer terms combine to produce four equations for heat balance for the valve four fluid nodes:

For volume one,

$$\begin{aligned}\frac{MCp}{DELT} * (TO(1)-TO(1)_{OLD}) = {} & RMF(1)*CPFN*(TF(L1)-TL(1))+ \\ & R(1)*(TF(L1)-TO(1))+RMD(4)*CPFN* \\ & (TO(4)-TO(1))+U(4)*A(1)*(DT(TVW) \\ & -TO(1))+DCAPT(4)*RMD(4)*CPFN\end{aligned} \tag{1}$$

For volume two,

$$\begin{aligned}\frac{MCp}{DELT} * (TO(2)-TO(2)_{OLD}) = {} & RMD(1)*CPFN*(TO(1)-TO(2))+ \\ & U(1)*A(2)*(DT(TVW)-TO(2))+ \\ & RMD(1)*CPFN+DCAPT(1)\end{aligned} \tag{2}$$

For volume three,

$$\begin{aligned}\frac{MCp}{DELT} * (TO(3)-TO(3)_{OLD}) = {} & RMF(3)*CPFN*(TF(L3)-TO(3))+ \\ & R(3)*(TF(L3)-TO(3)+RMD(2)*CPFN* \\ & (TO(2)-TO(3))+RMD(5)*CPFN*(TO(1)- \\ & TO(3))+U(2)*A(3)*(DT(TVW)- \\ & TO(3))+U(5)*A(3)*(DT(TVW)-TO(3))+ \\ & DCAPT(2)*RMD(2)*CPFN+DCAPT(5)* \\ & RMD(5)*CPFN\end{aligned} \tag{3}$$

For the fourth fluid volume,

$$\begin{aligned}\frac{MCp}{DELT} (TO(4) -TO(4)_{OLD}) = {} & RMD(3)*CPFN(TO(3)-TO(4))+RMD(6)* \\ & CPFN*(TO(2)-TO(4))+U(3)*A(4)* \\ & (DT(TVW)-TO(4))+U(6)*A(4)*(DT(TVW)- \\ & TO(4))+DCAPT(3)*RMD(3)*CPFN+DCAPT(6) \\ & *RMD(6)*CPFN\end{aligned} \tag{4}$$

where all MCp's are equal to FMASS*CPFN

The fifth equation represents three modes of heat transfer relative to the valve wall node;

1(a) Convection with all six internal volume flow rates

```
U(4)*A(1)*(TO(1)-DT(TVW))+U(1)*A(2)*
(TO(2)-DT(TVW))+(U(2)+U(5))*A(3)*
(TO(3)-DT(TVW))+(U(3)+U(6))*A(4)
*(TO(4)-DT(TVW))
```

where the U's and A's were defined previously

1(b) Convection with the external atmosphere

```
D(UAV)*D(ASAV)*(D(TA) -DT(TVW)).
```

D(UAV)*D(ASAV) is the convection heat transfer coefficient between the wall and the atmosphere.

2. Conduction with all four connecting lines

```
RI*(TW(L(I))-DT(TVW))
```

where RI is the conduction coefficient between line wall segment I and the valve wall and is equal to,

```
1.0/(DXF(L(I)/(ACS(LI)*C(LI)+DXVW/(ACVW*CV))
```

and I designates the Ith (connection) number.

3. Radiation exchange with the surrounding structure is

```
CIP*(D(TST)-(DT(TVW)+460.)**4).
```

CIP is the radiation coefficient equal to SIGMA*SHAPF*EPSION*D(ASAV).

These heat transfer reactions combine to produce an equation for the heat balance in time DELT, for the valve wall node.

$$\frac{MCp}{DELT} * (DT(TVW)-DT(TVW)_{OLD}) = U(4)*A(1)+(TO(1)-DT(TVW))+ U(1)*(A2)*(TO(2)-DT(TVW))+ (U(2)*U(5))*A(3)*(TO(3)- DT(TVW))+(U(3)+U(6))* A(4)*(TO(4)-DT(TVW))+R1* (TW(L1)-DT(TVW))+ \sum_{I=2}^{4} (RI* (TW(LI)-DT(TVW))+D(UAV)*D(ASAV)* (D(TA)-DT(TWV))+CIP*(D(TST))- CIP*(DT(TVW)+460.)**4 \tag{5}$$

where MCp is equal to D(VMASS)*CPVW

A thermal model of the above heat transfer terms for the valve is shown in Figure 6.22-4. Equations (1) through (5) are solved for the appropriate temperatures.

CALCULATION OF ORIFICE AREAS

Spool and sleeve type servo valves can have a variety of orifice configurations, the most common of which are round holes and square or rectangular slots.

Because of radial clearances between the spool and sleeve, there is usually a leakage flow when the orifice is completely covered. This leakage tends to round the ends of what would otherwise be a linear flow versus spool position characteristic. In order to simplify the flow calculations, we have assumed that the valve area is an equivalent area which allows the orifice equations to be used at all times.

To obtain the valve area, for a given valve position, a characteristic curve is generated based on the projected cut-off position, the projected max open position and the max valve area.

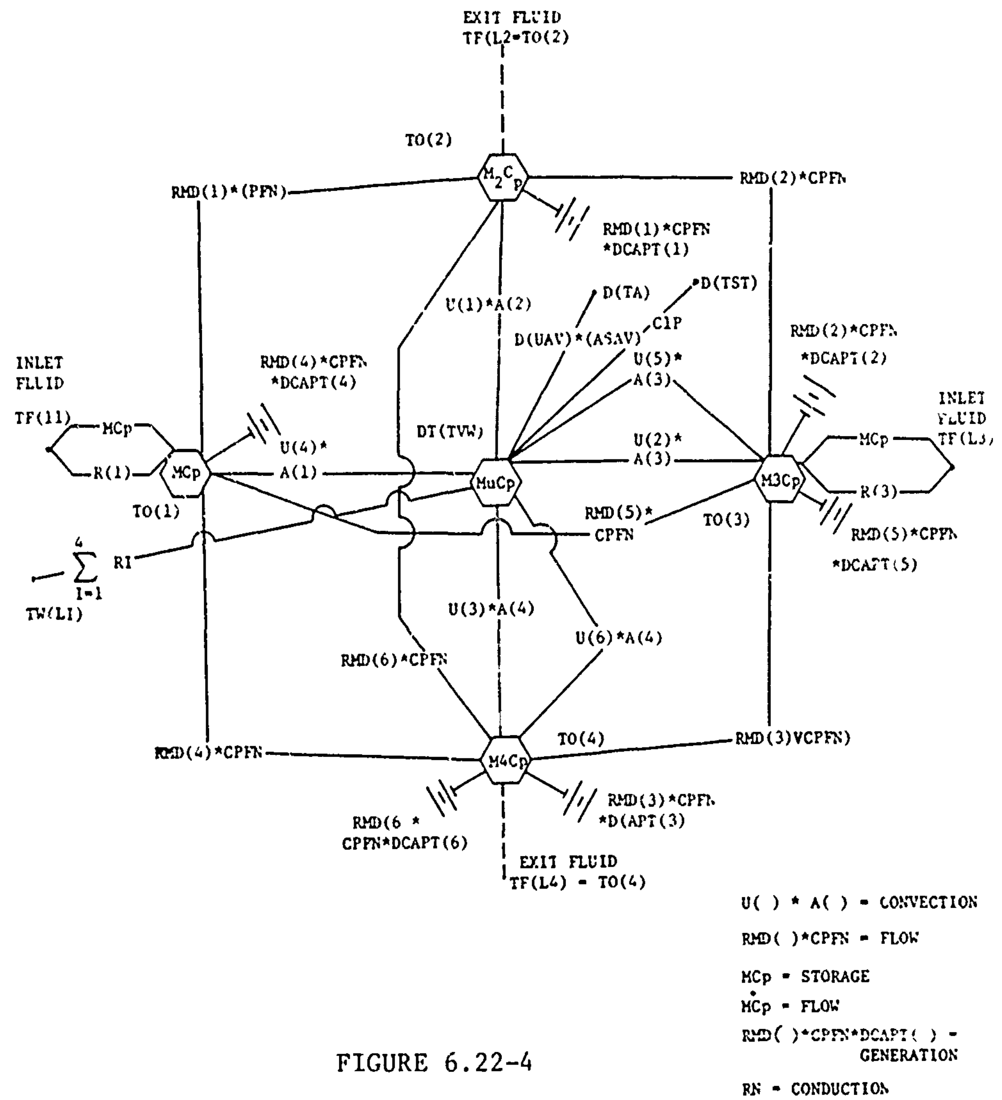


FIGURE 6.22-4

THERMAL MODEL

The maximum valve area is combined with the discharge coefficient and the SQRT(2/RHO) to give an orifice resistance. The formula used to generate the characteristic curve is

X = (.5 + XT/ (1 + ABS(XT*2)**Y)II(1/Y)

where $0 \leq X \leq 1.0$ for all values of XT.

when Y is large, ie. 64, the characteristic curve is almost a straight line between projected cut-off and projected maximum opening. Family of curves for different values of Y is shown in Figure 6.22-5.

6.22.2 Assumptions

1. The piston is not considered a node since it is only a storage device and becomes the same temperature as that of the fluid.
2. The atmospheric and structure temperatures remain constant.
3. All four fluid volumes are the same volume, each 1/4 of the total volumes inside the valve.
4. The interface conductance between the lines and the valve walls is infinite, since the limiting condition is conduction in the line itself.
5. The emissivity of the valve walls remain constant at .3, which is the value for steel.
6. The temperatures of the fluid leaving the valve are equal to the fluid node temperatures calculated, TO(2) to TO(4).
7. Complete fluid mixing occurs in the fluid volume.

6.22.3 Computational Methods

SECTION 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

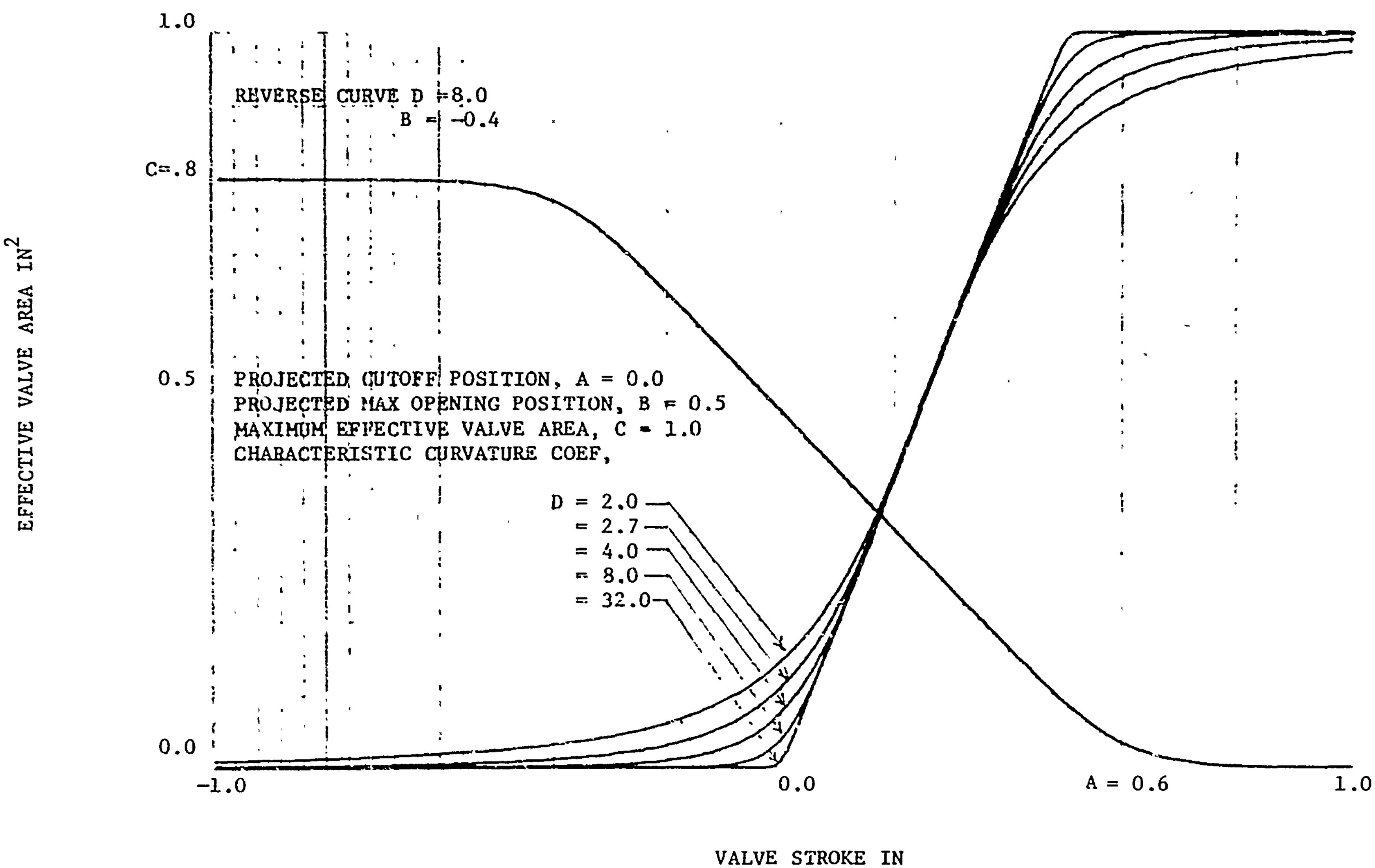


FIGURE 6.22-5
EFFECTIVE VALVE AREA CHARACTERISTICS

SECTION 2000

A test is made to determine if a port is not dimensioned, which can happen if it is a 3-way or 2-way valve. If the area is zero or XT is zero, XT is set to .0001 to prevent the computation blowing up when XT is used in the denominator.

The steady state section is straight forward, the valve pressure drop is subtracted from the upstream pressure PUP for each call to a particular connection.

SECTION 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connection line four (L4). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 5x5 matrix is loaded and the mathematical equations are solved for TO(1) thru TO(4) and DT(TVW) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions assigned to arrays (TC and TF) in common /TRANS/.

6.22.4 Approximations

1. The valve wall is only one node and is all at the same temperature.
2. The heat transfer coefficients are calculated on the velocity in the valve, by the function subroutine, UFW.
3. The heat transfer coefficient, external to the valve wall, is constant and input by the user.

4. All areas and distances are approximations based on the volume, the mass, or input data that is appropriate.

6.22.5 Limitations

The current limitation of TVALV22 is the possible need for a variable orifice coefficient, particularly in the overlap region.

An undesireable feature is the need for up to four nodes at the junctions with the lines when all the parts have a significant flow. Leakage flows between connect was 1 and 3, and 2 and 4 and not computed.

6.22.6 Variable Listing

Variable	Description	Dimension
A′)	Computational Array	-
AAA	Dummy variable	-
D(ACVF)	Cross sectional area of the valve fluid	IN^2
ACVN	Cross sectional area of the valve walls	IN^2
ADD	Dummy variable	-
D(ASAV)	Surface area - atmosphere to valve	IN^2
D(ASFV)	Surface area - fluid to valve walls	IN^2
A1,AVG	Dummy variables	-
B( )	Computational array	-
CIP	Dummy variable	-
CJ	Mechanical equivalent of heat	$FT\text{-}LB_m/WATTS\text{-}SEC$
CPVW	Specific heat of the valve	$WATTS\text{-}SEC/LB_m\text{-}°F$
CV	Thermal conductivity of the valve	WATTS/IN-°F
DCAPT( )	Heat added to fluid due to pressure drop	°F

Variable	Description	Dimensions
DDD	Dummy variable	-
D(DELTAX)	Distance from entrance to exit or valve openings	IN.
DXVF	Distance from a fluid node to the interface with a connecting line	IN.
DXVW	Distance from node to interface of valve and lines	-
EPSION	Emissivity factor	-
FAC	Dummy variable	-
FMASS	Fluid mass of each node	LB_m
I,IERR,IJ,IS	Dummy variables	-
D(ITF)	Initial fluid temperature	°F
D(ITV)	Initial valve wall temperature	°F
KTYPE	Dummy variable	-
D(LEAK5)	Laminar leakage coefficient	PSI/CIS
D(LEAK6)	Laminar leakage coefficient	PSI/CIS
L1,L2,L3,L4, M5,M6,NM	Dummy Variables	-
D(MTYPE)	Value material type	-
D(PERC)	Percentage heat added to fluid due to pressure drop	-
QI( )	Array of internal volume flow rates	CIS
RHOIL	Fluid density	LB_m/IN^3
RHOV	Density of the valve mass	LB_m/IN^3
RMD( ), RMF( )	Computational arrays	-
RMT,R1, R2,R3,R4	Dummy variables	-
SHAPF	Shape factor valve case to surrounding structure, constant, .96	-
SIGMA	Stefan-Boltzmann constant for radiation	$WATTS/IN^2 - °R^4$

Variable	Description	Dimension
D(TA)	Temperature of the surrounding atmosphere	°F
TEMP1,TERM	Dummy variables	-
TO(I)	Array of valve fluid node temperatures	°F
T(TST)	Temperature of the surrounding structure	°F
DT(TVW)	Temperature of the valve walls	°F
U( )	Heat transfer coefficients internal to the valve walls	$WATTS/IN^2$-°F
D(UAW)	Heat transfer coefficient - atmosphere to valve walls	$WATTS/IN^2$-°F
D(VMASS)	Valve mass	LB_m
D(VOLUME)	Volume inside valve	IN^3
XT,XV	Dummy variables	-

6.22.7 Subroutine Listing

```
      SUBROUTINE TVALV22 (D,DT,DD,L)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300)
     + ,ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NLL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /STEADY/PN(90),QN(90),PLX(90),POLDG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,NMODE,NLLG,NCPN,TERM,LEGN,ICON,INV,
     + INX,INZ,NUP(90),NDWN(90),NLLEG(90),ILEGAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION D(1),DT(1),DD(1),L(1)
      DIMENSION TO(4),DCAPT(4,6),RAD(4,3),RAF(4),A(5,5),B(5),
     + H(4,6),QI(6),R(6)
      INTEGER UAV,ASAV,TST,TA,ITV,VMASS,DELTAX,PERC,ASFV,TVW,
     + ITF,VOLUME,ACVF
C     D ARRAY VARIABLES
      DATA MTYPE/17/,VMASS/18/,VOLUME/19/,DELTAX/20/,ASFV/21/,ASAV/22/
     + ,UAV/24/,PERC/25/,TST/26/,TA/27/,ITF/28/,ITV/29/,ACVF/23/
     + ,LEAK5/31/,LEAK6/32/
C     DT ARRAY VARIABLES
      DATA TVW/17/
      DATA SIGMA/.349E-11/,SHAPE/1.0/,EPSION/.3/,CJ/3.85/
      IF(IENTR) 1000,2000,3000
 1000 CONTINUE
C     DELTAX =AVERAGE DISTANCE FROM ONE INLET TO ANOTHER INLET
C      (AVERAGE OF ALL POSSIBLE COMBINATIONS)
C     ASAV   =SURFACE AREA CASE TO AMBIENT
C     ASFV   =WETTED SURFACE AREA OF  LUID TO CASE(INCLUDING PISTON AREA
).
C     PERC   =PERCENTAGE OF HEAT ADDED TO FLUID DUE TO PRESSURE DROP
C     QI(I)   =FLOW IN VALVE FROM I TO I+1 PORTS
C     MTYPE  =VALVE MATERIAL TYPE
C     VOLUME  =TOTAL VOLUME OF LIQUID IN VALVE
C     L(5)=NUMBER OF X DATA PTS.
      LL=(L(5)+7)/8
      L(7)=33+LL*8
      L(6)=33
      KTYPE=D(MTYPE)+.001
      CPVM=PROP(KTYPE,1)
      RHOV=PROP(KTYPE,2)
      CV=PROP(KTYPE,3)
      DO 1010 I=1,4
      LI=L(I)
      TC(LI)=D(ITV)
      TF(LI)=D(ITF)
 1010 TO(I)=D(ITF)
      DT(TVW)=D(ITV)
      D(ASFV)=D(ASFV)/4.0
      IF (D(UAV).LE.0.0) D(UAV)=.0069
      D(TST)=(D(TST)+460.)**4
      RETURN
 2000 CONTINUE
      N=4*ICON-3
```

6.22.7 (Continued)

```
      FAC=D(N+2)*SQRT(2./RHO(TF(L(ICON)),PUP))*.65
      XT=D(N+1)-D(N)
      AVG=(D(N)+D(N+1))/2.
      IF(XT.EQ.0.0)XT=.0001
      IF(FAC.LE.0.0)FAC=.00001
      CALL INTERP(TIML,D(L(6)),D(L(7)),10,L(5),XV,IERR)
      XT=(XV-AVG)/XT
      TERM=FAC*(.5+XT/(1.+ABS(XT*2.)**D(N+3))**(1./D(N+3
     + )))+.0001
C     WRITE(6,900) N,AVG,QI,FAC,XT,XV,TERM
  900 FORMAT(10X,I10,6E12.5)
      PUP=PUP-QS*(QA/TERM)**2
      QI(ICON)=QI
      RETURN
 3000 CONTINUE
      L1=L(1)
      L2=L(2)
      CPVW=PROP(KTYPE,1)
      RHOV=PROP(KTYPE,2)
      CV=PROP(KTYPE,3)
      L3=L(3)
      L4=L(4)
      QI(5)=(P(L1)-P(L3))/D(LEAK5)
      QI(6)=(P(L2)-P(L4))/D(LEAK6)
      QI(5)=0.0
      QI(6)=0.0
      TO(5)=TO(1)
      TO(6)=TO(2)
      RHOIL=386.4*RHO((TO(1)+TO(2)+TO(3)+TO(4))/4.0,(P(L1)+P(L2)
     + +P(L3))/3.)
      A1=D(UAV)*D(ASAV)
      FMASS=D(VOLUME)*RHOIL/4.0
      CIP=SIGMA*SHAPF*EPSIDN*D(ASAV)
      ACVW=D(VMASS)/(RHOV*D(DELTAX))
      DXVW=D(DELTAX)/2.0
      DXVF=D(DELTAX)/4.
      R1=1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXVW/(ACVW*CV))
      R2=1.0/(DXF(L2)/(ACW(L2)*C(L2))+DXVW/(ACVW*CV))
      R3=1.0/(DXF(L3)/(ACW(L3)*C(L3))+DXVW/(ACVW*CV))
      R4=1.0/(DXF(L4)/(ACW(L4)*C(L4))+DXVW/(ACVW*CV))
      DO 3003 I=1,4
      DO 3003 J=1,3
 3003 RAD(I,J)=0.0
      DO 3006 I=1,5
      DO 3006 J=1,5
      A(I,J)=0.0
 3006 B(I)=0.0
      AAA=D(ACVF)
      DDD=SQRT(AAA*4./PI)
      M5=4
```

6.22.7 (Continued)

```
      M5=5
      IS=1
 3009 DO 3400 I=1,4
      N=L(I)
      RHOIL=386.4*RHO(TO(I),P(N))
      RMF(I)=Q(N)*RHOIL
      IF(RMF(I).LT.0.0) RMF(I)=0.0
      R(I)=CF/(DXP(L(I))/ACP(L(I))+DXVF/D(ACVF)+RMF(I)*
     + DELT/(D(ACVF)**2*RHOIL))
 3103 RMT=QI(I)*RHOIL
      IF(RMT.GE.0.0) RMT=0.0
      U(I,I)=UFW(AAA,DDD,ABS(QI(I)),TO(I),P(N))
      TL P1=DT(TVW)
      IF(RMT.GE.0.0) U(I,I)=0.0
      RMD(I,1)=ABS(RMT)
 3106 RMT=QI(M5)*RHOIL
      IF(RMT.LT.0.0) RMT=0.0
      U(I,M5)=UFW(AAA,DDD,ABS(QI(M5)),TO(M5),P(L(M5)))
      IF(RMT.LE.0.0) U(I,M5)=0.0
      RMD(I,2)=RMT
      IF(I.EQ.3.OR.I.EQ.4) GO TO 3200
 3109 RMT=QI(M6)*RHOIL
      IF(RMT.GE.0.0) RMT=0.0
      U(I,M6)=UFW(AAA,DDD,ABS(QI(M6)),TO(I),P(N))
      IF(RMT.GE.0.0) U(I,M6)=0.0
      RMD(I,3)=ABS(RMT)
      GO TO 3300
 3200 RMT=QI(M6)*RHOIL
      IF(RMT.LE.0.0) RMT=0.0
      U(I,M6)=UFW(AAA,DDD,ABS(QI(M6)),TO(I),P(N))
      IF(RMT.LE.0.0) U(I,M6)=0.0
      RMD(I,3)=ABS(RMT)
 3300 CONTINUE
      A(I,I)=RMASS*CPFN/DELT+RMD(I,1)*CPFN+RMD(I,2)*CPFN+RMD(I,3)
     + *CPFN+(U(I,I)+U(I,M5)+U(I,M6))*D(ASFV)+RMF(I)*CPFN+R(I)
      IJ=I+2
      IF(I.GT.2) IJ=I-2
      A(I,IJ)=-RMD(I,3)*CPFN
      IJ=I+1
      IF(I.EQ.4) IJ=1
      M=L(IJ)
      A(I,IJ)=-RMD(I,1)*CPFN
      DCAPT(I,I)=(1.0/RHOIL)*ABS(P(N)-P(M))/(CJ*CPFN)
      A(I,M5)=-RMD(I,2)*CPFN
      M=L(M5)
      DCAPT(I,M5)=(1.0/RHOIL)*ABS(P(M)-P(N))/(CJ*CPFN)
      M=L(I+2)
      IF(I.GT.2) M=L(I-2)
      DCAPT(I,M6)=(1.0/RHOIL)*ABS(P(M)-P(N))/(CJ*CPFN)
      A(I,5)=(-U(I,I)-U(I,M5)-U(I,M6))*D(ASFV)
```

6.22.7 (Continued)

```
      B(I)=FMASS*CPFN*TO(I)/DELT+RMD(I,1)*CPFN*D(PLRC)*DCAPT(I,I)+
     + RMD(I,2)*CPFN*DCAPT(I,M5)*D(PLRC)+RMD(I,3)*D(PLRC)*DCAPT(I,M6)*
     + CPFN+RMF(I)*CPFN*TF(L(I))+R(I)*TF(L(I))
      A(5,I)=(-U(I,I)-U(I,M5)-U(I,M6))*D(ASFV)
      A(5,5)=A(5,5)-A(5,I)
      M5=I
      M6=M6+IS
      IS=-IS
 3400 CONTINUE
      A(5,5)=A(5,5)+D(VMASS)*CPVW/DELT+R1+R2+R3+R4+A1
      B(5)=D(VMASS)*CPVW*DT(TVW)/DELT+R1*TW(L1)+R2*TW(L2)+
     + R3*TW(L3)+R4*TW(L4)+CIP*D(TST)-CIP*((DT(TVW)+460.)**4)
     + +A1*D(TA)
      CALL SIMULT(A,B,5,IERROR)
      DO 3500 M=1,4
      NM=L(M)
      TO(M)=TF(NM)
      TF(NM)=B(M)
      IF(C(NM).GE.0.0)TF(NM)=TO(M)
      TO(M)=B(M)
 3500 TC(NM)=B(5)
      DT(TVW)=B(5)
      RETURN
      END
```

6.31 SUBROUTINE TCVAL31

TCVAL31 simulates a simple undamped check valve as shown in Figure 6.31-1. Although the actual mechanical configurations of these values vary greatly, the basic method of operation remains the same. The subroutine calculates the valve wall temperature, the valve fluid temperature, and the valve poppet temperature.

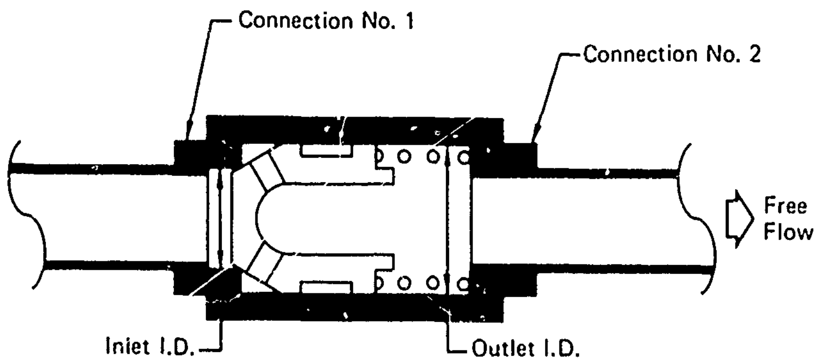


FIGURE 6.31-1
TYPE NO. 31 CHECK VALVE

6.31.1 Math Model

A check valve has a variable geometry orifice, which is opened for forward flow and closed for reverse flow. The thermal math model for the check valve includes heat transfer to and from two connecting line segments, one upstream and one downstream. Seven nodes are considered: three fluid nodes, three wall nodes, and one poppet node.

Temperature nodes are indicated in Figure 6.31-2. For forward flow the temperatures of the upstream line segment wall and fluid nodes are TW(L1) and TF(L1), the temperatures of the valve wall and fluid nodes are DT(TV) and DT(TVF), the temperature of the poppet node is DT(TP), and the temperatures of the downstream line segment wall and fluid nodes are TW(L2) and TF(L2). Three heat balance equations are written to solve for DT(TV), DT(TVF), and DT(TP), using the valve and line material properties and dimensions, the atmosphere and structure temperatures external to the valve, and TW(L1), TW(L2), and TF(L1). (Note TF(L2) = DT(TVF), see assumptions). One equation is a heat balance for the valve fluid node. The second equation is a heat balance for the valve wall node. The third equation is a heat balance for the poppet.

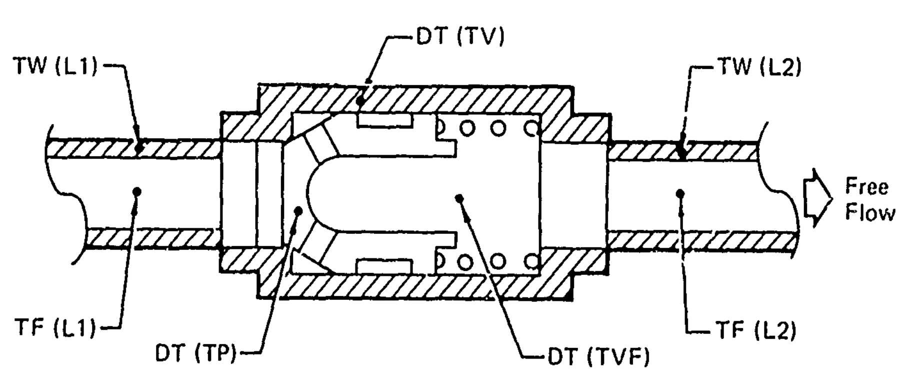


FIGURE 6.31-2
CHECK VALVE NODE REPRESENTATION

GP77-0085-3

The first equation represents three modes of heat transfer relative to the fluid node:

1. Heat transfer due to mass transfer into the valve from upstream of the valve

   B4 * (TF(L1) - DT(TVF))

   where B4 is equal to RMF(1) *CPFN

2. convection to or from the valve walls and poppet

   B1 * (DT(TV) - DT(TVF))

   B2 * (DT(TP) - DT(TVF)) respectively.

   where B1 and B2 are convection coefficients and are equal to UFWIL * D(ASFV) and UFWIL * D(ASFP)

3. heat addition due to a pressure drop across the valve

   B4 * DCAPT * D(PERC) = 1.0/RHOIL*(P(L1) - P(L2))/

   (CJ * CPFN) * D(PERC)*B4

   If the fluid experiences a substantial pressure drop across the valve (greater than 100 psi) then there is heat added directly to the fluid due to this pressure change.

The above heat transfer terms are combined to produce the equation for heat balance for the valve fluid node:

$$\frac{MCp}{DELT} * (DT(TVF) - DT(TVF)_{OLD}) = \begin{aligned} & B4*(TF(L1) - DT(TVF)) \\ + \; & B1*(DT(TV) - DT(TVF)) \\ + \; & B2*(DT(TP) - DT(TVF)) \\ + \; & B4*DCAPT * D(PERC) \end{aligned} \qquad (1)$$

where MCp is equal to FMASS * CPEN

The second equation represents three modes of heat transfer relative to the valve wall node:

1. conduction to or from the upstream and downstream line wall nodes

R3 * (TW(L1) - DT(TV))

R4 * (TW(L2) - DT(TV))

where R3 and R4 are the conduction coefficients are equal to 1.0/(DXF(L1)/(ACF(L1)*C(L1)) + DXV (ACV * CV))

where I = 1 for R3 and 2 for R4

2. (a) convection to or from the fluid in the valve

B1 * (DT(TVF) - DT(TV))

where B1 was described previously

2. (b) convection to or from the external atmosphere

B3 * (D(TA) - DT(TV))

where B3 is the convection coefficient and is equal to D(UAV) * D(ASAV)

3. radiation exchange with the surrounding structure

CIP * (D(TST) - (DT(TVW) + 460) **4)

where CIP is the radiation coefficient and is equal to SIGMA * SHAPF * EPSION * D(ASAV)

These heat transfer modes are combined to produce the equation for heat balance for the valve wall node.

$$\frac{MCp}{DELT} * (DT(TV) - DT(TV)_{OLD}) = B1*(DT(TVF) - DT(TV))+R3*(TW(L1)-DT(TV)) + R4*(TW(L2) - DT(TV))+B3*(D(TA)-DT(TV)) + CIP*D(TST) - CIP*(DT(TV)+460.)**4 \quad (2)$$

where MCp is equal to D(VMASS) * CPVN

The third equation represents a heat balance for the poppet:

$$\frac{MCp}{DELT} * (DT(TP) - DT(TP)_{OLD}) = B2*(DT(TVF) - DT(TP)) \qquad (3)$$

where B2 is a convection coefficient and was described previously, and MCp is equal to D(T PMASS)*CPPN

Equations (1), (2), and (3) are solved for the appropriate temperatures.

For reverse flow as shown in Figure 6.31-3 for the connecting lines are reversed. Three equations can also be written to solve for DT(TVF), DT(TV) and DT(TP).

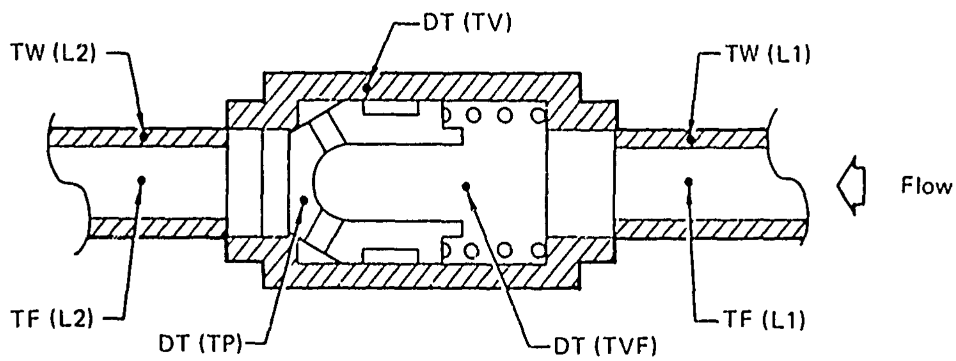


FIGURE 6.31-3

NODE REPRESENTATION FOR REVERSE FLOW

The first equation represents two modes of heat transfer relative to the valve fluid node:

1. conduction to or from upstream connecting line

R1*(DT(TVF) - TF(L1))

where R1 is the conduction coefficient and is equal to CF/(DXF(L1)/ACF(L1) + DXV/ACFV +RMF(1)*DELT/(ACVF **2*RHOIL))

2. convection to or from the valve wall and poppet nodes respectively

$$B1*(DT(TV) - DT(TVF))$$

$$B2*(DT(TP) - DT(TVF))$$

where B1 and B2 were defined previously

The above heat transfer modes are combined to produce the heat balance equation for the valve fluid node when flow is by the valve.

$$\frac{MCp}{DELT} * (DT(TVF) - DT(TVF)_{OLD}) = B1*(DT(TV) - DT(TVF)) + B1*(DT(TP) - DT(TVF)) + R1*(TF(L1) - DT(TVF) \qquad (4)$$

with all terms previously defend.

The second equation is the same as Equation (2) for a heat balance for the valve wall node.

The third equation includes convection between the poppet, and the downstream line fluid node

$$\frac{MCp}{DELT} * (DT(TP) - DT(TP)_{OLD}) = B2*(DT(TVF) - DT(TP)) + B5*(TF(L2) - DT(TP) \qquad (5)$$

where B2 was defined previously and B5 is equal to D(UAV)*ACF(L2)

Equations (2), (4), and (5) are solved for the appropriate temperatures.

A thermal model of the heat transfer terms for the check valve with forward flow is shown in Figure 6.31-4.

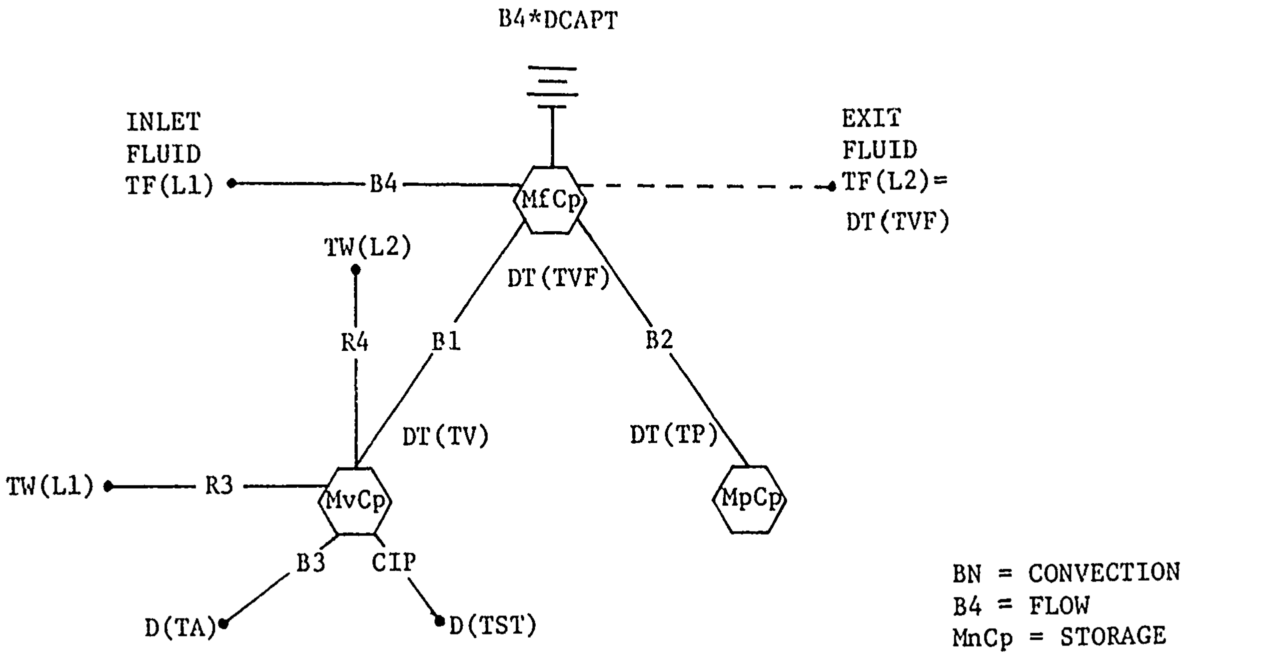


FIGURE 6.31-4
THERMAL MODEL
FORWARD FLOW

The hydraulic math model used to calculate the steady state pressure drop assumes a straight line flow/pressure drop characteristic between the cracking pressure and the fully open position. The cracking pressure drop is set equal to the inlet area divided by the spring preload and the slope, DT(5), is set to the change in pressure required to fully open the poppet divided by the flow at that condition which is

DT(4) = D(1)*CV*SQRT(DT(2)*AHO( )/2.0) (6)

where D(1) is considered to be the maximum valve area.

The orifice resistance at the fully open position, is used when the flow exceeds DT(4). Figure 6.31-5 shows graphically how this is done.

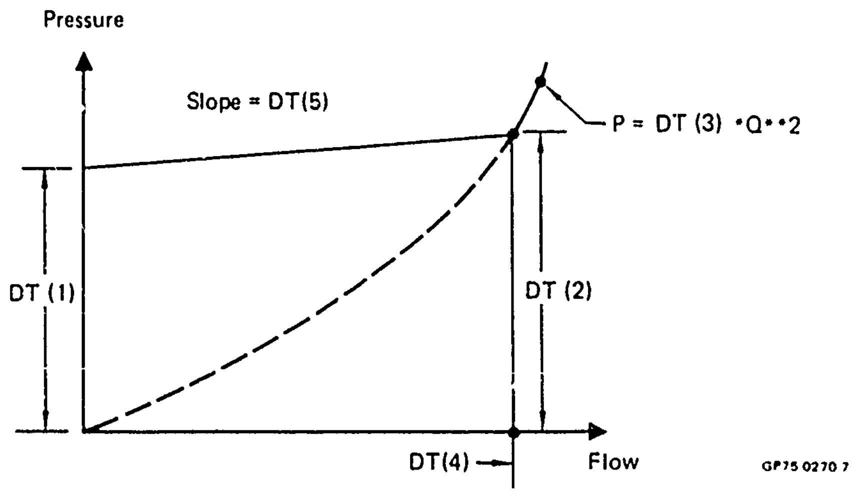


FIGURE 6.31-5

CVAL31 STEADY STATE PRESSURE DROP CHARACTERISTICS

6.31.2 Assumptions -

1. There is no conduction between the poppet and the walls since there is little contact area and the poppet is completely submerged in oil

2. The interface conductance between the valve and line walls is infinite

3. The atmosphere and structure temperatures remain constant

4. The emissivity of the wall material is a constant, .3

5. No friction is generated when the poppet moves.

6. The fluid exiting from the check valve is equal to DT(TVF)

7. There is complete mixing of the fluids in the fluid volume.

6.31.3 Computational Methods

SECTION 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

SECTION 2000

This section is called from TLEGCAL via COMPE using CON #1, if the check valve is connected so that the free flow direction is the same as the positive flow in the leg, or CON #2 if the valve is in backwards. When the valve is closed

L(3) = 1 and QS = -1 or ((3) = 0 and QS = 1

The value impedance is set at 1.0E8, which is essentially an open circuit.

When the valve is fully open (ENTR = 1, QS = 1, or IENTR = 2, QS = -1) plus Q > DT(4), the valve orifice impedance DT(3) is multiplied by the QA**2 term to obtain a pressure drop used in TLEGCAL.

With the same basic conditions but with Q1 > DT(4) the valve characteristics are assumed to be a constant pressure differential, plus a linear flow/pressure gain.

When the flow guess is negative for CON #2 the constant differential becomes a pressure rise.

The three modes of the check valve, closed, partially open and fully open will show up in the leg as a pressure drop or rise.

SECTION 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connection line two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 3x3 matrix is loaded and the mathematical equations are solved for DT(TVF), DT(TV) and DT(TP) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to arrays in COMMON /TRANS/.

6.31.4 Approximations

1. The distance from the valve wall node to the interface of the valve and tube wall is approximated by

   DXV = (D(VLENGTH)/2.0)

2. The cross sectional area of the valve walls

   ACV = D(VMASS)/(D(VENGTH)*RHOV))

3. The check valve wall is treated as one node, thus the entire valve is at the same temperature

4. The shape factors is constant at .96 as described in Section 2.0 of this manual.

6.31.5 Limitations - Not applicable.

6.31.6 Variable Listing

Variable	Description		Dimension
A( )	Computational Array		--
AAA	Dummy Variable		--
ACFV	Cross Sectional Area of the Fluid in value		IN^2
ACV	Cross Section Area of the Valve Wall		IN^2
D(ASAV)	External Surface Area		IN^2
D(ASFP)	Poppet Surface Area		IN^2
D(ASFV)	Internal Check Valve Surface Area	IN^2	IN^2

VARIABLE	DESCRIPTION	DIMENSION
B( )	Computational Array	--
B1, B2, B3, B4, B5	Dummy Variables	--
CIP	Radiation Coefficient	WATTS/°R^4
CJ	Mechanical Equivalent of Heat	IN-LB_m/WATTS-SEC
CPPN	Specific Heat of Check Valve	WATTS-SEC/LB_m-°F
CPVN	Specific Heat of the Poppet	WATTS-SEC/LB_m-°F
CV	Conductivity of the Valve	WATTS/IN-°F
DCAPT	Heat Added Due to Pressure Drop	°F
DDD	Dummy Variables	--
DXV	Distance from Valve Wall Mode to Line Wall Interface	IN
EPSION	Emissivity Factor	--
FMASS	Fluid Mass	LB_m
D(ITF)	Initial Fluid Temperature	°F
D(ITV)	Initial Valve Temperature	°F
KTYPE	Dummy Variable	--
D(MTYPE)	Valve Material Type	--
NTYPE	Dummy Variable	--
D(PERC)	Percentage Heat Added to Fluid Due to ΔP	--
PMASS	Fluid Mass in Valve	LB_m
D(PTYPE)	Poppet Material Type	--
RHOIL	Fluid Density	LBm/IN^3
RHOV	Valve Wall Density	LB_m/IN^3
RMF,R3,R4	Dummy Variables	---
SHAPF	Shape Factor, Valve Walls to Surroundings	--
SIGMA	Stefan - Boltzmann Constant	WATTS/IN^2-°R^4
D(TA)	Surrounding Ambient Temperature	°F

Variable	Description	Dimension
TEMP1,TFO, TFOO	Dummy Variables	--
DT(TP)	Poppet Temperature	°F
D(TPMASS)	Poppet Mass	LB_m
D(TST)	Surrounding Structure Temperature	°F
DT(TV)	Valve Wall Temperature	°F
DT(TVF)	Fluid Temperature in the Valve	°F
D(UAV)	Heat Transfer Coefficient - External to Valve Wall	$WATTS/IN^2$-°F
UFWIL	Heat Transfer Coefficient - Fluid to Valve Wall	$WATTS/IN^2$-°F
D(VLENGTH)	Valve Length	IN
D(VMASS)	Valve Mass	LB_m
D(VOL1)	Volume of Fluid Inside Valve	IN^3

For variables in common refer to Paragraph 3.3.

6.31.7 SUBROUTINE LISTING

```
      SUBROUTINE PCVAL31 (D,DT,DD,L)
C**** REVISED AUGUST 5, 1975 ****
      DIMENSION D(1),DT(1),DD(1),L(1)
      DOUBLE PRECISION DD
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,ILIMIT,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IRD,ICNTR,LINE
      COMMON /STEADY/WR(90),QN(90),PLX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PLOWR,NNODE,NLEG,NCPN,TERM,LLGN,ICON
     + ,INV,INX,INZ,NOP(90),NODEJ(90),NELEM(90),ILEGAP(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFJ,FTEMP,PROP(13,3)
      DIMENSION A(3,3),B(3),R(2),PMF(2)
      INTEGER VOL1,ASFP,ASFV,ASAV,UAV,TPMASS,VMASS,PA,TST,ITF,ITV
     + ,TV,MTYPE,PTYPE,VLENGTH,PERC,TP,TVF
C      D ARRAY VARIABLES
      DATA MTYPE/7/,PTYPE/8/,VMASS/9/,TPMASS/10/,VOL1/11/
     + ,VLENGTH/12/,ASFP/13/,ASFV/14/,ASAV/15/,UAV/16/,PERC/17/
     + ,TST/18/,PA/19/,ITF/20/,ITV/21/
C      DT ARRAY VARIABLES
      DATA TVF/7/,TV/8/,TP/9/
      DATA SIGMA/.349E-11/,SHAPE/.96/,EPSION/.3/,CJ/8.85/
C      VOL1  =VOLUME OF FLUID INSIDE VALVE
C      ASFP  =SURFACE AREA FLUID TO POPPET
C      ASFV  =SURFACE AREA FLUID TO VALVE WALL
C      MTYPE =MATERIAL TYPE OF VALVE CASE
C      PTYPE =MATERIAL TYPE OF POPPET
C      VMASS =MASS OF VLAVE INCLUDING POPPET (LB.)
C      D(1)=INTERNAL DIAMETER (INLET)
C      D(2)=INTERNAL DIAMETER (OUTLET) NOT IN USE
C      D(3)=POPPET MASS
C      D(4)=SPRING CONSTANT
C      D(5)=MAX POPPET DISPLACEMENT
C      D(6)=SPRING PRELOAD
C      DT(1)=CONSTANT STORAGE
C      DT(2)=CONSTANT STORAGE
C      DT(3)= STORAGE FOR FULLY OPEN ORIFICE COEF.
C      DT(4)= STORAGE FOR PREVIOUS POPPET VELOCITY
C      DT(5)=STORAGE FOR PREVIOUS POPPET ACCELERATION
C      DT(6)= STORAGE FOR PREVIOUS POPPET POSITION
      IF(IENTR) 1000,2000,3000
C *** 1000 SECTION
 1000 CONTINUE
C      INITIALIZING TEMPERATURES
      L1=L(1)
      L2=L(2)
      DT(TVF)=D(ITF)
      DT(TV)=D(ITV)
      DT(TP)=D(ITV)
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
```

```
      TC(L1)=D(ITV)
      TC(L2)=D(ITV)
      TPO=D(ITF)
      D(TST)=(D(TST)+460.)**4
      IF(D(UAV).EQ.0.0) D(UAV)=.0069
      D(1)=D(1)**2.*PI/4.
      DT(1)=D(6)/D(1)
      DT(2)=D(4)*D(5)/D(1)+DT(1)
      L(3)=0
      RETURN
C      SECTION FOR STEADY STATE CALCULATION
 2000 CONTINUE
      DT(4)=D(1)*.65*SQRT(DT(2)*RHO(TF(L(1)),PUP)/2.0)
      DT(5)=(DT(2)-DT(1))/DT(4)
      DT(3)=(RHO(TF(L(1)),PUP)/2.0)/(D(1)*.65)**2
      IF(ICON.NE.1.) GO TO 1600
C      THE VALVE IS CONNECTED CONVENTIONALY WHEN CON #1 IS USED
      L(3)=1
      IF(DS.EQ.1.) GO TO 1700
      GO TO 1650
 1600 IF(ICON.NE.2) GO TO 1900
      IF(L(3).NE.0) GO TO 1900
C      THE VALVE IS CONNECTED BACKWARDS WHEN CON #2 IS USED
      IF(DS.LT.1.0) GO TO 1700
C      THE VALVE IS CLOSED
 1650 PUP=PUP-DS*DA*1.013
      RETURN
 1700 IF (DA.LT.DT(4)) GO TO 1800
C      THE VALVE IS FULLY OPEN
      DT(6)=D(5)
      PUP=PUP-DS*DA**2*DT(3)
      RETURN
C      THE FLOW IS LESS THAN THE 'FULL OPEN FLOW'
 1800 PUP=PUP-DT(1)*DA-DA*DA*DT(5)
      RETURN
C*****ILLEGAL INPUT DATA TERMINATES PROGRAM*****
C      EITHER CON. 1 AND 2 OR AN ILLEGAL CON. # WAS CALLED
 1900 STOP 1531
 3000 CONTINUE
      L1=L(1)
      L2=L(2)
      KTYPE=D(KTYPE)+.001
      NTYPE=D(PTYPE)+.001
      CV=PROP(KTYPE,3)
      CP=PROP(NTYPE,3)
      CPP =PROP(NTYPE,1)
      CPV =PROP(KTYPE,1)
      RHOP=PROP(NTYPE,2)
      RHOV=PROP(KTYPE,2)
      TFO=TF(L1)
```

6.31.7 (Continued)

```
      DXV=(D(VLENGTH)/2.0)
      ACFV=PI*(D(1))**2/4.0
      ACV=D(VMASS)/(RHOV*D(VLENGTH))
      RHOIL=336.4*RHO(DT(IVF),P(L2))
      FMASS=D(VOL1)*RHOIL
      AAA=ACFV
      DDD=SQRT(AAA*4./PI)
      UFOIL=UFW(AAA,DDD,ABS(Q(L2)),DT(IVF),P(L2))
      Q1=Q(L2)
      TEMP1=TC(L1)
      RMF(1)=ABS(Q(L1))*RHOIL
      RMF(2)=ABS(Q(L2))*RHOIL
      A1=UFOIL*D(ASFV)
      A2=UFOIL*D(ASFP)
      A3=D(UAV)*D(ASAV)
      R(1)=0.0
      R5=0.0
      IF(Q(L1).GT.0.0)GO TO 3300
      L1=L(2)
      L2=L(1)
      R(1)=CF/(DXF(L1)/ACF(L1)+DXV/ACFV+RMF(1)*DELT/(ACFV**2*RHOIL))
      R5=D(UAV)*ACF(L2)
 3300 CONTINUE
 3303 IF(Q(L1).LT.0.0) RMF(2)=0.0
      A4=RMF(1)*CPFS
      R4=1.0/(DXF(L1)/(ACF(L1)*C(L1))+DXV/(ACV*CV))
      R3=1.0/(DXF(L2)/(ACF(L2)*C(L2))+DXV/(ACV*CV))
      CIR=SIGMA*SHAPE*EPSION*D(ASAV)
      DCAPT=(1.0/RHOIL)*ABS(P(L1)-P(L2))/(CJ*CPFS)
      IF(Q(L1).EQ.0.0) DCAPT=0.0
      A(1,1)=FMASS*CPFS/DELT+A1+A2+R(1)+A4
      A(1,2)=-A1
      A(1,3)=-A2
      A(2,1)=-A1
      A(2,2)=D(VMASS)*CPVS/DELT+A1+A3+R4+R3
      A(2,3)=0.0
      A(3,1)=-A2
      A(3,2)=0.0
      A(3,3)=D(FPMASS)*CPPS/DELT+A2+R5
      B(1)=FMASS*CPFS*DT(IVF)/DELT+(A4+R(1))*TF(L1)+
     + DCAPT*A4*D(PERC)
      B(2)=D(VMASS)*CPVS*DT(IV)/DELT+R4*TW(L1)+R3*TW(L2)+
     + A3*D(TA)+CIR*D(TSP)-CIR*((DT(IV)+460.)**4)+(1.-D(PERC))*
     + A4*DCAPT
      B(3)=D(FPMASS)*CPPS*DT(IP)/DELT+R5*TF(L2)
      CALL SIMULT(A,B,3,IERROR)
      TF(L1)=TFO
      TFOO=TF(L2)
      TF(L2)=B(1)
      IF(Q1.GE.0.0) TF(L2)=TFOO
```

6.31.7 (Continued)

```
TC(L1)=3(2)
TC(L2)=3(2)
DT(PVF)=3(1)
DT(TV)=3(2)
DT(TP)=3(3)
RETURN
END
```

6.41 SUBROUTINE TREST41

TREST41 simulates a fixed, two-way orifice restrictor with two connecting lines as sketched in Figure 6.41-1. The same discharge coefficient is assumed for flow in either direction so that the unit may be installed backwards, i.e., either end may be the entering line.

This subroutine calculates the temperature of the fluid in the restrictor, and the temperature of the restrictor wall.

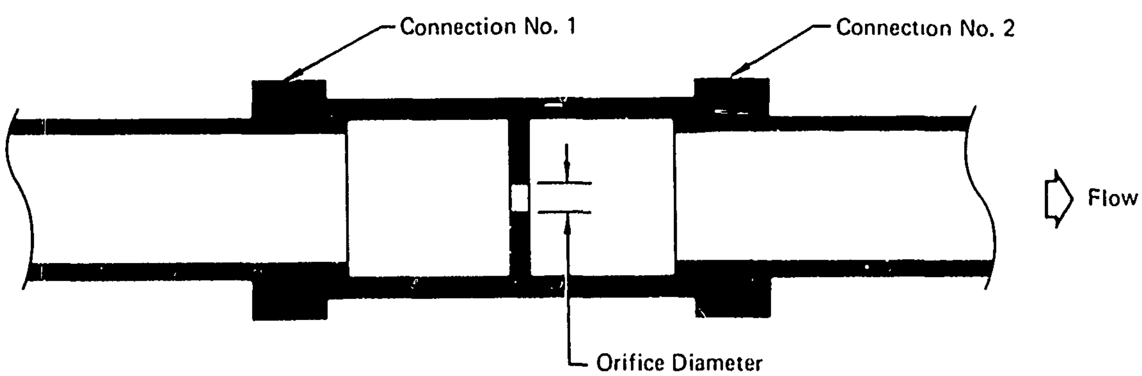


FIGURE 6.41-1
TYPE NO. 41 ORIFICE RESTRICTOR

6.41.1 Math Model - The thermal math model for the restrictor includes heat transfer to and from two line segments, one upstream and one downstream. Six nodes are considered: three fluid nodes and three wall nodes (as shown in Figure 6.41.2). The temperatures of the upstream line segment wall and fluid nodes are TW(L1) and TF(L1), the temperatures of the restrictor

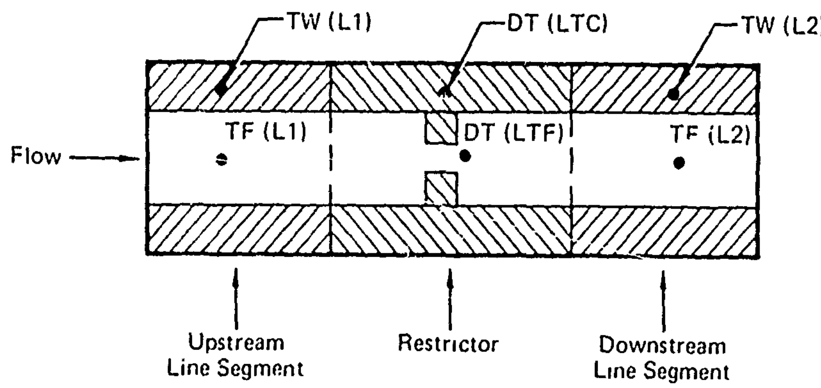


FIGURE 6.41-2
RESTRICTOR AND CONNECTOR NODE REPRESENTATION

GP77-0065-8

wall and fluid nodes are DT(LTC) and DT(LTF), and the temperature of the downstream line segment wall and fluid nodes are TW(L2) and TF(L2). Two heat balance equations are written to solve for DT(LTF) and DT(LTC), using the restrictor and connecting line material properties and dimensions, the atmosphere and structure temperatures external to the restrictor, and TW(L1), TW(L2), and TF(L1). (Note: TF(L2) = DT(LTF), see assumptions).

One equation is for heat transferred to and from the restrictor fluid node. The other equation is for the heat balance for the restrictor wall node.

The first equation represents four modes of heat transfer relative to the restrictor fluid node:

1. conduction to and from the upstream line segment fluid node

   R1 * (TF(L1) - DT(LTF))

   where R1 is the conduction coefficient between the fluids, and is equal to CF/(DXF(L1)/ACF(L1) + DXR/ACFR + RMFL1 * DELT/(ACFR**2*RHOIL))

2. convection with the restrictor wall node

   B1 * (DT(LTC) - DT(LTF))

   where B1 is the convection coefficient between the fluid and the wall and is equal to UFWIL*ASFR

3. heat transfer due to mass transfer into the restrictor node from the upstream of the restrictor node

   $\dot{M}$Cp*(TF(L1)-DT(LTF))

   where $\dot{M}$Cp is the flow rate and is equal to Q(L1)*RHOIL*CPFN

4. heat transfer due to a pressure drop across the orifice

   $\dot{M}$Cp*DCAPT*D(PERC)

   where DCAPT is the temperature rise due to a pressure drop and is equal to (1.0/RHOIL)*(P(L1) - P(L2))/(CJ * CPFN)

These heat transfer terms are combined to produce the equation for heat balance for the restrictor fluid:

$$\mathrm{MCp(DT(LTF)-DT(LTF)_{OLD})} = \mathrm{R1*(TF(L1)-DT(LTF)+B1*(DT(LTC)-DT(LTF))+} \\ \mathrm{\dot{M}Cp*(TF(L1)-DT(LTF))+\dot{M}Cp*DCAPT*D(PERC)} \quad (1)$$

where MCp is equal to DT(RFM)*CPFN.

The second equation represents four nodes of heat transfer relative to the restrictor wall node:

1a. conduction to and from the upstream line segment wall

R3 * (TW(L1) - DT(LTC))

where R3 is the conduction coefficient and is equal to 1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXR/(DT(ACWR)*CR))

1b. conduction to and from the downstream line segment wall

R4 * (TW(L2) - DT(LTC))

where R4 is the conduction coefficient and is equal to 1.0/(DXF(L2)/(ACW(L2)*C(L2))+DXR/(DT(ACWR)*CR))

2a. convection to and from the restrictor fluid

B1 * (DT(LTF) - DT(LTC))

where B1 is the convection coefficient, defined previously

2b. convection to and from the external atmosphere

B2 * (D(TA) - DT(LTC))

where B2 is the convection coefficient and is equal to D(UAR) * D(ASAR)

3. radiation exchange with the surrounding structure

$$CIP*(D(TST)-(DT(LTC)+460.0)^4)$$

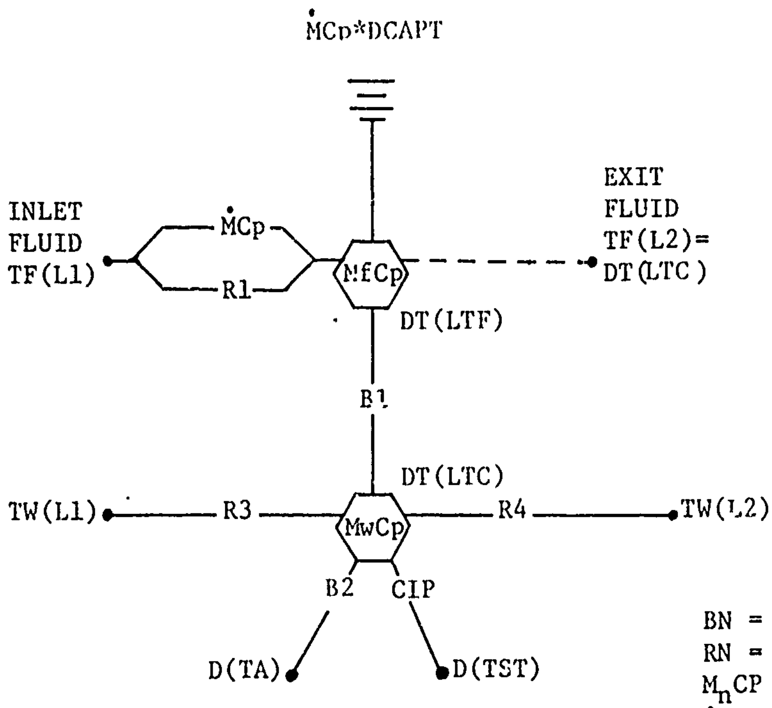
where CIP is the radiation coefficient and is equal to SIGMA*EPSION*SHAPF*D(ASAR)

These heat transfer terms are combined to produce the equation for heat balance for the restrictor wall node:

$$\begin{aligned}\frac{Mcp}{DELT}(DT(LTC)-DT(LTC)_{OLD}) &= R3*(TW(L1)-DT(LTC))+R4*(TW(L2)-DT(LTC)) \\ &+ B1*(DT(LTF)-DT(LTC))+B2*(D(TA)-DT(LTC)) \\ &+ CIP*D(TST)-CIP*(DT(LTC)+460.)**4)\end{aligned} \qquad (2)$$

where MCp is equal to RMASS*CPWR

A thermal model of the above heat transfer terms for the restrictor is shown in Figure 6.41-3. Equations (1) and (2) are solved simultaneously for the fluid and component wall temperatures.



BN = CONVECTION
RN = CONDUCTION
M_nCP = STORAGE
$\dot{M}$Cp = FLOW
CIP = RADIATION
$\dot{M}$Cp*DCAPT = GENERATION

FIGURE 6.41-3
THERMAL MODEL

In the hydraulic math model, the basic equation for flow through an orifice is used to compute the orifice pressure drop.

ΔP = Q1**2*RHO( )/(D(13)*D(12)**2*2) (3)

where	Q1	=	flow (CIS)
	RHO	=	fluid density (LB-SEC^2/IN^4)
	D(13)	=	orifice area (IN**2)
	D(12)	=	discharge coefficient
	ΔP		pressure drop (PSI)

6.41.2 <u>Assumptions</u> - The following assumptions are made to write equations (1) and (2) discussed in Section 6.41.1.

1. The temperature of the fluid leaving the restrictor is equal to the restrictor fluid node temperature, DT(LTF)
2. The pressure drop across the restrictor orifice raises the temperature of the restrictor fluid, not the temperature of the restrictor wall.
3. The temperatures of the atmosphere and structure surrounding the restrictor are constant.
4. The emissivity of the wall material is constant. (.3 for steel)
5. The interface conductance between the restrictor wall and line walls is infinite.
6. The discharge coefficient is considered the same in either flow direction.
7. Complete fluid mixing in the restrictor volume.

6.41.3 Computation Methods

SECTION 1000

The fluid and wall temperatures are initialized; the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

The input orifice diameter D(13) is converted to an area and a steady state orifice equation constant is calculated using the formula:

D(13) = 1./((D(13)*D(12))**2*2)

SECTION 2000

The pressure drop through the orifice is computed using equation (4).

PUP = PUP - Q1*QA*RHO(TF(L1)),PUP)*D(13) (4)

where PUP = upstream pressure

QA = absolute value of Q1

SECTION 3000

Property values are assigned. Dimensions and coefficients are calculated and the flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connection line two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 2x2 matrix is loaded and the mathematical equations are solved for DT(LTF) and DT(LTC) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to arrays TC and TF in COMMON /TRANS/ for distribution throughout the entire program.

6.41.4 Approximations

1. The shape factor is 0.96 (described in Section 2.0).
2. The coefficient for heat transfer between the wall and the external atmosphere is assumed equal to .0069, if not input by the user.

6.41.5 Limitations

The subroutine is limited to fixed two way restrictors having the same discharge coefficient for flow in either direction.

6.41.6 Variable Listing

Variable	Description	Dimension
A( )	Computational Array	--
AAA	Dummy variable	--
ACFR	Cross sectional area of the fluid in restrictor	IN^2
DT(ALWR)	Cross sectional area of the restrictor walls	IN^2
D(ASAR)	Surface area surrounding atmosphere to case	IN^2
ASFR	Surface area of the fluid and wall	IN^2
A1,A2	Dummy variables	--
B( )	Computational array	--
BL,B2, B3,B4	Dummy variables	--
CIP	Radiation coefficient	WATTS/°R^4
CJ	Mechanical equivalent of heat	FT-LB_m/WATTS-SEC
CPRW	Specific heat of the wall	WATTS-SEC/LB-°F
CR	Thermal conductivity of the wall	WATTS/IN-°F
Cl	Dummy variable	--

Variable	Description	Dimension
DCAPT	Heat added to fluid due to pressure change	°F
DDD	Dummy variable	--
D(DIA)	Orifice diameter	IN^2
DXR	Distance from wall node to interface with the connecting line segment	IN
EPSION	Emissivity factor for the walls	--
IERROR	Dummy variable	--
D(ITC)	Initial temperature of the wall	°F
D(ITF)	Initial temperature of the fluid	°F
DT(LTC)	Restrictor wall temperature	°F
DT(LTF)	Restrictor fluid temperature	°F
L1,L2	Addresses of leg and component data	--
MTYPE	Dummy variable	--
D(PERC)	Percentage heat DCAPT, added to fluid	--
DT(RFM)	Mass of the fluid	LB_m
RHOIL	Fluid density	LB_m/IN^3
RHOR	Density of the restrictor wall	LB_m/IN^3
D(RLENGT)	Length of restrictor	IN
D(RMASS)	Mass of the restrictor	LB
RMFL1	Entering mass flow rate	LB_m/SEC
RMFL2	Exiting mass flow rate	LB_m/SEC
D(RTYPE)	Material type	--
R1,R3,R4	Dummy variables	--
SHAPF	Shape factor walls to surrounding structure	--
SIGMA	Stefan-Boltzmann constant for radiation	$WATTS/IN^2-R^4$

Variable	Description	Dimension
DT(SPMAFR), DT(SPMAR)	Dummy variables	--
D(TA)	Surrounding atmospheric temperature	°F
D(TST)	Surrounding structure temperature	°F
D(UAR)	Heat transfer coefficient (surrounding atmosphere to walls)	WATTS/IN^2-°F
UFWIL	Heat transfer coefficient for fluid to walls	WATTS/IN^2-°F
D(VOLF)	Volume of fluid in restrictor	IN^3

6.41.7 Subroutine Listing

```
      SUBROUTINE TRESP41 (D,DT,DD,L)
C *** REVISED AUGUST 20,1976 ***
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + PA,QS,Q1,PUP,PDOWN,NNODE,NLEG,NCPR,TLR.,
     +  LEGN,ICON,INV,INX,INZ,NUP(90),NDWN(90),NELEM(90),
     +ILEGAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      INTEGER UAR,TST,TA,VOLF,ASAR,ITC,ITF,RMASS
     + ,SPMAFR,SPMAR,RLENGT,PERC,ACWR,RFM,RTYPE,DIA
      DIMENSION A(2,2),B(2)
C     D ARRAY VARIABLES
      DATA RTYPE/1/,RMASS/2/,RLENGT/4/,VOLF/3/,ASAR/5/,
     + UAR/6/,PERC/7/,TST/8/,TA/9/,ITF/10/,ITC/11/
      DATA SIGMA/.349E-11/,SHAPE/.96/,EPSION/.3/
     +   ,CJ/8.85/
C     DT ARRAY VARIABLES
      DATA LTC/1/,LTF/2/,SPMAFR/3/,SPMAR/4/,ACWR/5/,RFM/6/,
     + DIA/7/
      IF(IENTR) 1000,2000,3000
 1000 CONTINUE
      DT(DIA)=D(13)
      D(13)=D(13)**2*PI/4.
      D(13)=1./((D(13)*D(12))**2*2.)
      DT(LTC)=D(ITC)
      DT(LTF)=D(ITF)
      L1=L(1)
      L2=L(2)
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
      TC(L1)=D(ITC)
      TC(L2)=D(ITC)
      D(TST)=(D(TST)+460)**4
      IF(D(UAR).EQ.0.0) D(UAR)=0.0069
      MTYPE=D(RTYPE)+.001
      DT(ACWR)=D(RMASS)/(PROP(MTYPE,2)*D(RLENGT))
C     MTYPE   =MATERIAL TYPE
C     RLENGT  =RESTRICTOR LENGTH
C     SPMAR   =MASS(#)*SPECIFIC HEAT
C     D(ASAR) =SURFACE AREA RESTRICTOR TO AMBIENT
C     D(ITC)  =INITIAL TEMPERATURE OF COMPONENT
C     D(UAR)  =HEAT TRANSFER COEFFICIENT WALL TO AMBIENT
C     D(TA)   =TEMPERATURE OF AMBIENT
C     D(TST)  =TEMPERATURE OF STRUCTURE, DEG. R
C     D(VOLF) =VOLUME OF FLUID INSIDE RESTRICTOR
C     MFL1 =MASS FLOW RATE INTO RESTRICTOR
C     D(RMASS)  =MASS OF RESTRICTOR WALLS
```

6.41.7 (Continued)

```
C       DT(LTC) =TEMPERATURE OF COMPONENT
        RETURN
C       STEADY STATE SECTION
 2000 CONTINUE
        PUP=PUP-D1*OA*D(13)*RHO(TF(L(1)),PUP)
        RETURN
 3000 CONTINUE
        ITYPE=D(RTYPE)+.001
        CPRW=PROP(ITYPE,1)
        CR=PROP(ITYPE,3)
        RHOR=PROP(ITYPE,2)
        L1=L(1)
        L2=L(2)
        ACFR=D(VOLF)/(D(RLENGT))
        ASFR=SQRT(4.*ACFR/PI)*PI*D(RLENGT)
C       B1=HEAT TRANSFER COEFF.*SURFACE AREA WALL TO FLUID
        AAA=(ACFR+PI*DP(DIA)**2/4.)
        DDD=SQRT(AAA*4./PI)
        UFWIL=UFW(AAA,DDD,Q(L1),TF(L1),P(L1))
        B1=UFWIL*ASFR
        RHOIL=386.4*RHO(TF(L(1)),P(L(1)))
        DP(RFM)=D(VOLF)*RHOIL
        B2=D(HAR)*D(ASAR)
        CIP=SIGMA*EPSION*SHAPE*D(ASAR)
        DXR=D(RLENGT)/2.0
        DP(SPHFR)=DP(RFM)*CPRW
        DP(SPHAR)=D(RMASS)*CPRW
C       COMPUTE NEW TEMPERATURES
        C1=CIP*(DP(LTC)+460.)**4
        IF(Q(L1).GT.0.0) GO TO 4100
        L2=L1
        L1=L(2)
 4100 CONTINUE
        DCAPT=((1.0/RHOIL)*1.0*ABS(P(L1)-P(L2)))/(CJ*CPFR)
        RMFL1=ABS(Q(L1))*RHOIL
        RMFL2=ABS(Q(L2))*RHOIL
        A2=RMFL1*CPRW
        F1=CF/(DXF(L1)/ACF(L1)+DXR/ACFR+RMFL1*DELT
     + /(ACFR**2*RHOIL))
        B3=1.0/((DXF(L1)/(C(L1)*ACW(L1)))+DXR/(CR*DP(ACWR)))
        B4=1.0/((DXF(L2)/(C(L2)*ACW(L2)))+DXR/(CR*DP(ACWR)))
        D4=DP(SPHFR)/DELT+B1+A2+B1
        A1=DP(SPHAR)/DELT+B3+B4+B2+B1
        A(1,1)=D4
        A(1,2)=-B1
        A(2,1)=-B1
        A(2,2)=A1
        B(1)=DP(SPHFR)*DP(LTF)/DELT+(F1+A2)*TF(L1)+D(PLFC)
     + *DCAPT*A2
        B(2)=DP(SPHAR)*DP(LTC)/DELT+B3*TW(L1)+B4*TW(L2)+B2*
```

6.41.7 (Continued)

```
     + D(TA)+CIP*D(TST)-C1
      CALL SIMULT(A,B,2,IERROR)
      DT(LTC)=B(2)
      DT(LTF)=B(1)
      TF(L2)=B(1)
      TC(L1)=B(2)
      TC(L2)=B(2)
      RETURN
      END
```

6.51 Subroutine TPUMP51

Subroutine TPUMP51 simulates a variable displacement inline piston pump sketched in Figure 6.51-1. The subroutine calculates the temperatures of the exit fluid, the inlet chamber fluid, the case fluid, the internal moving parts (assumed one node), and the pump walls.

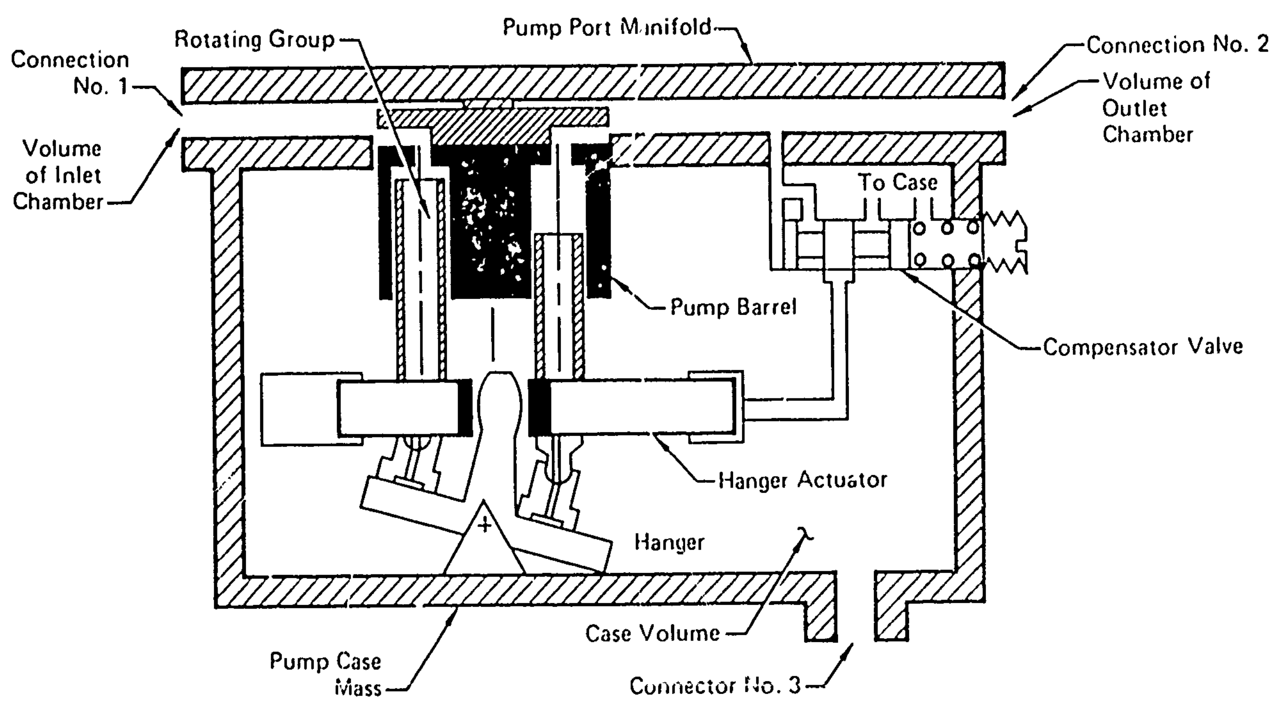


FIGURE 6.51-1
TYPE NO. 51 PRESSURE REGULATED VARIABLE DISPLACEMENT PUMP

6.51.1 Math Model

The thermal math model for the pump includes heat transfer to and from three connecting line segments, one upstream, one downstream and one at the case drain. Thirteen nodes are considered: six fluid nodes, six wall nodes, and one node for the internal moving parts of the pump, called the piston node (as shown in Figure 6.51-2). The pump consists of

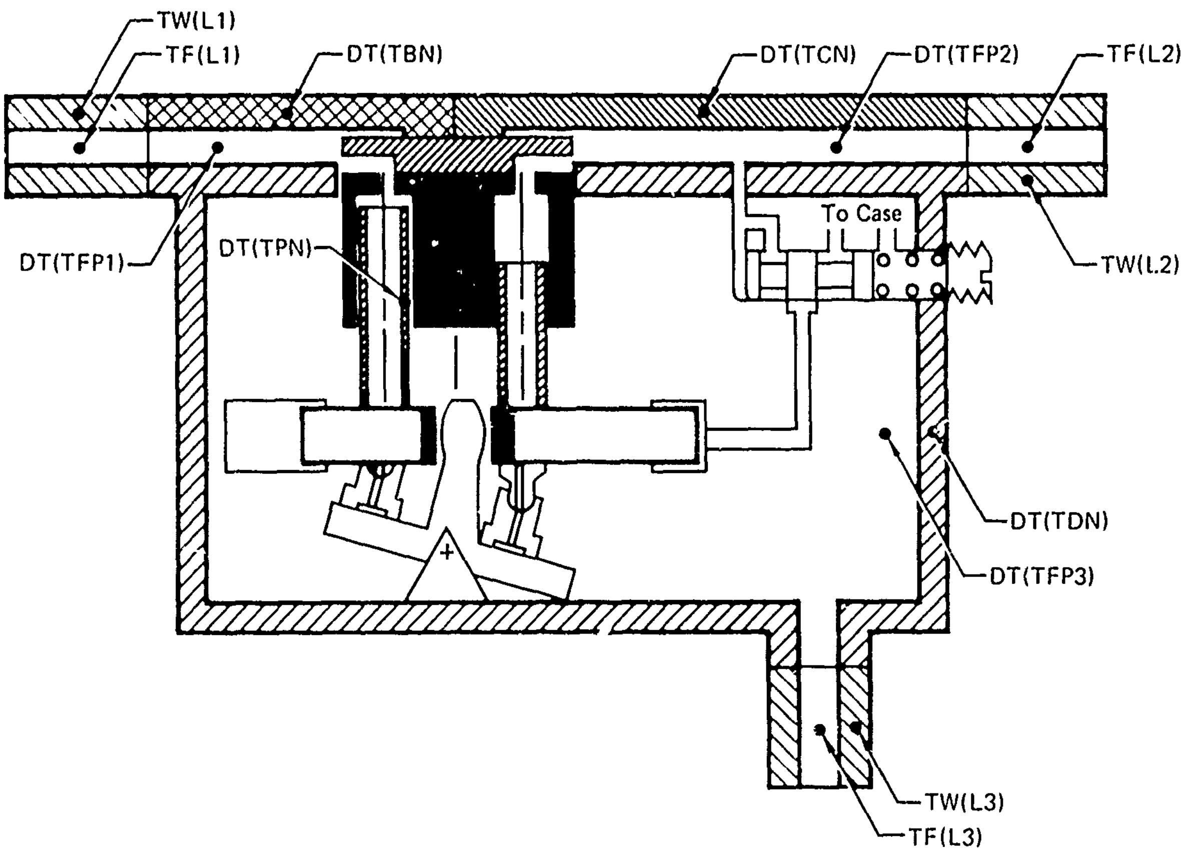


FIGURE 6.51-2
TYPE NO. 51 PRESSURE REGULATED VARIABLE DISPLACEMENT PUMP AND LINE NODE REPRESENTATION

GP77-0065-22

seven nodes: three fluid (one inlet, one outlet, and one case), three walls, one inlet, one outlet, and one around the case drain, and one node for the internal moving parts of the pump, the piston.

The temperatures of the three line segment nodes are TF(L1) and TW(L1), TF(L2) and TW(L2), and TF(L3) and TW(L3) for the inlet segment, exit segment, and case drain line segment fluid and wall nodes respectively. The pump inlet volume temperature is DT(TFP1), exit volume temperature is DT(TFP2), the case drain fluid volume temperature is by DT(TFP3), the pump wall temperature, around the inlet, is denoted by DT(TBN), the wall temperature around the exit is DT(TCN), the wall temperature around the case drain is DT(TDN), and the pistons temperature denoted by DT(TPN).

Seven heat balance equations are written to solve for the seven pump node temperatures, using the pump and line segment material properties and dimensions, the atmosphere and structure temperatures external to the pump, and TF(L1), TW(L1), TW(L1), TW(L2) and TW(L3). (Note TF(L2) = DT(TFP2) and TF(L3) = DT(TFP3), see assumptions).

The first equation represents three modes of heat transfer relative to the pump inlet fluid volume (the volume within the wall node DT(TBN)).

1. Heat transfer due to mass transfer into the pump volume from the upstream line segment.

   $\dot{M}$Cp*(TF(L1)-DT(TFP1))

   where $\dot{M}$Cp is equal to Q(L1)*RHOIL*CPFN

2. Convection to or from the pump walls around the inlet volume

   B1*(DT(TBN)-DT(TFP1))

   where B1 is equal to D(UP1B)*ASP1B, a convection coefficient.

3. Conduction to or from the upstream fluid line segment

   R1*(TF(L1)-DT(TFP1))

   R1 is the conduction coefficient equal to

   CF/(DXF(L1)/ACF(L1)+DXP1/ACP1+(RMFL1*DELT)/(ACF(L1)**2*RHOIL))

   where RMFL1=Q(L1)*RHOIL

These three heat transfer modes then combine to produce the heat balance equation for the pump inlet fluid node.

$$\frac{MCp}{DELT}*(DT(TFP1)-DT(TFP1)_{OLD}) = (R1+\dot{M}Cp)*(TF(L1)-DT(TFP1))+$$
$$B1*(DT(TBN)-DT(TFP1))$$

with MCp equal to FMASS1*CPFN

The second equation represents three modes of heat transfer relative to exit volume two (the volume within wall node DT(TCN) of the pump manifold).

1a. Convection to or from the piston node.

B3*(DT(TPN)-DT(TFP2))

where B3 is the convection coefficient equal to D(UP2P)*ASP2P.

1b. Convection to or from the pump walls at the exit chamber of the pump manifold, node DT(TCN)

B8*(DT(TCN)-DT(TFP2))

where B8 is equal to UP2C*ASP2C

2. Heat transfer due to mass transfer into the fluid volume from the inlet volume.

$\dot{M}$Cp*(DT(TFP1)-DT(TFP2))

where $\dot{M}$Cp is equal to Q(L2)*RHOIL*CPFN

3. Heat added directly to the fluid due to compression, friction, and the piston moving parts.

D(HTREJ)*.323

where D(HIREJ) is defined in the Technical Summary.

These heat transfer modes are combined to produce the heat balanace equation for the pump exit fluid node.

$$\frac{MCp}{DELT}*(DT(TFP2)-DT(TFP2)_{OLD}) = B3*(DT(TPN)-DT(TFP1)) + \dot{M}Cp* \quad (2)$$
$$(DT(TFP1)-DT(TFP2)) + .323 * D(HTREJ)$$
$$+B8*(DT(TCN)-DT(TFP2))$$

The third equation represents three modes of heat transfer relative to volume three within the pump case.

1. Heat transfer due to mass transfer into the case volume three from the inlet, and exit volumes respectively. (Leakage flows).

DT(QLEAK1)*CPFN*(DT(TFP1)-DT(TFP3)) and

DT(QLEAK2)*CPFN*(DT(TFP2)-DT(TFP3))

where DT(QLEAK1) is equal to D(COECIN)*(P(L1)-P(L3)) and DT(QLEAK2) is equal to D(COEPLK)*(P(L2)-P(L3)).

2a. Convection to or from the pump mass node around the case.

B5*(DT(TDN)-DT(TFP3))

and B5 is equal to UP3D*ASP3D

2b. Convection to or from the piston mass node

B2*(DT(TPN)-DT(TFP3))

where B2 is equal to UP3D*ASP3P

3. Heat added to the fluid due to the heat rejection term

.24*D(HTREJ)

where D(HTREJ) was defined previously.

These heat transfer terms combine to produce the heat balance equation for the fluid volume 3 in the case drain.

$$\frac{\dot{M}Cp}{DELT}*(DT(TFP3)-DT(TFP3)_{OLD}) = DT(QLEAK1)*CPFN*(DT(TFP1)-DT(TFP3)) \quad (3)$$
$$+DT(QLEAK2)*CPFN*(DT(TFP2)-DT(TFP3))$$
$$+B2*(DT(TPN)-DT(TFP3))+B5*(DT(TDN)-$$
$$DT(TFP3)) + .24*D(HTREJ)$$

where $\dot{M}Cp$ is equal to FMASS3*CPFN

The fourth equation represents four modes of heat transfer relative to the pump wall mass (inlet manifold mass) around the inlet volume.

1a. Conduction to or from the pump wall node around the exit volume.

(manifold node DT(TCN))

R9*(DT(TCN)-DT(TBN))

where R9 is equal to COB/(DXB/ACB+DXC/ACC)

1b. Conduction to or from the upstream line wall segment

R3*(TW(L1)-DT(TBN))

where R3 is the conduction coefficient equal to

1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXB/(ACB*COB)

1c. Conduction to or from the pump walls around the case fluid volume.

R11*(DT(TDN)-DT(TBN))

where R11 is equal to COB/((DXB/ACB+DXD/ACD)*2.0)

1d. Conduction to or from the piston node

R5*(DT(TPN)-DT(TBN))

where R5 is equal to 1./(2.*(DXP/(D(ACP)*COP)+DXB/(ACB*COB)+

1.0/(D(ASPB)*D(CBP)))).

2a. Convection to r from the pump fluid in inlet volume, fluid

volume one.

B1*(DT(TFP1)-DT(TBN))

with B1 defined previously.

2b. Convection to or from the surrounding atmosphere

B6*(D(TA)-DT(TBN))

where B6 is equal to D(UAB)*D(ASAB)*D1

D1 is equal to D(VOL1)/(D(VOL1)+D(VOL2))

3. Heat added due to the heat rejection term

.125*D(HTREJ)

D(HTREJ) has been defined previously .

4. Radiation exchange with the surrounding structure

C2*(D(TST)-(DT(TBN)+460)**4)

where C2 is a radiation coefficient equal to C1*D1 where

C1 equals SIGMA*EPSION*SHAPF*D(ASAB) and D1 defined previously.

These heat transfer terms combine to produce the heat balance equation for the pump wall (manifold wall node B around the inlet volume).

$$\frac{MCp}{DELT}*(DT(TBN)-DT(TBN)_{OLD}) = R3+(TW(L1)-DT(TBN))+R9*(DT(TCN)-DT(TBN)) + R11*(DT(TDN)-DT(TBN)) + R5*(DT(TPN)-DT(TBN)) + B1*(DT(TFP1)-DT(TBN))+ B6*(DLTA)-DT(TBN)) + .125*D(HTREJ) +C2*(D(TST))-C2*(DT(TBN)+460.)**4 \qquad (4)$$

where MCp is equal to D(TPMASS)*CPBN*D1.

The fifth equation represents three modes of heat transfer relative to the piston node.

1a. Convection to or from the fluid in the exit chamber

B3*(DT(TFP2)-DT(TPN))

with B3 described previously

1b. Convection to or from the case fluid

B2*(DT(TFP3)-DT(TPN))

with B2 being defined previously.

2. Conduction to or from the pump manifold walls

R5*(DT(TBN)-DT(TPN))

R8*(DT(TCN)-DT(TPN))

when R5 is as defined previously.

and R8 equals 1./((DXP/(D(ACP)*COP)+DXC/(ACC*COB)

+1./(D(ASBP)*D(CBP))*2.)

3. Heat added to the piston mass from the heat rejection term

.187*D(HTREJ)

These heat transfer terms combine to produce the heat balance equation for the piston node.

$$\frac{MCp}{DELT}*(DT(TPN)-DT(TPN)_{OLD}) = B3*(DT(TFP2)-DT(TPN)) + B2*(DT(TFP3)-DT(TPN)) + R5+(DT(TBN)-DT(TPN))+.187*D(HTREJ) +R8*(DT(TCN)-DT(TPN)) \qquad (5)$$

where MCp is equal to D(PMASS)*CPPN

The sixth equation represents four modes of heat transfer relative to the pump manifold wall node surrounding the exit volume, Node C.

1a. Conduction to or from the downstream connecting line segment

R4*(TW(L2)-DT(TCN))

where R4 equals 1.0/(DXF(L2)/(ACW(L2)*C(L2)) + DXC/(ACC*COB))

1b. Conecution to or from the piston mass

R8*(DT(TPN)-DT(TCN))

where R8 was defined previously.

1c. Conduction to or from the two pump wall node manifold B (inlet, volume wall)

R9*(DT(TBN) - DT(TCN))

1d. Conduction to or from the case wall node

R10*(DT(TDN)-DT(TCN))

where R10 is equal to COB/((DXC/ACC+DXD/ACD)*2.)

and R9 was defined previously.

2a. Convection to or from the exiting fluid node

B8*(DT(TFP2)-DT(TCN))

where B8 was defined previously.

2b. Convection to or from the surrounding atmosphere

B9*(D(TA)-DT(TCN))

where B9 is equal to D(UAB)*D(ASAB)*D2

3. Heat added to the walls due to a heat rejection term,

.125*D(HTREJ)

where D(HTREJ) was defined previously.

4. Radiation exchange with the surrounding structure.

C3*(D(TST)-(DT(TCN)+460.)**4)

where C3 equals C1*D2 and C1 was defined previously, and D2 equals D(VOL2)/(D(VOL1)+D(VOL2)).

These heat transfer terms combine to produce the heat balance for the pump wall around the exit volume, manifold wall node (DT(TCN).

$$\frac{MCp}{DELT}*(DT(TCN)-DT(TCN)_{OLD}) = R4*(TW(L2)-DT(TCN))+R9*(DT(TBN)-DT(TCN))+R8*(DT(TPN)-DT(TCN))+.125*D(HTREJ)+R10*(DT(TDN)-DT(TCN)) + B9*(D(TA)-DT(TCN))+B8*(DT(TFP2)-DT(TCN)) +C3*D(TST)-C3*(DT(TCN)+460.)**4 \quad (6)$$

where MCp is equal to D(TPMASS)*CPBN*D2.

The seventh equation represents three modes of heat transfer relative to the pump walls surround the case fluid.

1a. Conduction to or from the case drain connecting line wall segment.

R7*(TW(L3)-DT(TDN))

where R7 is equal to 1.0/(DXF(L3)/(ACW(L3)*C(L3))+DXD/(ACD*COB))

1b. Conduction to or from the two other pump manifold wall nodes, around the inlet and outlet respectively.

R11*(DT(TBN)-DT(TDN))

R10*(DT(TCN)-DT(TDN))

where R10 and R11 were defined previously.

2a. Convection to or from the fluid in the case volume

B5*(DT(TFP3)-DT(TDN))

with B5 defined previously.

2b. Convection to or from the external surrounding atmosphere

B10*(D(TA)-DT(TDN))

where B10 is a convection coefficient equal to D(UAD)*D(ASAD).

3. Radiation exchange with the surrounding structure.

$$C4*(D(TST)-(DT(TDN)+460)^{4})$$

where C4 is equal to SIGMA*EPSION*SHAPF*D(ASAD)

These heat transfer terms combine to produce the heat balance equation for the case wall node.

$$\frac{MCp}{DELT}*(DT(TDN)-DT(TDN)_{OLD}) = R7*(TW(L3)-DT(TDN))+R11*(DT(TBN)-DT(TDN)) + R10*(DT(TCN)-DT(TDN)) +B5*(DT(TFP3)-DT(TDN))+B10*(D(TA)-DT(TDN))+C4*D(TST)-C4*(DT(TDN)+460)**4 \qquad (7)$$

where MCp is equal to D(PDMASS)*CPBN.

A thermal model of the above heat transfer terms for the pump is shown in Figure 6.51-3.

Equations (1) thru (7) are solved for the appropriate temperatures.

In the hydraulic math model the variable delivery pump generates fluid flow in response to system flow demand. The output pressure is a function of outlet flow. The steady state pump simulation models the pump characteristic flow versus pressure out curve (Figure 6.5-4), the characteristic leakage from high pressure to pump case, the leakage from pump case back to inlet and the pump outlet flow versus inlet pressure curve.

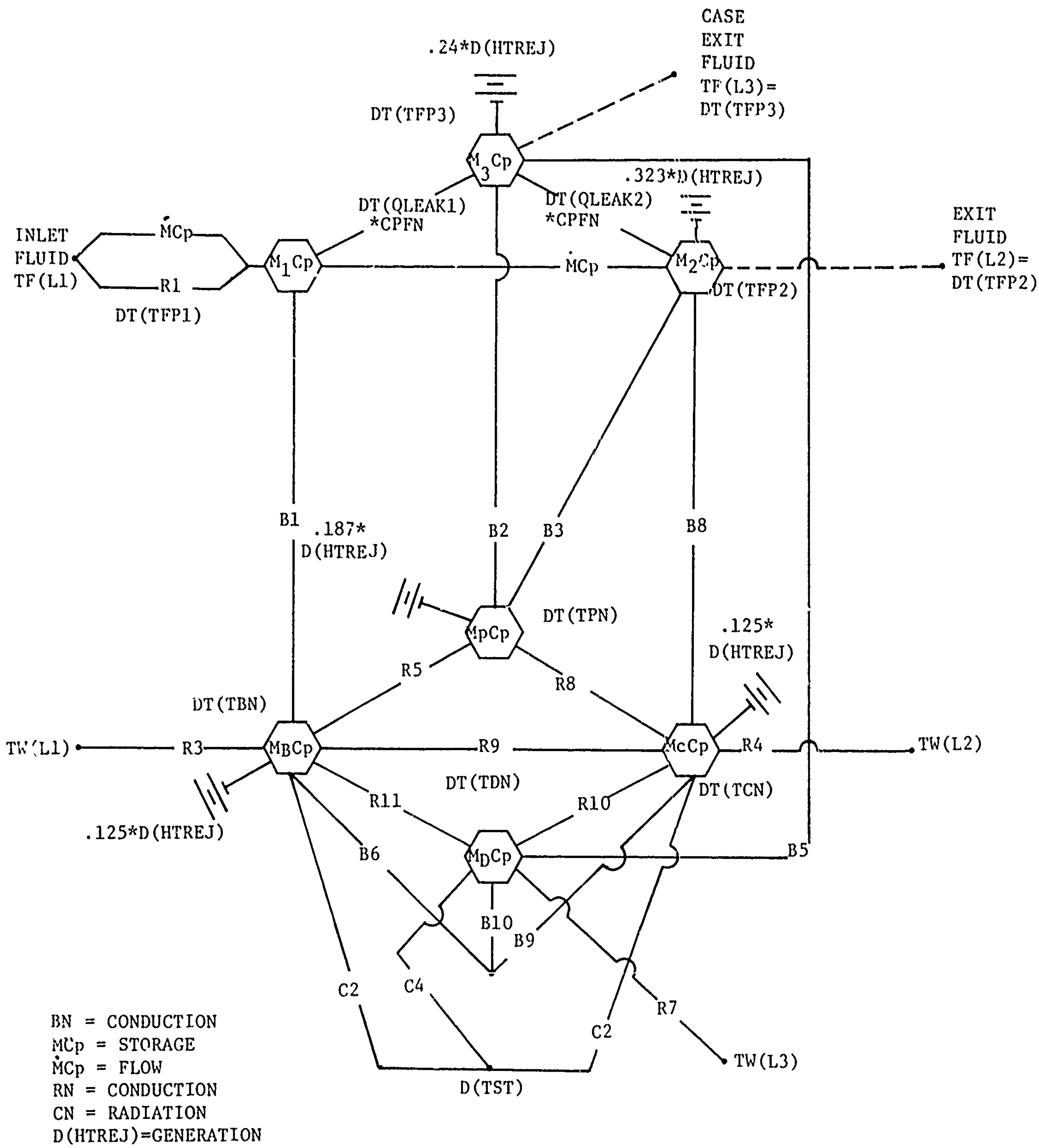


FIGURE 6.51-3

THERMAL MODEL

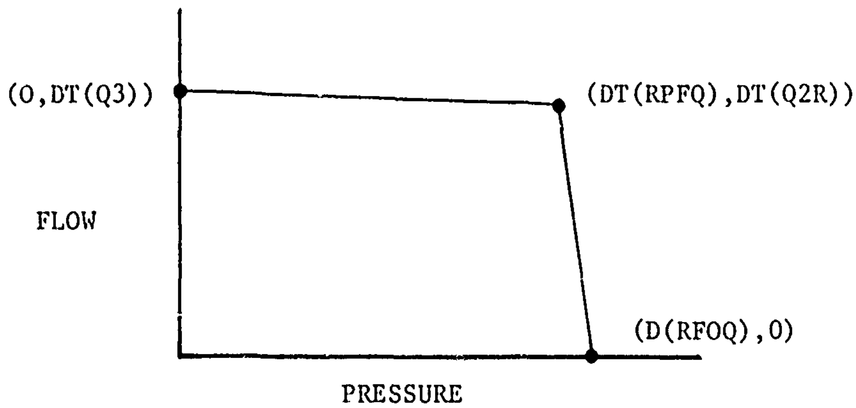


FIGURE 6.5-4

Pump Outlet Flow vs Outlet Pressure Curve

6.51.2 Assumptions

1. All internal moving parts are evaluated as one node, all at the same temperature, DT(TPN).
2. The mass of the pump walls are modeled as three nodes, two top mainfold nodes one each associated with the inlet and exit fluid volumes, and the third wall around the case volume.
3. External temperatures remain constant.
4. Interface conductances between pump wall and connecting lines is infinite.
5. The fluids leaving volumes two (exit) and volume three(case) are equal to DT(TFP2) and DT(TFP3) respectively, so there is no interaction with the downstream line fluids nodes.
6. The emissivity of the walls remains constant, .3 for steel.
7. Complete mixing occurs in the fluid volumes.

6.51.3 Computational Methods

Section 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degree Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

Section 2000

The pump subroutine is called in the order inlet, case drain and outlet. The inlet pressure is determined from the pump flow node. The pump rated flow at the operating RPM is calculated as

DT(Q2R) = D(RQ)*D(RPM)/D(RRPM)

The pump flow at zero system resistance is

DT(Q3) = 1.05*DT(Q2R)

If the inlet flow is less than the pump minimum inlet pressure D(PSMIN), a new rated flow is computed based on a straight line interpolation between 10 psia and D(PSMIN). When inlet pressure fails below 10 psia a warning message is printed.

On entry into the case drain section the leakage flow from high pressure to the pump case is calculated as

$$QPCD = D(RCDC)-(.3*(DT(QOUT)/D(RQ))$$

The flow that leaks back to the inlet from the case is a function of the case flow out the port and QP_CD. The case pressure is computed based on this flow difference.

$$DT(P_{CASE}) = D(RCDP)*(1-QQ/QPCD) + DT(PINLET)$$

Using the flow out of the pump the characteristic pressure out is calculated for flow less than DT(Q2R)

$$DT(POUTLT) = D(RPOQ)-(D(RPOQ)-D(RPFQ))*(QQ/DT(Q2R))$$

If the flow out is greater than DT(Q2R)

$$DT(POUTLT) = (DT(Q3)-QQ)*D(RPFQ)/(DT(Q2R)-DT(Q3))$$

The actual pump outlet pressure is calculated using DT(POUTLT) from the characteristic curve and adjusting this to account for the actual pressure in the case less the case pressure at which DT(POUTLT) was set.

$$POUT = DT(POUTLT) + DT(PCASE) - D(PSET)$$

<u>Section 3000</u>

Property values are assigned. Dimensions and coefficients are calculated. A 7 X 7 matrix is loaded and equations (1) through (7) are solved for DT(TFP1), DT(TFP2), DT(TFP3), DT(TBN), DT(TPN), DT(TCN) and DT(TDN). The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to TF and TC in COMMON/TRANS/.

6.51.4 Approximations

1. The heat transfer coefficients for fluid in the case to the case walls is one thirdof the coefficient from fluid in volume one to the case walls.

2. Many distances and areas are approximated.

6.51.5 Limitations

The pump model cannot handle cavitation at the inlet port.

6.51.6 Variable Listing

Variable	Description	Dimensions
ACB	Cross sectional of manifold wall node B, around the inlet volume	$IN.^2$
ACC	Cross Sectional area of manifold wall node C, around the exit volume	$IN.^2$
ACD	Cross sectional area of manifold wall node D, around the case volume	$IN.^2$
D(ACP)	Estimated cross-sectional area of the rotating group	$IN.^2$
ACP1	Estimated cross-sectional area of the inlet fluid	$IN.^2$
ACP2	Estimated cross-sectional area of the outlet fluid	$IN.^2$
ACP3	Estimated cross-sectional area of the case fluid	$IN.^2$
D(ASAB)	External surface area of the pump walls	$IN.^2$
D(ASPB)	Contact area, walls and the internal mass (pistons)	$IN.^2$
ASP1B	Surface area, inlet fluid to walls	$IN.^2$
ASP2P	Surface area, outlet fluid to pistons	$IN.^2$
ASP3P	Surface area, case fluid to walls	$IN.^2$
ASP3P	Surface area, case fluid to internal mass (pistons)	$IN.^2$
B	Dummy computational array	--

6.51.6 Variable Listing (Continued)

Variable	Description	Dimensions
B1,B2,B3, B5, B6	Dummy variables	--
D(CPB)	Interface Conductance between the piston and walls	WATTS/IN.^{2}°F
CJ	Mechanical Equivalent of Heat	FT-LB_m/WATTS-SEC
COB	Thermal conductivity of the walls	WATTS/IN-°F
COP	Thermal conductivity of the pistons	WATTS/IN.-°F
CPBN	Specific heat of the walls	WATTS-SEC/LB_m-°F
CPPN	Specific heat of the pistons	WATTS-SEC/LB_m-°F
C1	Dummy variable	--
D(DELTA)	Distance from connection one to piston chamber	IN.
D(DELTA1)	Case Depth	IN.
DXB	Distance from wall node to interface of lines	IN.
DXC	Distance from internal fluid node to interface of lines	IN.
DXD	Distance from exit fluid node to interface	IN.
DXP	Distance from piston node to interface	IN.
DXP1	Distance from fluid one node to interface with line	IN.
D1,D2	Dummy variables	--
EPSION	Emissivity factor	--
FMASS1	Inlet fluid mass	LB_m
FMASS2	Outlet fluid mass	LB_m
FMASS3	Case fluid mass	LB_m
D(HTREJ)	Heat rejection term	WATTS
D(ITF)	Initial temperature of the fluid in the pump	°F
D(ITB)	Initial temperature of the pump & piston masses	°F
LTYPE	Dummy variable	-
NTYPE	Dummy variable	-

6.51.6 Variable Listing (Continued)

Variable	Description	Dimensions
DT(PCASE)	Case Pressure	PSI
DT(PINLET)	Inlet Pressure	PSI
D(PMASS)	Piston Mass (all internal moving parts)	LB_m
DT(POUTLT)	Outlet Pressure	PSI
POUT	Dummy Variable	-
PP	Computational array	-
D(PSET)	Pump Case Pressure at rated flow and pressure	PSI
D(PSMIN)	Minimum inlet pressure	PSI
D(PTYPE)	Piston Material Type	-
DT(QCD)	Case Drain Flow	CIS
DT(QLEAK1)	Leakage flow high pressure to case	CIS
DT(QLEAK2)	Leakage flow case to inlet	CIS
Q2C	Dummy Variable	-
DT(Q2R)	Rate Flow adjusted for operating RPM	CIS
D(RCDL)	Case drain flow at rated conditions	CIS
D(RCDP)	Maximum pressure difference between pump case and inlet	PSID
RHOB	Case material density	LB_m/IN^3
RHOIL	Fluid density	LB_m/IN^3
RHOP	Rotating group material density	LB_m/IN^3
RMFL1, RMFL2, RMFL3	Dummy Variables	-
D(RPFQ)	Rated pressure at full flow	PSI
D(RPM)	Pump operating speed	RPM

6.51.6 Variable Listing (Continued)

Variable	Description	Dimensions
D(RPOQ)	Rated pressure at zero flow	PSI
D(RQ)	Rated flow	CIS
D(RRPM)	Pump speed at rated flow and pressure	RPM
RPM	Dummy variable	--
R1,R2, R3,R4, R5,R7, R10,R11	Dummy variables	--
SHAPF	Radiation shapf factor for the external walls	--
SIGMA	Stefan-Boltzmann radiation constant	WATTS/IN.^{2}-°R^4
D(TA)	Surrounding atmospheric temperature	°F
DT(TBN)	Temperature of the pump walls	°F
DT(TFP1)	Temperature of the inlet fluid	°F
DT(TFP2)	Temperature of the outlet fluid	°F
DT(TFP3)	Temperature of the case fluid	°F
D(TPMASS)	Pump wall mass	LB_m
DT(TPN)	Temperature of the internal parts, piston	°F
D(TST)	Temperature of the surrounding structure	°F
D(UAB)	External heat transfer coefficient of the pump	WATTS/IN.^{2}-°F
D(UP1B)	Heat transfer coefficient, inlet fluid to the walls	WATTS/IN.^{2}-°F
D(UP2P)	Heat transfer coefficient, outlet fluid and the piston	WATTS/IN.^{2}-°F
UP3P	Heat transfer coefficient, case fluid and the walls	WATTS/IN.^{2}-°F
D(VOL1)	Inlet volume	IN.^{3}
D(VOL2)	Outlet volume plus cylinders volume	IN.^{3}
D(VOL3)	Case volume	IN.^{3}

6.51.7 Subrouting Listing

```
      SUBROUTINE TPUMP51 (D,DT,DD,L)
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/ P(300),Q(300),C(300),TC(300),TW(300),TF(300)
     + ,ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NLL
      COMMON /COMP/ LTYPE(99),NC(99),KTEMP(99),IND,ILNTR,INIT
      COMMON /STEADY/PN(90),QN(90),PLX(90),PDLLG(90),QL(90),
     + QA,IX,Q1,PUP,PDOWN,NNODE,NLEG,NCPN,TERM,LEGN,ICON,INV,
     + INX,INZ,NUP(90),NDWN(90),NELLN(90),ILLGAD(90),ILEG(1000)
      COMMON /FLUID/ ATPRES,CF,CPFN,FP  P,PROP(13,3)
      DIMENSION PP(7,7),B(7)
      INTEGER TST,PA,VOL1,VOL2,PMASS,VOL3,TPMASS,ACP,ASAB,
     + UA3,ASPB,CBP,RPN,UP2P,DELTA,UP1B,QOUT
     + ,PTYPE,TFP1,TFP2,TFP3,TBN,TPN,QLEAK1,QLEAK2,PCASE
     + ,ASAD,PDMASS,TDN,TCN,RPFQ,RPOQ,RQ,RRPN,PSMIN,RCDP,
     + PSET,DELTA1,POUTLT,PINLET,Q3,Q2R,RCDL,HTRLJ,QCD,POUT
C       D ARRAY VARIABLES
      DATA PTYPE/1/,PTYPE/2/,TPMASS/3/,PMASS/4/,PDMASS/5/,VOL1/6/,
     + VOL2/7/,VOL3/8/,ACP/9/,ASPB/10/,UP2P/11/,HTRLJ/12/,DELTA/13/,
     + ASAB/14/,ASAD/15/,UA3/16/,CBP/17/,UP1B/18/,TST/19/,
     + PA/20/,ITF/21/,IT3/22/,
     + RQ/23/,RRPN/24/,RPN/25/,RPOQ/26/,RPFQ/27/,
     + PSMIN/28/,RCDP/29/,RCDL/30/,PSET/31/,DELTA1/32/
C       DT ARRAY VARIABLES
      DATA TFP1/1/,TFP2/2/,TFP3/3/,TBN/4/,TPN/5/,QLEAK1/6/,QLEAK2/7/
     + ,Q2R/8/,Q3/9/,PINLET/10/,TCN/11/,TDN/12/,PCASE/13/,POUTLT/14/
     + ,QCD/15/,POUT/16/
      DATA SIGMA/.349E-11/,SHAPE/.96/,EPSION/0.3/,CJ/8.85/
C     DELTA  =THE DISTANCE FROM INLET TO OUTLET THROUGH THE PISTON
C     DELTA1 =THE TOTAL DEPTH OF THE DRAIN BOWL
C     ITF&IT3=INITIAL TEMPERATURE
C      UA3   =HEAT TRANSFER COEFF. ATMOSPHERE TO CASE
C     ASAD  =SURFACE AREA EXTERNAL TO CASE DRAIN WALL
C     PDMASS =PUMP WALL MASS OF CASE DRAIN WALLS
C      CBP   =INTERFACE CONDUCTANCE, PISTON TO CASE
C     VOL1  =INLET VOLUME
C     VOL2    =EXIT VOLUME INCLUDES CYLINDER VOLUMES
C     VOL3    =CASE VOLUME
C     ASPB =CONTACT AREA INTERNAL PARTS,PISTON,TO CASE
C      ACP  =CROSS SECTIONAL AREA OF TOTAL PISTON, INTERNAL
C      MASS
C      UP2P =HEAT TRANSFER COEFFICIENT PISTON TO EXIT FLUID
C     QLEAK1   =LEAKAGE FLOW FROM INLET TO DRAIN
C     UP1B     =HEAT TRANSFER COEFF. CASE OR PISTON TO FLUIDS
C     PMASS    =PISTON MASS
C     TPMASS =PUMP MASS SURROUNDING VOLUMES 1& 2,TOP MOUNTING
C     320 BTU/MIN.=5625.WATTS
      IF(ILNTR)1000,2000,3000
C *** 1000 SECTION
 1000 CONTINUE
      L1=L(1)
```

6.51.7 (Continued)

```
      L2=L(2)
      L3=L(3)
C     INITIALIZING TEMPERATURES
      TC(L1)=D(IT3)
      TC(L3)=D(IT3)
      TF(L3)=D(ITF)
      TC(L2)=D(IT3)
      TF(L2)=D(ITF)
      TF(L1)=D(ITF)
      DT(TFP1)=D(ITF)
      DT(TCN)=D(IT3)
      DT(TDN)=D(IT3)
      DT(TFP2)=D(ITF)
      DT(TFP3)=D(ITF)
      DT(TBN)=D(IT3)
      DT(TFN)=D(ITF)
      D(TST)=(D(TST)+460.)**4
      IF(D(UP1B).EQ.0.0) D(UP1B)=.1
      IF(D(UAB).EQ.0.0) D(UAB)=.006
      IF(D(UP2P).EQ.0.0) D(UP2P)=3.0
      DT(PINLET)=-1.
      DT(QOUT)=10.
      RETURN
C *** 2000 SECTION
 2000 CONTINUE
C     THE PUMP IS CALLED IN THE ORDER
C      CON 1 - INLET: CON 3 - CASE DRAIN: CON 2 - OUTLET
      IF(ICON-2) 2100,2010,2200
C     INLET
 2100 DT(Q2R)=D(RQ)*D(RPM)/D(RRPM)
      DT(Q3)=1.05*DT(Q2R)
      IF(DT(PINLET).EQ.-1.)GO TO 2101
      DT(PINLET)=PN(NODN(INEL))
      IF(DT(PINLET).GE.D(PSMIN))RETURN
      Q2C=.1
      IF(DT(PINLET).GT.10.)Q2C=DT(Q2R)*((D(PSMIN)-DT(PINLET))
     + /(D(PSMIN)-10.))
      DT(Q2R)=Q2C
      IF(DT(PINLET).LT.10.)WRITE(6,999)
  999 FORMAT(10X,40HWARNING PUMP INLET PRESSURE BELOW 10 PSI)
      RETURN
 2101 DT(PINLET)=50.
      RETURN
 2200 IF(INX.NE.1) GO TO 2700
C     CASE DRAIN
      QQ=Q1
      IF(Q1.LT.0.0)QQ=0.0
      QPCD=D(RCDL)-(.3*(DT(QOUT)/D(RQ)))
      DT(QLEAK2)=QPCD
      IF(Q1.GT.QPCD)QQ=QPCD
```

6.51.7 (Continued)

```
      DT(PCASE)=D(RCDP)*(1.-QQ/QPCD)+DT(PINLET)
      DT(QCD)=QQ
      DT(QLEAK1)=DT(QCD)-DT(QLEAK2)
      PUP=+DT(PCASE)
      INV=0
      PDLEG(INLL)=DT(PCASE)-DT(PINLET)
      RETURN
 2010 IF(INX.NE.1) GO TO 2700
C     OUTLET
      QQ=Q1
      IF(Q1.LT.0.0)QQ=0.0
      IF(Q1.GT.DT(Q3))QQ=DT(Q3)
      DT(QOUT)=QQ
      IF(Q1.GT.DT(Q2R))GO TO 2020
      DT(POUTLT)=D(RPOQ)-(D(RPOQ)-D(RPFQ))*(QQ/DT(Q2R))
      GO TO 2030
 2020 DT(POUTLT)=(DT(Q3)-QQ)*D(RPFQ)/(DT(Q2R)-DT(Q3))
      IF(DT(POUTLT).LT.0.0)DT(POUTLT)=0.0
 2030 DT(POUT)=DT(POUTLT)+DT(PCASE)-D(PSET)
      IF(DT(POUT).LT.DT(PCASE))DT(POUT)=DT(PCASE)
      PUP=DT(POUT)
      INV=0
      PDLEG(INLL)=DT(POUT)
 2600 RETURN
 2700 WRITE(6,2800) IND,ICON,INLL
 2800 FORMAT(5X,46H CALL SEQUENCE ERROR DETECTED IN COMPONENT NO ,
     + I5,14H CONNECTION NO,I5,7HLEG NO.,I5)
      STOP 5000
C *** 3000 SECTION
 3000 CONTINUE
      L1=L(1)
      L2=L(2)
      L3=L(3)
      KTYPL=D(KTYPL)+.001
      NTYPL=D(PTYPL)+.001
      CPPN=PROP(KTYPL,1)
      CPPN=PROP(NTYPL,1)
      COB=PROP(KTYPL,3)
      COP=PROP(NTYPL,3)
      CPCN=CPBN
      CPCN=CPBN
      RHOP=PROP(NTYPL,2)
      RHOB=PROP(KTYPL,3)
C     AREAS & DISTANCES ARE ESTIMATES
      D1=D(VOL1)/(D(VOL1)+D(VOL2))
      D2=D(VOL2)/(D(VOL1)+D(VOL2))
      DXP1=D(DELTA)/4.0
      DXB=D(DELTA)/4.0
      DXC=D(DELTA)/4.0
      DXD=D(DELTA1)/2.
```

6.51.7 (Continued)

```
      DXP=D(PMASS)/(RHOP*D(ACP))
      C1=SIGMA*EPSION*SHAPF*D(ASA3)
      C2=C1*D1
      C3=C1*D2
      C4=SIGMA*EPSION*SHAPF*D(ASAD)
      RHOIL=336.4*RHO(DT(IFP1),(P(L3)+P(L1))/2.)
      FMASS1=D(VOL1)*RHOIL
      FMASS2=(D(VOL2))*RHOIL
      FMASS3=D(VOL3)*RHOIL
      CMASS=D(TPMASS)*D2
      BMASS=D(TPMASS)*D1
      ACB=BMASS/(RHO3*D(DELTA)/2.)
      ACC=CMASS/(RHOB*D(DELTA)/2.)
      ACD=D(PDMASS)/(RHO3*D(DELTA1))
      ACP1=D(VOL1)/D(DELTA)
      ACP2=D(VOL2)/(D(DELTA)/2.0)
      ACP3=D(VOL3)/(D(DELTA1)/2.0)
      ASP13=SQRT(4.*ACP1/PI)*PI*D(DELTA)
      ASP3D=SQRT(4.*ACP3/PI)*PI*D(DELTA1)
      ASP2P=SQRT(4.*ACP2/PI)*PI*D(DELTA)/2.
      ASP2C=SQRT(4.*ACP2/PI)*PI*D(DELTA)/2.
      ASP3P=SQRT(4.*ACP3/PI)*PI*D(DELTA1)
      ASP3B=SQRT(ACD*4./PI)*PI*D(DELTA)
C     HEAT TRANSFER COEFF.ARE CONSTANTS=50BTU/HR-FT2-F(.11 WATTS
C     /IN2-F) DEFAULT VALUE
      UP3B=D(UP13)/2.
      UP3D=UP3B*2./3.
      UAB=D(UA3)
      UP2C=D(UP2P)/2.0
      RMFL1=ABS(Q(L1))*RHOIL
      RMFL2=ABS(Q(L2))*RHOIL
      RMFL3=+ABS(Q(L3))*RHOIL
 3200 R1=CF/(DXF(L1)/ACF(L1)+DXP1/ACP1+(RMFL1*DELT)/(ACF(L1)
     + **2*RHOIL))
      R3=1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXB/(ACB*COB))
      R4=1.0/(DXF(L2)/(ACW(L2)*C(L2))+DXC/(ACC*COB))
      R5=1.0/((DXP/(D(ACP)*COP)+DXB/(ACB*COB)+
     + 1./(D(ASP3)*D(C3P)))*2.)
      R7=1.0/(DXF(L3)/(ACW(L3)*C(L3))+DXD/(ACD*COB))
      R9=COB/(DXB/ACB+DXC/ACC)
      R10=COB/((DXC/ACC+DXD/ACD)*2.)
      R11=COB/((DXB/ACB+DXD/ACD)*2.)
      R8=1./((DXP/(D(ACP)*COP)+DXC/(ACC*COB)+1./(D(ASP3)*D(C3P)))*2.)
      B1=D(UP13)*ASP13
      B2=UP3D*ASP3B
      B3=D(UP2P)*ASP2P
      B5=UP3D*ASP3D
      B6=D(UAB)*D(ASA3)*D1
      B8=UP2C*ASP2C
      B9=D(UA3)*D(ASA3)*D2
```

```
      31)=UAD*D(ASAD)
C      P1,P2 P3 B,P,C,D, NODES IN ORDER
      DO 3333 I=1,7
      DO 3333 J=1,7
      PP(I,J)=0.0
 3333 B(I)=0.0
 3300 PP(1,1)=FMASS1*CPFN/DELT+R1+CPFN*RMFL1+31
      PP(1,4)=-31
      B(1)=FMASS1*CPFN*DT(TFP1)/DELT+(R1+CPFN*RMFL1)*TF(L1)
      PP(2,1)=-RMFL2*CPFN
      PP(2,2)=FMASS2*CPFN/DELT+RMFL2*CPFN+33+38
      PP(2,6)=-38
      PP(2,5)=-33
      B(2)=FMASS2*CPFN*DT(TFP2)/DELT+0.323*D(HTREJ)
      PP(3,1)=-DT(QLEAK1)*CPFN
      PP(3,2)=-DT(QLEAK2)*CPFN
      PP(3,3)=FMASS3*CPFN/DELT+35+DT(QLEAK2)*CPFN
     + +DT(QLEAK1)*CPFN+32
      PP(3,7)=-35
      PP(3,5)=-32
      B(3)=FMASS3*CPFN*DT(TFP3)/DELT+0.24*D(HTRLJ)
      PP(4,1)=-31
      PP(4,4)=BMASS*CPBN/DELT+R9+R3+R5+31+36+R11
      PP(4,6)=-R9
      PP(4,7)=-R11
      PP(4,5)=-R5
      B(4)=BMASS*CPBN*DT(TBN)/DELT+36*D(TA)+R3*TW(L1)+
     + C2*D(TST)-C2*(DT(TBN)+460.)**4+0.125*D(HTRLJ)
      PP(5,6)=-R8
      PP(5,2)=-33
      PP(5,3)=-32
      PP(5,4)=-R5
      PP(5,5)=D(PMASS)*CPPN/DELT+33+R5+32+R8
      B(5)=D(PMASS)*CPPN*DT(TPN)/DELT+0.187*D(HTREJ)
      PP(6,4)=-R9
      PP(6,5)=-R8
      PP(6,7)=-R10
      PP(6,2)=-38
      PP(6,6)=CMASS*CPBN/DELT+R4+R8+R9+R10+38+39
      B(6)=CMASS*CPBN*DT(TCN)/DELT+R4*TW(L2)+39*D(TA)+C3*D(TST)
     + -C3*(DT(TCN)+460.)**4+.125*D(HTREJ)
      PP(7,3)=-35
      PP(7,4)=-R11
      PP(7,6)=-R10
      PP(7,7)=D(PDMASS)*CPBN/DELT+R7+R10+R11+B5+B10
      B(7)=D(PDMASS)*CPBN*DT(TDN)/DELT+R7*TW(L3)+310*D(TA)
     + +C4*D(TST)-C4*(DT(TDN)+460.)**4
 3600 CALL SIMULT(PP,B,7,IERROR)
      DT(TFP1)=B(1)
      DT(TFP2)=B(2)
```

6.51.7 (Continued)

```
DT(TFP3)=B(3)
DT(TBN)=B(4)
DT(TCN)=B(6)
DT(TDN)=B(7)
DT(TPN)=B(5)
TF(L2)=B(2)
TF(L3)=B(3)
TC(L1)=B(4)
TC(L2)=B(6)
TC(L3)=B(7)
END
```

6.61 SUBROUTINE TRSVR61

TRSV61 models a hypothetical constant pressure, constant temperature reservoir that can be used in test simulation work as sketched in Figure 6.61-1. The input pressure is maintained without fluctuation while the flow rates are adjusted to meet the line requirements. A maximum of four connections can be used.

This subroutine calculates the fluid and wall temperatures of the component at each connection.

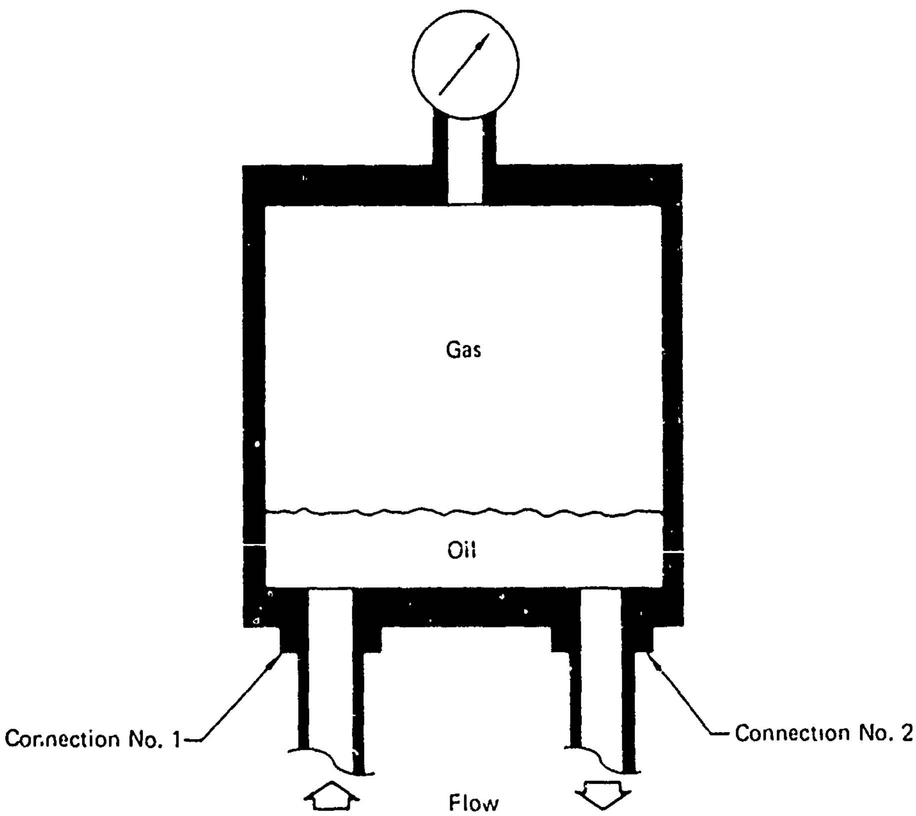


FIGURE 6.61-1
TYPE NO. 61 CONSTANT PRESSURE RESERVOIR

6.61.1 Math Model

The thermal math model for the reservoir includes heat transfer to and from one to four connecting line segments. They can be either downstream or upstream of the reservoir. To understand TRSV61 we shall look at a hydraulic system with two reservoirs connected by a line as shown in Figure 6.61-2. The line is downstream of reservoir one and

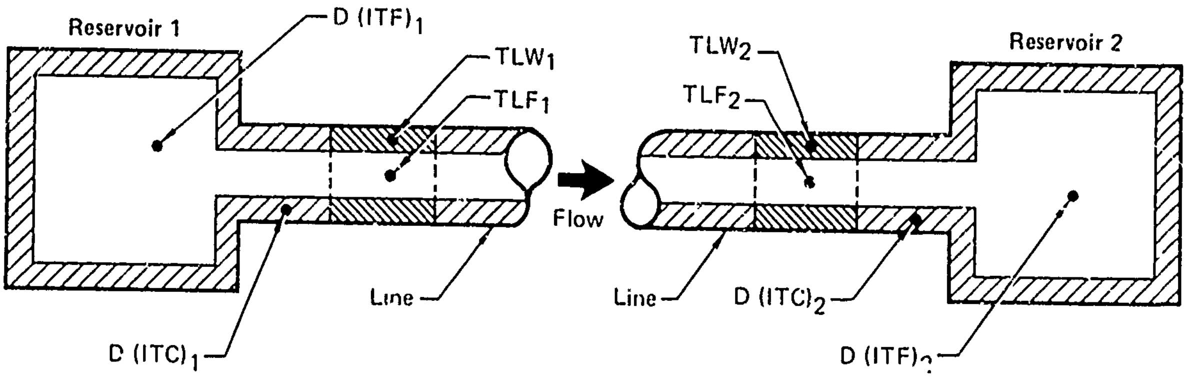


FIGURE 6.61-2
RESERVOIR NODE REPRESENTATION FOR SAMPLE SYSTEM

upstream of reservoir two. As is discussed in subroutine TLINEA. the line is divided into equal segments. In Figure 6.61-2 the temperatures of the fluid and wall of reservoir 1 are D(ITF), and D(ITC), the temperature of the fluid and wall of the line segment connecting reservoir 1 are TLF_1 and TLW_1, the temperatures of the fluid and wall of the line segment connecting reservoir 2 are TLF_2 and TLW_2, and the temperatures of the fluid and wall of reservoir 2 are $D(ITF)_2$ and $D(ITC)_2$. For downstream connecting line segments, such as these similar to segment 1 in Figure 6.61-2, the subroutine assigns the temperatures of the reservoir fluid and wall as end conditions of the reservoir, and boundary conditions of the first line segment,

$$TF(L1) = D(ITF)_1$$

$$TW(L1) = D(ITC)1$$

For upstream connecting line segments, such as those similar to segment 2 in Figure 6.61-2, the subroutine assigns the temperatures of the reservoir wall to the reservoir connection, or the boundary condition of the line segment

$$TW(L2) = D(ITC)_2$$

and also assigns the temperatures of the fluid entering the reservoir to the temperature of the line segment, $TF(L2) = TLF_2$. Note however that the temperature of the fluid in reservoir 2 is $D(ITF)_2$ and eventually the fluid entering reservoir 2 will equal TLF_2. In the hydraulic math model the input constant reservoir pressure is assigned to the reservoir node number.

6.61.2 Assumptions

1. Fluid and wall temperatures of the reservoir remain constant, except the fluid entering a reservoir makes the reservoir fluid the same temperature as the fluid in the connecting line.
2. The reservoir is assumed to have an infinitely large gas volume so that the pressure remains unchanged.
3. The interface conductance between the reservoir walls and the line walls is infinite.

6.61.3 Computation Methods

SECTION 1000

The number of active reservoir connections is determined from the NC( ) array. A DO loop is then set up to initialize all the connecting line wall and fluid temperatures.

SECTION 2000

The node number of the reservoir is determined and the flow into and/or out of the reservoir is summed for each active connection.

D(4) = D(4) + QR

Counter, L(6), is incremented by 1 each time an entry is made until the counter is equal to the number of active connections. Once the total net flow has been determined, QN(N) is calculated

QN(N) = D(PRESS) * 20.-D(4)

An external pressure array is set to a constant value

PEX(N) = 20.0

The total flow and counter, L(6), are then set to zero.

SECTION 3000

The number of connecting line segments and flow direction are first determined. Property values are assigned. The exiting fluid and all wall connection temperatures are assigned. The assigned values are put into arrays TC and TF in /TRANS/.

6.61.4 Variable Listing

Variable	Description	Dimension
D(ITC)	Initial temperature of the reservoir walls	°F
D(ITF)	Initial temperature of the reservoir fluid	°F
Q(LI)	Flow in connector I	CIS
N	Reservoir node number	--

Variable	Description	Dimensions
D(PRESS)	Pressure of reservoir	PSI
TF(LI)	Temperature of fluid leaving reservoir	°F
TC(LI)	Temperature of reservoir wall connected to connection I	°F

6.61.5 Subroutine Listing

```
      SUBROUTINE TRSVR61 (D,DT,DD,L)
C *** REVISED AUGUST 5,1975 ***
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,ITER,INEL
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     + QA,QS,QI,PUP,PDOWN,NNODE,NLEG,NCPJ,TURN,
     +  LEGN,ICON,INV,INX,INS,NUP(90),NDWN(90),NELEM(90),
     +ILEGAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CP,CPFW,FTEMP,PROP(13,3)
      INTEGER PRESS
      DIMENSION D(1),DT(1),DD(1),L(1)
      DATA PRESS/1/,ITF/2/,ITC/3/
      IF(ITER) 1000,2000,3000
 1000 CONTINUE
      D(4)=0.0
      NCI=NC(IND)
      L(5)=NCI
      L(6)=0
      DO 1010 I=1,NCI
      N=L(I)
      TF(N)=D(ITF)
      TC(N)=D(ITC)
 1010 CONTINUE
      RETURN
 2000 CONTINUE
      N=NDWN(INEL)
      QP=QI
C     INX = ELEMENT NUMBER IN LEG
      IF(INX.NE.1) GO TO 1500
      QP=-QI
      N=NUP(INEL)
 1500 D(4)=D(4)+QP
      L(6)=L(6)+1
C     WRITE(6,999)INEL,N,QI,PUP
  999 FORMAT(10X,*RSVR*,2I10,2E12.5)
      IF(L(6).NE.L(5))RETURN
      QN(N)=D(PRESS)*20.-D(4)
      PEX(N)=20.0
      D(4)=0.0
      L(6)=0
C     PN(N)=D(PRESS)
      RETURN
 3000 CONTINUE
      DO 3009 I=1,NCI
      N=L(I)
      IF(Q(N).GT.0.0) GO TO 3006
      TF(N)=D(ITF)
 3006 TC(N)=D(ITC)
 3009 CONTINUE
      RETURN
      END
```

6.62 SUBROUTINE TRSVR62

TRSVR62 simulates a bootstrap reservoir. The subroutine can accommodate up to four low pressure lines along with a high pressure (bootstrap) line, as shown in Figure 6.62-1. The calculated variables are, the reservoir fluid, wall and piston temperatures.

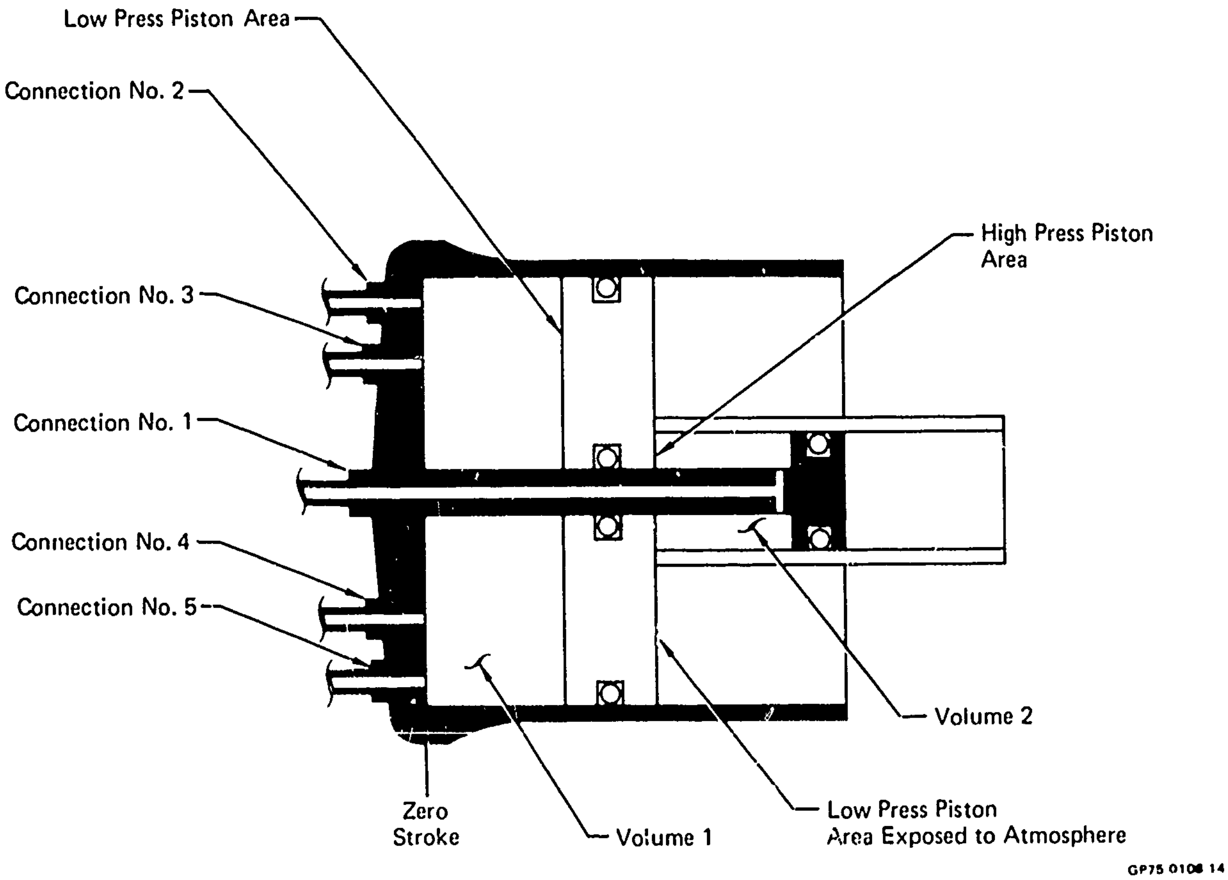


Figure 6.62-1

TYPE NO. 62 BOOTSTRAP RESERVOIR

6.62.1 Math Model

The thermal math model for the reservoir includes heat transfer from three to five connecting lines, one high pressure line, at least one upstream line and at least one downstream line. Three reservoir nodes are considered, one wall, one piston, and one fluid node. There are two nodes, one fluid and one wall for each connecting line segment, and in our model we will consider five lines or ten nodes, so that there are a total of thirteen nodes in the math model, as shown in Figure 6.62-2.

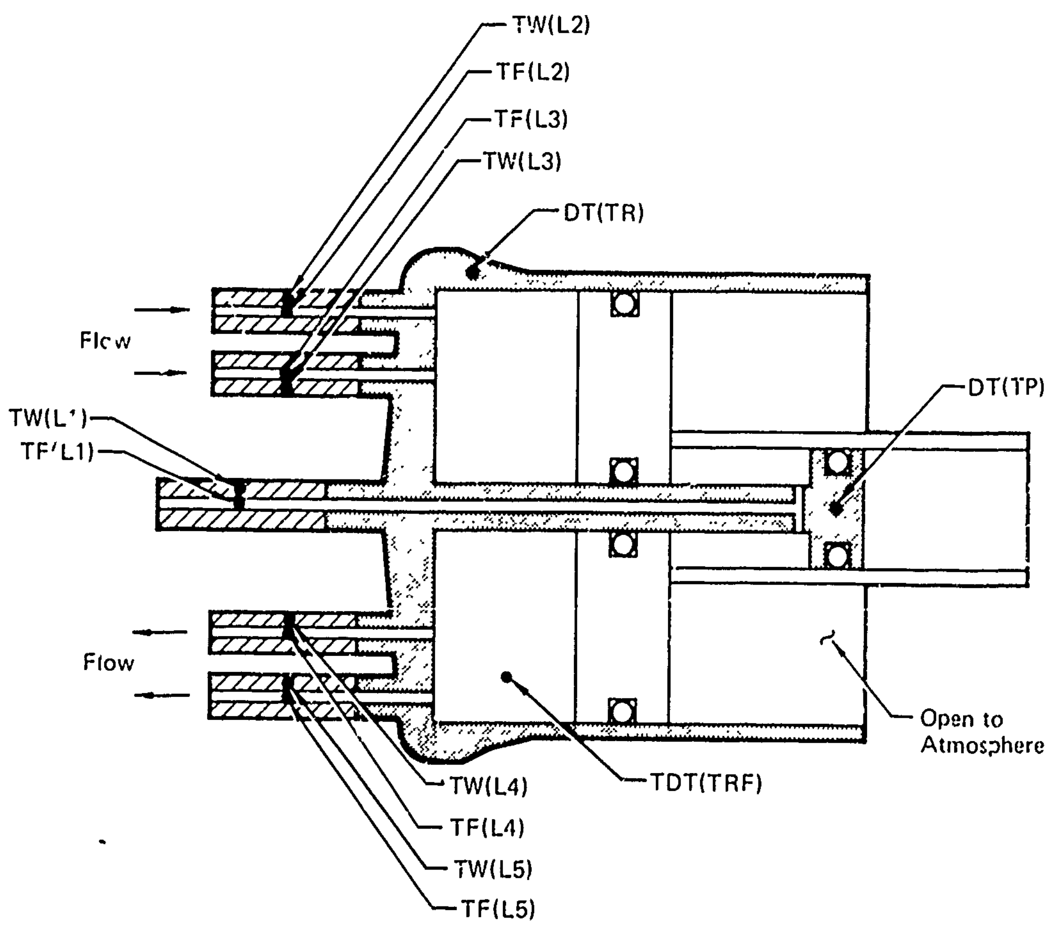


FIGURE 6.62-2
BOOTSTRAP RESERVOIR AND LINE NODE REPRESENTATION

GP77-0065-20

The temperatures of the high pressure line segment nodes are TW(L1), and TF(L1) for the wall and fluid respectively. The two upstream line node temperatures are TW(L2), TW(L3), TF(L2), and TF(L3) for the wall and fluids, and the two downstream connecting line segment node temperatures are TW(L4), TW(L5), TF(L4) and TF(L5) for the wall and fluids respectively. The reservoir fluid temperature is DT(TRF), the wall temperature is DT(TR), and the piston temperature is DT(TP). Not every case has five connecting lines, but for this discussion there will be.

Three heat balance equations are written to solve for the through reservoir node temperatures, using the reservoir and line segment material properties and dimensions, the external atmosphere and structure temperatures, and TW(L1). TW(L2), TW(L3), TW(L4), TW(L5), TF(L1), TF(L2), and TF(L3) (note: TF(L4) and TF(L5) equal DT(TRF), see assumptions).

The first equation represents two modes of heat transfer relative to the reservoir fluid.

1a) convection to and from the reservoir walls

B1 * (DT(TR)-DT(TRF))

where B1 is a convection coefficient equal to

DT(ASFR) * D(UFR)

1b) convection to and from the piston node

B2 * (DT(TP)-DT(TRF))

where B2 is also a convection coefficient equal to

D(AREA1)*D (UFR)

2) heat transfer due to mass transfer into the reservoir volume from the connecting lines segment $\dot{M}$Cp*(TF(L2)-DT(TRF)) for line two and $\dot{M}$Cp*(TF(L3)-DT(TRF) for line three where $\dot{M}$Cp is the mass flow rate term equal to RMF(I)*CPFN and RMF(I) is equal to Q(L(I))*RHOIL with I=2 for line two and I=3 for line three.

These heat transfer terms then combine to produce the heat balance for the reservoir fluid node

$$\frac{MCp}{DELT}*(DT(TRF)-DT(TRF)_{OLD}) = B1*(DT(TR)-DT(TRF))+B2*(DT(TP)-DT(TRF)) + \dot{M}Cp*(TF(L2)-DT(TRF))+\dot{M}Cp*(TF(L3)-DT(TRF)) \quad (1)$$

where MCp is equal to FMASS*CPFN

The second equation represents three modes of heat transfer relative to the reservoir wall node.

1a) convection to and from the reservoir fluid

B1*(DT(TRF)-DT(TR))

where B1 was defined previously.

b) convection to and from the external atmosphere

B3*(D(TA)-DT(TR))

where B3 is equal to D(ASAR)*D(UAR)

2a) conduction to and from the connecting line segment walls

R(I) * (TW(L(I))-DT(TR))

where R(I) is the conduction coefficient equal to

1.0/(DXF(L(I))/(ACF(L(I))*C(L(I)) +DXR/(ACR*CR))

and I= 1 to 5 for each of the five connecting lines considered.

b) conduction to and from the piston node

R9*(DT(TP)-DT(TR))

where R9 is equal to 1.0/(DXR/(ACR*CR)+DXP/(ACP*CP)+1.0/(ACP*2.0*D(CRP))).

3. radiation exchange with the surrounding external structure

CIP*(D(TST)-(DT(TR)+460)**4)

where CIP is a radiation coefficient equal to SIGMA*EPSION*SHAPF*D(ASAR)

These terms then combine to produce the heat balance for the reservoir wall

$$\frac{MCp}{DELT} * (DT(TR)-DT(TR)_{OLD}) = B1*(DT(TRF)-DT(TR))+B3*(D(TA)-DT(TR)) + \sum_{I=1}^{5} R(I)*(TW(L(I))-DT(TRF)) + R9*(DT(TP)-DT(TR)+CIP*D(TST)-CIP*(DT(TR)+460.)**4$$

where MCp is equal to D(RMASS)*CPRW

The third equation represents two modes of heat transfer relative to the reservoir piston

1a) convection to and from the reservoir fluid node

B2* (DT(TRF)-DT(TP))

with B2 defined previously

b) convection to and from the high pressure fluid

B5*(TF(L1)-DT(TP))

where B5 is equal to D(UAR)*D(AREA2)

c) convection to and from the external atmosphere

B4*(D(TA)-DT(TP))

where B4 is equal to D(ASAP)*D(UAR)

2. conduction to and from the reservoir wall node

R9*(DT(TR)-DT(TP))

where R9 has been defined previously

3. radiation exchange with the surrounding structure

C1PP*(D(TST) - (DT(TP)+460.)**4)

where C1PP is a radiation coefficient equal to

SIGMA*SHAPF*EPSION*D(ASAP)*.69.

.69 was used since the shape factor for the piston is

not nearly equal to .96 but .96*.69 = .6624

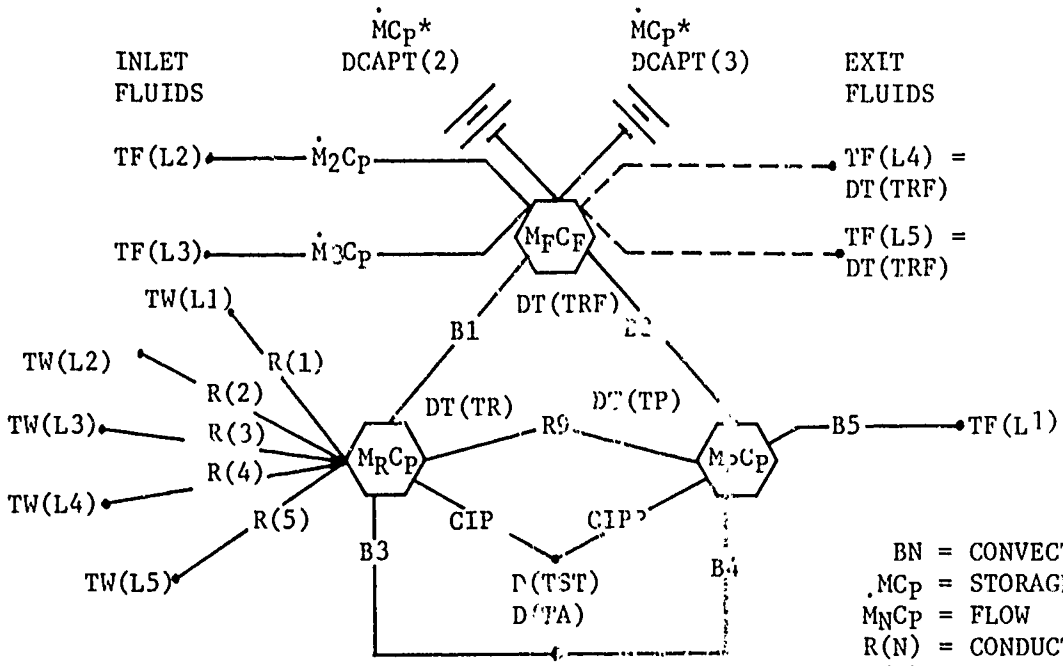
These terms than combine to produce the heat balance equation for the piston

$$\frac{MCp}{DELT} * (DT(TP)-DT(TP)_{OLD}) = B2*(DT(TRF)-DT(TP))+B5*(TF(L1)-DT(TP)) + B4*(D(TA)-DT(TP))+R9*(DT(TR)-DT(TP))+C1PP*D(TST)-C1PP*(DT(TP)+460.)**4 \qquad (3)$$

where MCp is equal to PMASS*CPPW

Equations (1), (2), and (3) are solved for the appropriate temperatures

A thermal model of the above heat transfer terms for the reservoir is shown in Figure 6.62-3.



BN = CONVECTION
MCp = STORAGE
M_NCp = FLOW
R(N) = CONDUCTION
CIP(N) = RADIATION
$\dot{M}Cp$*DCAPT(N) = GENERATION

FIGURE 6.62-3

THERMAL MODEL

The steady state high and low reservoir pressures are determined as follows.

A sign convention is established such that flow into the low pressure end is positive. A pseudo leg (see Figure 6.62-4) that terminates at low pressure node N is established. The pressure at the external end of this leg is set as the average of the pressure at node N PN(N) and pressure calculated for node N using the piston area ratio times pressure at node M, PMN where:

PMN = PN(M)*DT(NAREAR)+DT(EXPRES)

Any difference in PMN and PN(N) will produce flow in the pseudo leg and hence an unbalanced system. As TLEGCAL balances the flows at all system nodes, the pseudo leg flow is forced to zero which in turn forces PMN and PN(N) pressures to be equal.

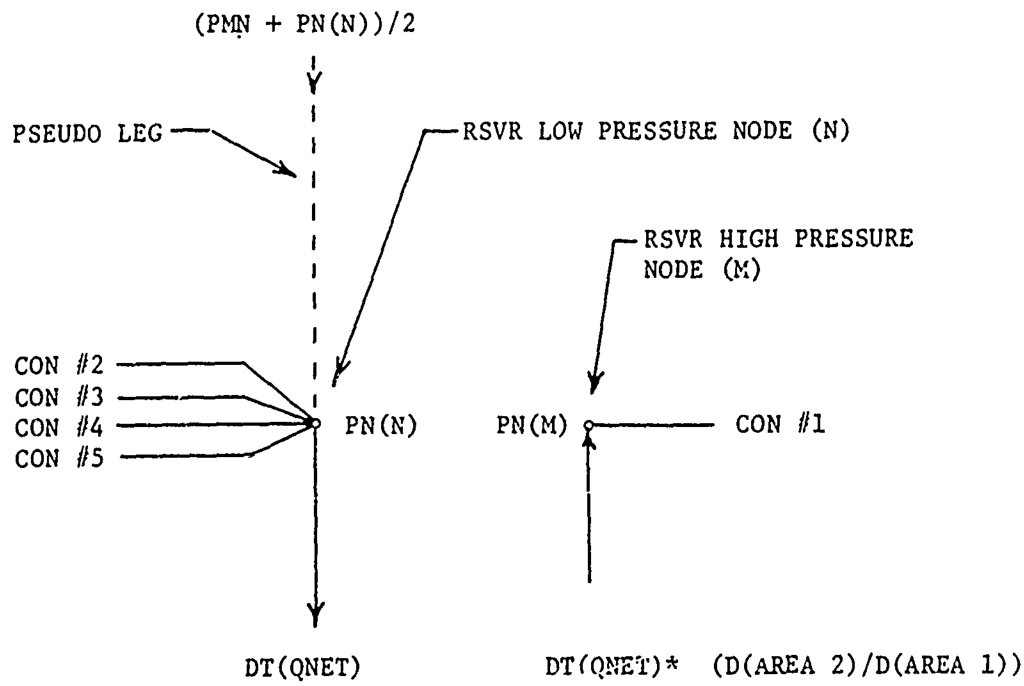


Figure 6.62-4

RSVR62 STEADY STATE FLOW DIAGRAM

6.62.2 Assumptions

1. The fluids exiting from the reservoir are equal to the calculated value of the fluid in the reservoir, DT(TRF).
2. The emissivity of the walls remain constant, .3
3. The entire mass of the reservoir walls is at the same temperature
4. The temperatures external to the reservoir remain constant
5. The interface conductance between the reservoir walls and the line segment walls is infinite.
6. Seal friction is zero
7. Complete mixing occurs in the fluid volume.

6.62.3 Computational Methods

SECTION 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

SECTION 2000

This section sums the flows into and/or out of the low pressure chamber as the entry is called for each active connection. It also determines overboard flow at the high pressure node (M) and low pressure node (N).

SECTION 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connection lines four and five (L4) and (L5). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection lines three and two). Some coefficients are then recalculated if the flow is reassigned. A 3x3 matrix is loaded and the mathematical equations are solved for DT(TRF), DT(TR) and DT(TP) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to arrays (TC and TF) in common /TRANS/ for distribution throughout the entire program.

6.62.4 Approximations

1. Shape factor for the piston, SHAPF is multiplied by .69 since this is a good representation of the real shape factor piston to the surrounding structure, .96*.69 = .6224

6.62.5 Limitations

Reservoir 62 is limited to four low pressure connections and one high pressure connection.

6.62.6 Variable Listing

Variable	Description	Dimensions
A( )	Dummy computational array	
ACP	Cross sectional area of the piston	$IN.^2$
ACR	Cross sectional area of the reservoir walls	$IN.^2$
D(AREA1)	Piston surface area, low pressure side	$IN.^2$
D(AREA2)	Piston surface area, high pressure side	$IN.^2$
D(ASAP)	External surface area of piston	$IN.^2$
D(ASAR)	External surface area of the reservoir walls	$IN.^2$
DT(ASFR)	Internal surface area of the reservoir walls	$IN.^2$
B( )	Dummy computational array	
B1,B2,B3,B4, B5	Dummy variables	
C1P	Radiation coefficient for the reservoir	
C1PP	Radiation coefficient for the piston	
CJ	Mechanical equivalent of heat	$IN\text{-}LB_m$/WATTS-SEC
CP	Thermal conductivity of the piston node	WATTS/IN.-°F

Variable	Description	Dimension
CPPW	Specifc heat of the piston node	WATTS-SEC/LB_m-°F
CPRW	Specific heat of the reservoir walls	WATTS-SEC/LB_m-°F
CR	Thermal conductivity of the reservoir walls	WATTS/IN-°F
D(CRP)	Interface conductance between the piston & reservoir	WATTS/IN^2-°F
DXP	Distance from the piston node to the wall interface	IN.
DXR	Distance from the reservoir node to the line segment wall interface	IN.
EPSION	Emissivity of the walls	
PMASS	Reservoir fluid mass	LB_m
D(ITF)	Initial temperature of the fluid	°F
D(ITR)	Initial temperature of the reservoir walls	°F
KTYPE	Dummy variable	--
D(MTYPE)	Reservoir material type	--
D(PERC)	Percentage of DCAPT added to fluid	--
D(PHEIGHT)	Piston height	IN.
PMASS	Piston material mass	LB_m
D(PTHICK)	Piston wall thickness	IN.
D(PTYPE)	Piston material type	
R	Dummy array	
RHOIL	Density of the fluid	$LB_m/IN.^3$
RHOP	Density of the piston mass	$LB_m/IN.^3$
RHOR	Density of the reservoir mass	$LB_m/IN.^3$
D(RMASS)	Mass of the reservoir walls	LB_m
R1,R9	Dummy variables	--

Variable	Description	Dimensions
SHAPF	Shape factor for the walls	--
SIGMA	Stefan-Boltzmann radiation constant	WATTS/IN^2-°R^4
D(STROKE)	Total piston stroke	IN.
D(TA)	Temperature of the surrounding atmosphere	°F
TFO	Dummy variable	--
DT(TP)	Temperature of the piston mode	°F
DT(TR)	Temperature of the reservoir wall node	°F
DT(TRF)	Temperature of the reservoir fluid node	°F
D(TST)	Temperature of the surrounding structure	°F
D(VAR)	Heat transfer coefficient external to the reservoir	WATTS/IN^2-°F
D(VFR)	Heat transfer coefficient internal to the reservoir	WATTS/IN^2-°F
DT(VOLUME)	Calculated volume of the reservoir	IN.^{3}
D(VOL1)	Initial volume, low pressure side	IN.^{3}
D(VOL2)	Initial volume, high pressure side	IN.^{3}

6.62.7 Subroutine Listing

```
      SUBROUTINE PRSVR62 (D,DT,DD,L)
C**** REVISED JANUARY 24, 1976 ****
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     +ACF(300),ACW(300),DXF(300),TIME,DELT,FI,NLINE,NLL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INLL
      COMMON /STEADY/PN(90),QN(90),PLX(90),PDLEG(90),QL(90),QA,QS,Q1,
     + PUP,PDOWN,NNODE,NLEG,NCPN,PERM,LEGN,ICON,INV,INX,INZ,
     +NUP(90),NDWN(90),NLLEG(90),
     +ILEGAD(99),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION D(1),DT(1),DD(1),L(1)
      DIMENSION A(3,3),B(3),RHF(6),TFO(6),R(6),DCAPT(5)
      INTEGER AREA1,AREA2,VOL1,VOL2,STROKE,QNET,EXPRES,P2,
     + RMASS,ASAR,UAR,UFR,PHEIGHT,PTHICK,ASAP,PLRC,
     + MTYPE,PTYPE,TA,TST,ITF,ITR,ASFR,VOLUME,TR,TP,TRF,CRP
      DATA AREA1/4/,AREA2/5/,VOL1/6/,VOL2/7/,INPOS/8/,MTYPE/1/
     + ,PTYPE/2/,RMASS/3/,PHEIGHT/9/,PTHICK/10/
     + ,ASAR/11/,ASAP/12/,UAR/13/,UFR/14/,PLRC/17/,STROKE/16/
     + ,TST/18/,TA/19/,ITF/20/,ITR/21/,CRP/15/
C     DT ARRAY VARIABLES
      DATA QNET/1/,EXPRES/2/,MAREAR/3/,P2/4/,TRF/5/,TR/6/,TP/7/,
     + ASFR/8/,VOLUME/9/,MAXVOL/10/
      DATA SIGMA/.3490E-11/,SHAPE/1.0/,EPSION/.3/,CJ/3.35/
      IF(IENTR) 1000,2000,3000
 1000 CONTINUE
C     PTYPE  =PISTON MATERIAL TYPE
C     MTYPE  =RSVR CASE MATERIAL TYPE
C     AREA1  =PISTON AREA(PHEIGHT*PWIDTH)
C     AREA2  =HIGH PRESSURE AREA
C     RMASS  =RSVR CASE MASS (LBS.)
C     NCI  =NUMBER OF CONNECTIONS (LINES TOTAL)
C     ASAP   =SURFACE AREA PISTON TO AMBIENT
C     VOL1   =VOLUME OF FLUID IN CHAMBER AT ZERO STROKE
C     VOL2   = VOLUME OF FLUID IN HIGH PRESSURE CHAMBER AT
C               ZERO STROKE
C     STROKE   = MAX PISTON STROKE
      D(TST)=(D(TST)+460.)**4
C     INITALIZING TEMPERATURES
      NCI=NC(IND)
      DO 1001 I=1,NCI
      TF(L(I))=D(ITF)
      TC(L(I))=D(ITR)
 1001 CONTINUE
      DT(TRF)=D(ITF)
      DT(TR)=D(ITR)
      DT(TP)=D(ITR)
      NTYPE=D(PTYPE)+.001
      RHOP=PROP(NTYPE,2)
      IF(D(UAR).EQ.0.0) D(UAR)=.0056
      IF(D(UFR).EQ.0.0) D(UFR)=.09
```

6.62.7 (Continued)

```
      DT(VOLUME)=D(AREA1)*D(IMPOS)
      DT(MAXVOL)=D(STROKE)*D(AREA1)+D(VOL1)
      DT(ASFR)=D(AREA1)+2.*DT(VOLUME)/D(AREA1)*D(PHEIGHT)
     ++2.*DT(VOLUME)/D(PHEIGHT)
      IF(D(ASAP).EQ.0.0) D(ASAP)=D(AREA2)*.96
      D(RMASS)=D(RMASS)*.93-D(PTHICK)*(D(AREA1)+D(AREA2))*RHOP
      DT(NAREAR)=D(AREA2)/D(AREA1)
      DT(EXPPLS)=ATPRES*(D(AREA1)-D(AREA2))/D(AREA1)
      DT(ONLT)=0.0
      L(7)=0
      L(6)=NC(IND)
      RETURN
C*****STEADY STATE SECTION*****
C
 2000 CONTINUE
C
C     M IS THE BOOTSTRAP NODE, N IS THE LOW PRESSURE NODE
      L(7)=L(7)+1
      IF(ICON.NE.1) GO TO 2600
      M=NDWN(INEL)
      L(8)=M
      IF(INX.NE.1) GO TO 2800
      M=NUP(INEL)
      L(8)=M
      GO TO 2800
 2600 QR=Q1
      N=NDWN(INEL)
      L(9)=N
      IF(INX.NE.1) GO TO 2700
      QR=-QR
      N=NUP(INEL)
      L(9)=N
 2700 DT(ONLT)=DT(ONLT)+QR
 2800 IF(L(6).NE.L(7)) RETURN
      M=L(8)
      N=L(9)
      IF(N.EQ.M) WRITE(6,2900)
      IF(PN(M).EQ.0.0) PN(M)=3000.
      IF(PN(N).EQ.0.0) PN(N)=PN(M)*DT(NAREAR)+DT(EXPPLS)
      PNN=PN(M)*DT(NAREAR)+DT(EXPPLS)
      QN(N)=((PNN+PN(N))*20.)/2.-DT(ONLT)
      QN(M)=DT(ONLT)*DT(NAREAR)
      DT(P2)=PN(N)
      PLX(N)=20.
      DT(ONLT)=0.0
      L(7)=0
 2900 FORMAT(5X,45HRSVR62 REQUIRES TWO NODES FOR BOOTSTRAP FLOW    )
      RETURN
 3000 CONTINUE
      KTYPE=D(KTYPE)+.001
```

6.62.7 (Continued)


```
      NTYPL=D(PTYPL)+.001
      CR=PROP(KTYPL,3)
      CP=PROP(NTYPL,3)
      CPRW=PROP(KTYPL,1)
      CPPW=PROP(NTYPL,1)
      RHOR=PROP(KTYPL,2)
      NCI=NC(IND)
      RHOIL=336.4*RHO(DT(TRF),DT(P2))
      DO 3003 I=2,NCI
 3003 DT(VOLUME)=DT(VOLUME)+Q(L(I))*DELT
      IF(DT(VOLUME).GT.DT(MAXVOL))DT(VOLUME)=DT(MAXVOL)
      IF(DT(VOLUME).LT.D(VOL1))WRITE(6,900)
  900 FORMAT(10X,35H**** WARNING THE RESERVOIR IS EMPTY,
     + 19H-PROGRAM CONTINUING,//)
      IF(DT(VOLUME).LT.D(VOL1))DT(VOLUME)=D(VOL1)
      RMASS=DT(VOLUME)*RHOIL
      DXR=D(STROKE)/2.0
      ACR=D(RMASS)/(RHOP*D(STROKE))
      DXP=D(PHEIGHT)/2.0
      ACP=D(PTHICK)*(D(AREA1)+D(AREA2))/D(PHEIGHT)
      PMASS=(D(AREA1)+D(AREA2))*D(PTHICK)*RHOP
      DT(ASFR)=D(AREA1)+2.*DT(VOLUME)/D(AREA1)*D(PHEIGHT)
     ++2.*DT(VOLUME)/D(PHEIGHT)
      B1=DT(ASFR)*D(UFR)
      B2=D(AREA1)*D(UFR)
      B3=D(ASAR)*D(UAP)
      B4=D(ASAP)*D(UAR)
      B5=D(UAW)*D(AREA2)
      R9=1.0/(DXR/(ACR*CR)+DXP/(ACP*CP)+1.0/(ACP*2.0*D(CRP)))
      CIP=SIGMA*SHAPE*EPSION*D(ASAR)
      CIPP=SIGMA*SHAPE*EPSION*D(ASAP)*.69
      RMF(L(1))=ABS(Q(L(1)))*RHOIL
      DO 3020 I=2,NCI
      TFO(L(I))=TF(L(I))
 3020 CONTINUE
C     FLUID, WALLS, PISTON, NODES IN ORDER
      A(1,1)=0.0
      A(2,2)=0.0
      B(1)=0.0
      B(2)=0.0
      DO 3700 I=2,NCI
      RMF(I)=Q(L(I))*RHOIL
      IF(Q(L(I)).LE.0.0) RMF(I)=0.0
      DCAPT(I)=(1./RHOIL)*(P(L(I))-P(L1))/(CJ*CPFW)
      R(I)=1.0/(DXF(L(I))/(ACW(L(I))*C(L(I)))+DXR/(ACR*CR))
      A(1,1)=A(1,1)+RMF(I)*CPFW
      A(2,2)=A(2,2)+R(I)
      B(1)=B(1)+RMF(I)*CPFW*TF(L(I))
      B(2)=B(2)+R(I)*TW(L(I))+RMF(I)*CPFW*DCAPT(I)
 3700 CONTINUE
```

6.62.7 (Continued)

```
      R1=1.0/(DXF(L(1))/(ACW(L(1))*C(L(1)))+DXR/(ACR*CR))
      B(2)=B(2)+D(RMASS)*CPRW*DT(TR)/DELT+R1*TW(L(1))+R3*D(TA)
     + +CIP*D(TST)-CIP*((D(TR)+460.)**4)
      B(1)=B(1)+FMASS*CPFW*DT(TRF)/DELT
      A(1,1)=A(1,1)+FMASS*CPFW/DELT+R1+R2
      A(1,2)=-R1
      A(1,3)=-R2
      A(2,1)=-R1
      A(2,2)=A(2,2)+D(RMASS)*CPRW/DELT+R1+R3+R9
      A(2,3)=-R9
      A(3,1)=-R2
      A(3,2)=-R9
      A(3,3)=PMASS*CPPW/DELT+R2+R4+R9+R5
      B(3)=PMASS*CPPW*DT(TP)/DELT+R4*D(TA)+CIPP*D(TST)-CIPP*((D(TP)
     + +460.)**4)+R5*TF(L1)
      CALL SIMULT(A,3,3,IERROR)
      DT(TRF)=B(1)
      DT(TR)=B(2)
      DT(TP)=B(3)
      DO 3800 I=2,NCI
      TF(L(I))=B(1)
      TC(L(I))=B(2)
      IF(D(L(I)).GE.0.0) TF(L(I))=TFO(L(I))
 3800 CONTINUE
      RETURN
      END
```

6.69 SUBROUTINE THEX69

Subroutine THEX69 simulates a variety of heat exchanger configurations which includes shell, tube and flat plate types. Each can have unidirection flow, counter flow, or cross flow, as shown in Figure 6.69-1.

The subroutine calculates the exterior wall temperature, the hydraulic fluid and cooling liquid temperatures, and the interior wall temperature, of either pipes, fins or flat plates (whichever is considered).

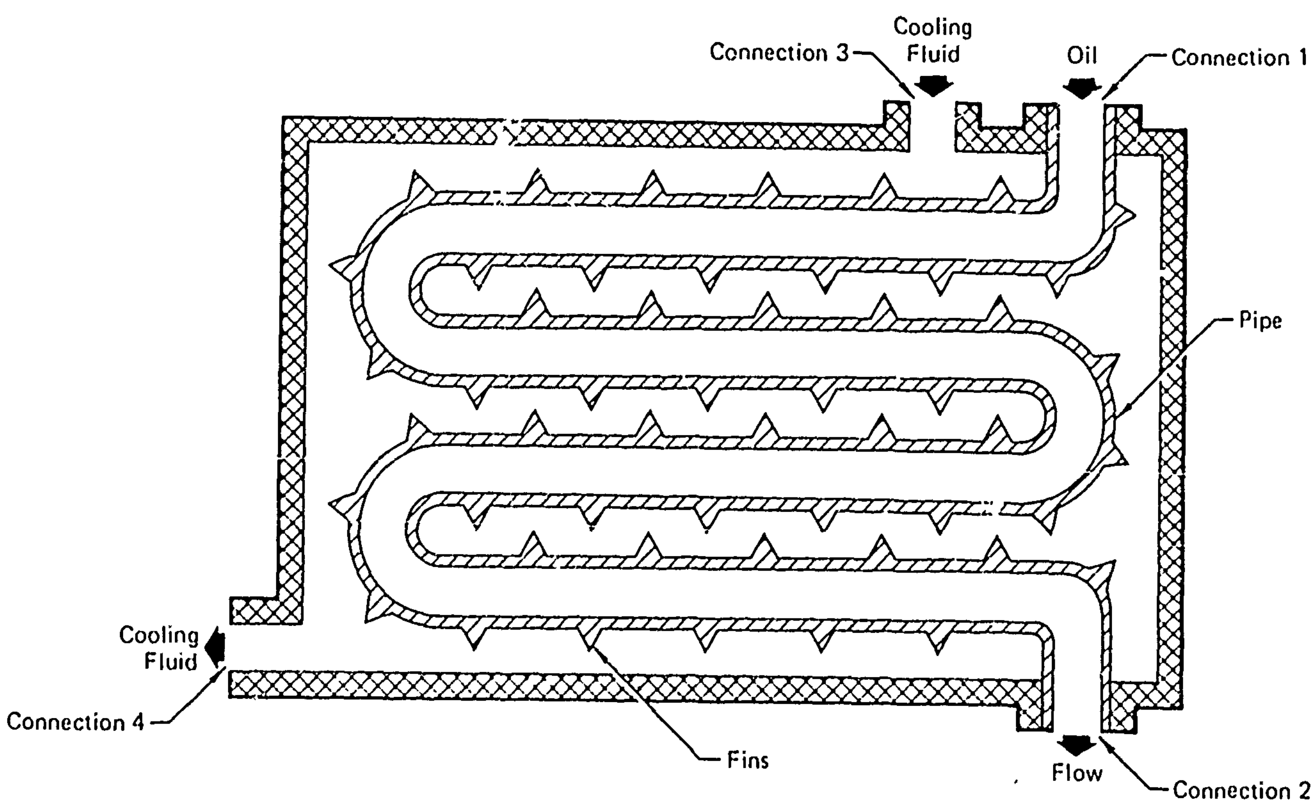


FIGURE 6.69-1

TYPE NO. 69 HEAT EXCHANGER

6.69.1 Math Model

The thermal math model for the heat exchanger includes heat transfer to and from four connecting line segments, two hydraulic segments and two cooling segments, as shown in Figure 6.62-2.

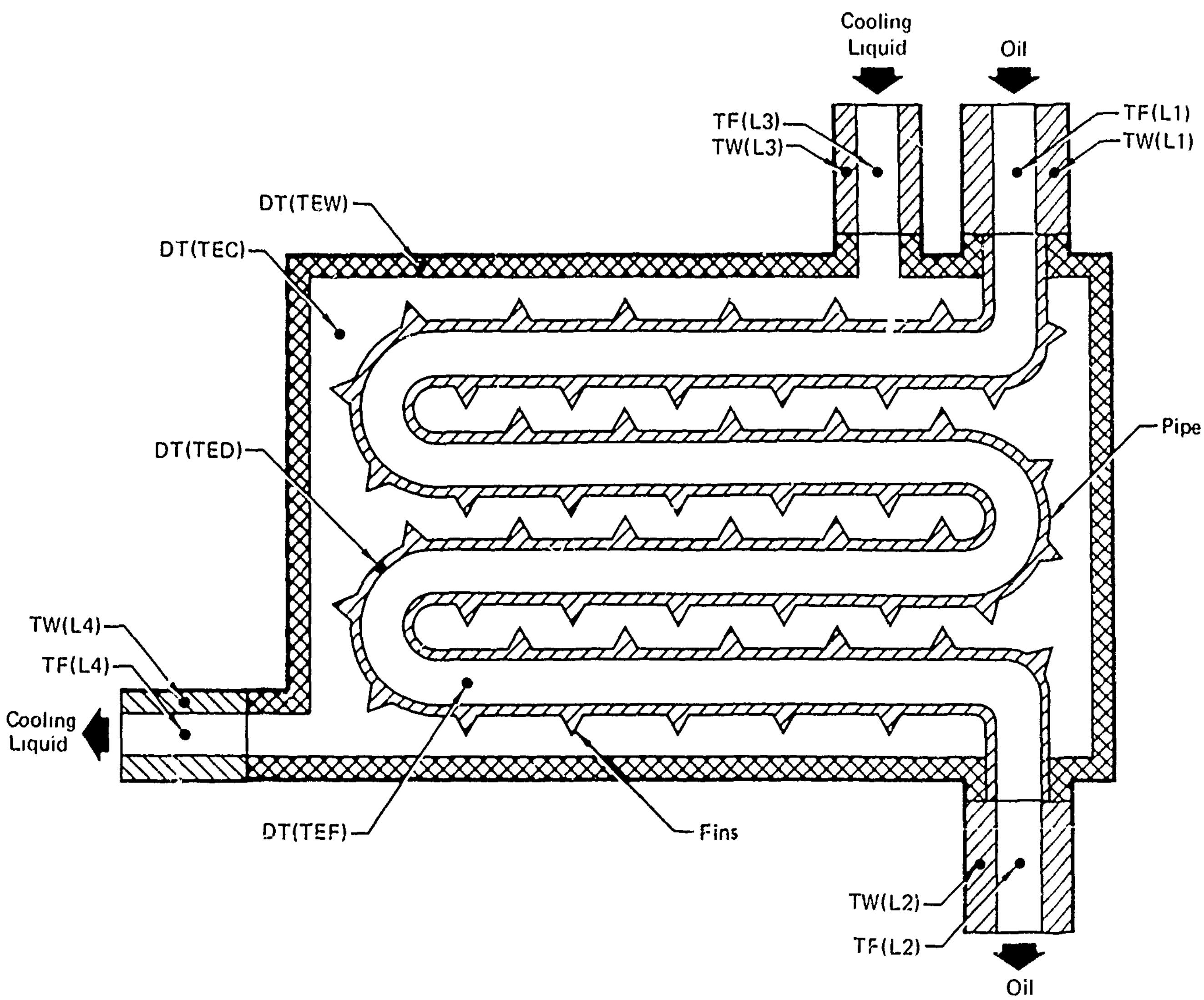


FIGURE 6.69-2
HEAT EXCHANGER AND LINE NODE REPRESENTATION

GP77-0065-21

The heat exchanger is represented by four nodes, one exterior wall node, one interior wall node (representing pipes or tubes etc), one hydraulic fluid node, and one cooling fluid node. Each connecting line segment (hydraulic and cooling) is represented by two nodes, one fluid and one wall. The temperature of the heat exchanger wall node is DT(TEW), interior wall node is DT(TEP), hydraulic fluid node is DT(TEF) and cooling fluid node is DT(TEC). The temperatures of the hydraulic connecting line segment wall and fluid nodes are TW(L1), TW(L2), TF(L1), and TF(L2). The temperatures of the cooling liquid connecting line segment wall and fluid nodes are TW(L3), TW(L4), TF(L3)=D(TEC1), and TF(L4) = TEMPCOT. (Note: TEMPCOT=DT(TEC), see assumptions).

Four equations are written to solve for DT(TEW), DT(TEP), DT(TEF), and DT(TEC), using the heat exchanger and line segment material properties and dimensions, the atmosphere and structure temperatures external to the heat exchanger and TW(L1), TW(L2), TF(L1), TW(L3), TW(L4), and D(TEC1). (Note: TF(L2) = DT(TEF), TW(L3)=D(TEC1) and TW(L4) =TEMPLOT, see assumptions.)

The first equation represents three modes of heat transfer with the heat exchanger hydraulic fluid node:

1. Convection to and from the interior wall (pipe)

   B1*(DT(TEP)-DT(TEF))

   where B1 is the convection coefficient and is equal to UFWIL*ASFP.

2. Heat transfer due to mass transfer into the heat exchanger from upstream of the heat exchanger.

   $\dot{M}$Cp*(TF(L1)-DT(TEF))

   where $\dot{M}$Cp is the mass flow rate and is equal to Q(L1)* RHOIL*CPFN

3. Heat transfer due to a pressure drop across the heat exchanger

$$\dot{M}Cp*DCAPT1$$

where $DCAPT = (1.0/RHOIL)*(P(L1)-P(L2))/(CJ*CPFN)$

These terms are combined to produce the equation for the heat balance for the heat exchanger fluid node.

$$\frac{\dot{M}Cp}{DELT}*(DT(TEF)-DT(TEF)_{OLD}) = B1*(DT(TEP)-DT(TEF) + \dot{M}Cp*(TF(L1)-DT(TEF)) + \dot{M}Cp\ DCAPT1 \tag{1}$$

where $\dot{M}Cp$ is equal to FMASS*CPFN

The second equation represents three modes of heat transfer relative to the heat exchanger exterior wall node:

1a. convection to and from the cooling fluid node

$$B7*(DT(TEC)-DT(TEW))$$

where B7 is a convection coefficient equal to D(UCW)*D(ASCW)

1b. convection to and from the atmospheric air

$$B5*(D(TA)-DT(TEW))$$

where B5 is the convection coefficient and is equal to D(UAW)* D(ASAW)

2a. conduction to and from the hydraulic fluid connecting lines

$$RI*(TW(LI)-DT(TEW))$$

where RI is the conduction coefficient and is equal to 1.0/(DXF(LI)/(ACW(LI)*C(LI)) + DXE/(ACEW*CEW)) and I=1 for line 1 and 2 for line 2

2b. conduction to and from the cooling fluid connecting lines

$$R3*(TEMPCIN-DT(TEW)) \text{ and}$$

$$R4*(TEMPCOT-DT(TEW))$$

where R3=R4 and are equal to 1.0/(2.0*DXE/ACEW*CEW)) and $TEMPCOT=DT(TEW)_{OLD}$

3.0 radiation exchange with surrounding structure

CIP*(D(TST)-(DT(TEW)+460.)**4)

where CIP is equal to SIGMA*EPSION*SHAPF*D(ASAW)

These terms combine to produce the equation for heat balance for the exterior wall node:

$$\frac{MCp}{DELT}*(DT(TEW)-DT(TEW)_{OLD}) = B7*(DT(TEC)-DT(TEW)) + R1*(TW(L1)-DT(TEW))+R2(TW(L2)-DT(TEW)) +R3*(TEMPCIN-DT(TEW))+R4(TEMPCOT-DT(TEW)) +B5(D(TA)-DT(TEW))+C1P*D(TST)- C1P*(DT(TEW)+460.)**4 \quad (2)$$

where MCp is equal to D(EMASS)*CPCN

The third equation represents three modes of heat transfer relative to the heat exchanger cooling liquid node:

1a. convection to and from the interior wall or pipe node

B4*(DT(TEP)-DT(TEC))

where B4 is the convection coefficient and is equal to D(UCP)*D(ASCP)

1b. convection to and from the exterior wall node

B7*(DT(TEW)-DT(TEC))

where B7 is equal to D(UCW)*D(ASCW)

2. heat transfer due to mass transfer into the cooling liquid node from upstream of the node

$\dot{M}Cp$(D(TEC1)-DT(TEC))

where $\dot{M}Cp$ is the flow coefficient and is equal to D(RMFCL)*CPCN

3. heat transfer due to pressure drop across the heat exchanger coding liquid node

$\dot{M}Cp$ * DCAPT2

where DCAPT2 is equal to (1.0/RHOIL)*D(PP3)/(CJ*CPCN)

These terms are combined to produce the equation for heat balance for the cooling fluid node:

$$\frac{MCp}{DELT}*(DT(TEC)-DT(TEC)_{OLD}) = B4*(DT(TEP)-DT(TEC)) \\ +B7*(DT(TEW)-DT(TEC)) \\ +\dot{M}Cp*(D(TEC1)-DT(TEC)) \\ +\dot{M}Cp*DCAPT2 \quad (3)$$

where MCp is equal to CMASS*CPCN

The fourth equation represents one mode of heat transfer relative to the interior wall (pipe) node:

1a. convection to and from the coding fluid node

B4*(DT(TEC)-DT(TEP))

where B4 was defined previously.

1b. convection to and from the hydraulic fluid node

B1*(DT(TEF)-DT(TEP))

where B1 was described previously

These terms combine to produce the equation for heat transfer to and from the interior wall node:

$$\frac{MCp}{DELT}(DT(TEP)-DT(TEP)_{OLD}) = B4*(DT(TEC)-DT(TEP)) \\ + B1*(DT(TEF)-DT(TEP)) \quad (4)$$

where MCp is equal to D(PMASS)*CPPN

Figure 6.69-3 is a thermal resistance for the heat exchanger and shows how all four nodes are interrelated. Equations (1) through (4) are solved simultaneously for the temperatures of each node.

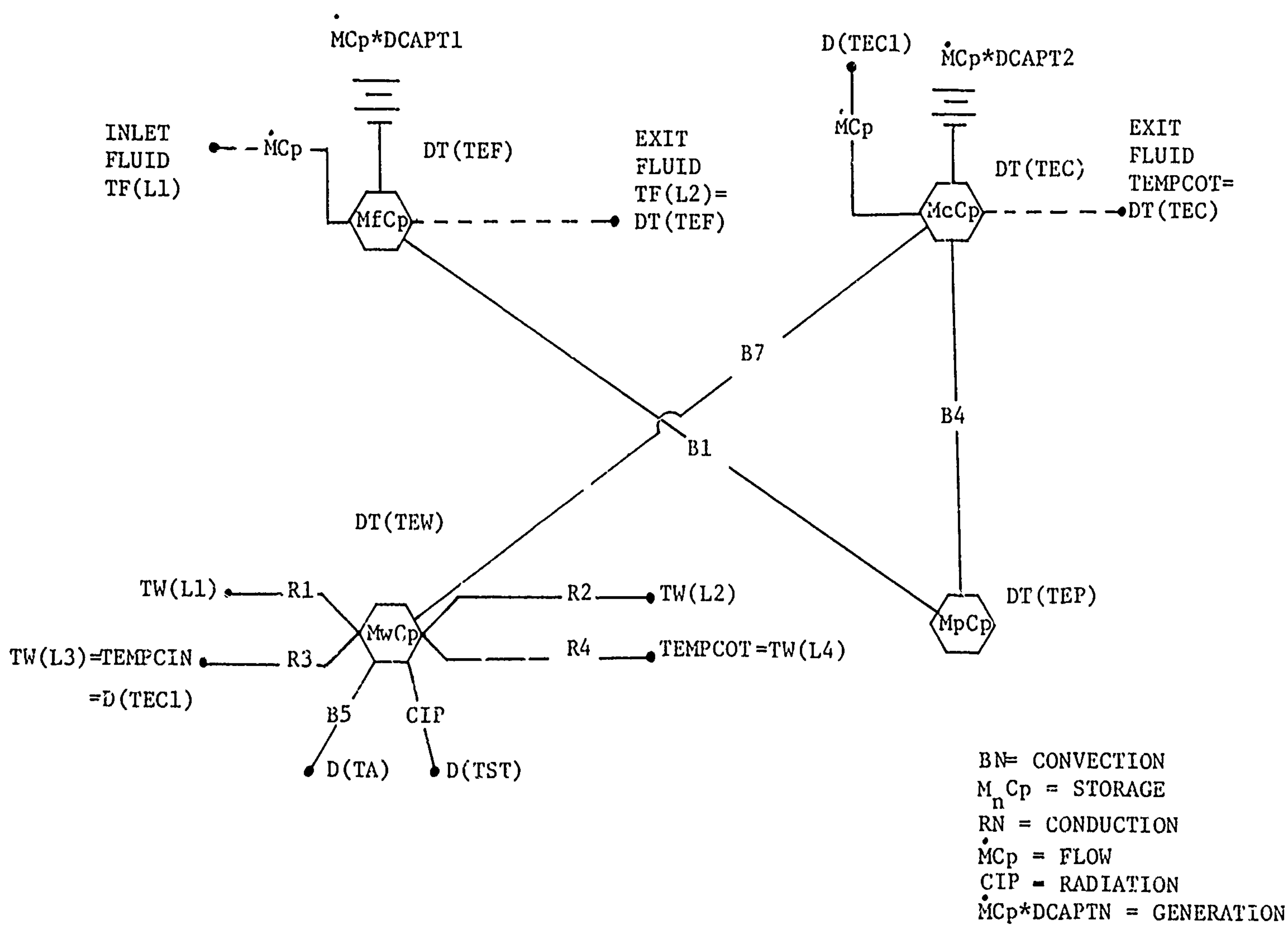


FIGURE 6.69-3

THERMAL MODEL

In the hydraulic math model the pressure drop through the heat exchanger is computed using equation (5).

```
PUP = PUP-Q1*D(LAM)*VISC(TF(L(ICON),PUP)                    (5)
    *RHO(TF(L(ICON)),PUP)/(.028*8.2E-5)
```

where

PUP = upstream pressure (PSI)

Q1 = flow (CIS)

D(LAM) = laminar flow coefficient

RHO( ) = fluid density ($LB\text{-}SEC^2/IN^4$)

VISC( ) = fluid viscosity (IN^2/SEC)

In equation (5) the laminar flow coefficient is corrected to the system temperature and fluid.

6.69.2 Assumptions

1. Atmosphere and structure temperatures remain constant
2. The entire wall or case of the exchanger is all at the same temperature.
3. The interface conductance between the exchanger wall and the line wall segment is infinite.
4. The emissivity of the walls remain constant at .3 for steel.
5. TW(L4), the downstream cooling liquid line segment temperature is at the temperature of the exiting fluid TEMPCOT which also equals DT(TEC).
6. No conductance between the pipe and the walls of the exchanger.
7. The temperature of the exiting hydraulic fluid is equal to the calculated temperature, TF(L2)=DT(TEF).
8. Complete mixing occurs in the fluid volume.
9. TW(L3), the upstream cooling liquid line segment temperature is equal

to D(TEC1) the cooling liquid inlet temperature.

6.69.3 Computation Methods

The subroutine executes the above discussed calculations as follows:

SECTION 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Fahrenheit to Rankine and raised to the fourth power, and the default values are assigned.

SECTION 2000

The pressure drop of the hydraulic fluid through the heat exchanger is computed using equation (5).

SECTION 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (the program is set up with the flow entering connection line one (L1) and leaving thru connection line two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 4x4 matrix is loaded and the mathematical equations are solved for DT(TEW), DT(TEP), DT(TEC) and DT(TEF). The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to arrays (TC and TF) in common /TRANS/.

6.69.4 Approximations

1. The exit cooling fluid line wall is at the temperature of the cooling fluid exiting.

6.69.5 Limitations - Not applicable.

6.69.6 Variable Listing

Variable	Description	Units
A( )	Computational array	$IN.^2$
ACEW	Cross sectional area of the exchanger walls	$IN.^2$
D(ASAW)	External surface area of the exchanger walls	$IN.^2$
D(ASCP)	Surface area cooling fluid to pipe (fins)	$IN.^2$
D(ASCW)	Surface area cooling fluid to exchanger exterior walls	$IN.^2$
ASFP	Internal surface area of the cooling pipe	$IN.^2$
B( )	Computational array	
B1,B2,B3,B4,B5,B7	Variable coefficients	
CEW	Thermal conductivity of the exchanger walls	WATTS/IN-°F
CJ	Mechanical equilivant of heat, 8.85	IN-LB_m/WATTS-SEC
CMASS	Cooling liquid mass	LB_m
CPCN	Specific heat of the cooling liquid	WATTS-SEC/LBm-°F
CPPN	Specific heat of the exchanger pipe	WATTS-SEC/LB_m-°F
CPWN	Specific heat of the exchanger walls	WATTS-SEC/LB_m-°F
D(CTYPE)	Cooling liquid type (use 1.)	
D(DELTA)	Overall length of the length	IN.
D(DELTAX)	Pipe length thru exchanger	IN.
DXE	Distance from node to interface, exchanger watts	IN.
D(EMASS)	Exchanger mass	LB_m
EPSION	Emissivity of the walls, constant .3	
FMASS	Fluid mass	LB_m
D(IDIA)	Inside diameter of the pipe	IN.
D(ITC)	Initial temperature of the walls	°F
D(ITF)	Initial temperature of the fluid	°F

Variable	Description	Units
D(ITL)	Initial temperature of the cooling liquid	°F
KTYPE	Dummy variable	-
D(LAM)	Laminar flow coefficient	PSI/CIS
D(MTYPE)	Exchanger wall material type	-
NTYPE	Dummy variable	-
D(PMASS)	Pipe mass (fins etc)	LB_m
D(PP3)	Pressure drop across exchanger for cooling liquid	PSI
RHOC	Density of the cooling fluid	$LB_m/IN.^3$
RHOE	Density of the exchanger walls	$LB_m/IN.^3$
RHOIL	Density of the hydraulic fluid	$LB_m/IN.^3$
RHOP	Density of the exchanger pipe (fins)	$LB_m/IN.^3$
D(RMFCL)	Mass flow rate of entering cooling liquid	$LB_m/SEC.$
R1,R2,R3,R4	Dummy variables coefficients	-
SHAPF	Shape factor of the walls, constant .96	-
SIGMA	Stefan-Boltzman radiation constant $.385 \times 10^{-11}$	$WATTS/IN^2\text{-}R^4$
D(TA)	Surrounding ambient temperature	°F
DT(TEC)	Temperature of the exchanger cooling liquid, to be calculated	°F
D(TEC1)	Inlet temperature of the cooling liquid	°F
DT(TEF)	Temperature of the exchanger fluid, to be calculated	°F
TEMP1	Temperature of the walls for heat transfer calculation	°F
TEMPCIN	Temperature of cooling liquid in	°F
TEMPCOT	Temperature of cooling liquid out	°F
DT(TEP)	Temperature of the exchanger pipe (fins, to be calculated)	°F
D(TST)	Surrounding structure temperature	°F

Variable	Description	Units
DT(TEW)	Temperature of the exchanger walls, to be calculated	°F
D(UAW)	Heat transfer coefficient ambient to walls	WATTS/IN^2-°F
D(UCP)	Heat transfer coefficient cooling liquid to pipe	WATTS/IN^2-°F
D(UCW)	Heat transfer coefficient cooling liquid to walls	WATTS/IN^2-°F
UFWIL	Heat transfer coefficient, fluid to pipe walls	WATTS/IN^2-°F
D(VOLC)	Volume of the cooling liquid in the exchanger	IN.^{3}
VOLF	Volume of the hydraulic fluid in exchanger	IN.^{3}

6.69.7 <u>Subroutine Listing</u>

```
      SUBROUTINE THLX69 (D,DT,DD,L)
C *** REVISED OCTOBER 28,1976 ***
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,IHLL
      COMMON /STEADY/PN(90),QF(90),PLX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,NNODE,NLEG,NCPN,TERM,
     +  LEGN,ICON,INV,INX,INZ,NUP(99),NDWN(90),NLLEM(90),
     +ILEGND(90),ILEG(1000)
      COMMON /FLUID/APPRES,CP,CPFN,FTEMP,PROP(13,3)
      DIMENSION A(4,4),B(4)
      INTEGER MTYPE,CTYPE,DELTAX,IDIA,ASCW,ASAW,UAW,MDFCL,
     + TEC1,VOLC,EMASS,UCW,DELTA,TST,TA,TEC,TEW,TEF,
     + TEP,UCP,PP3,PMASS,ASCP
C     D ARRAY VARIABLES
      DATA MTYPE/1/,CTYPE/2/,EMASS/3/,DELTAX/5/,IDIA/6/,DELTA/7/,PMASS
     1/4/,VOLC/8/,ASCW/9/,ASAW/10/,UAW/12/,UCW/13/,MDFCL/15/,TEC1/16/,
     2 TST/17/,TA/18/,ITF/19/,ITC/20/,ITL/21/,LAM/22/,UCP/14/,
     3 PP3/24/,ASCP/11/
      DATA SIGMA/.349E-11/,SHAPF/.96/,EPSION/.30/,CJ/3.85/
C     DT ARRAY VARIABLES
      DATA TEF/1/,TEW/2/,TEC/3/,TEP/4/
      IF(IENTR) 1000,2000,3000
 1000 CONTINUE
      L1=L(1)
      L2=L(2)
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
      TC(L1)=D(ITC)
      TC(L2)=D(ITC)
      DT(TEF)=D(ITF)
      DT(TEW)=D(ITC)
      DT(TEC)=D(ITL)
      DT(TEP)=(D(ITF)+D(ITL))/2.0
      D(TS4)=(D(TST)+460)**4
      IF(D(UAW).LE.0.0) D(UAW)=0.0069
      D(WMASS)=D(EMASS)-D(PMASS)
C     MTYPE    =MATERIAL TYPE
C     D(DELTA) =EXCHANGER LENGTH
C     D(DELTAX)=EXCHANGER TUBE LENGTH
C     D(ASAW)  =SURFACE AREA EXCHANGER TO AMBIENT
C     D(ITC)   =INITIAL TEMPERATURE OF COMPONENT
C     D(UCW)   =HEAT TRANSFER COEFFICIENT WALL TO COOLING LIQUID
C     D(TA)    =TEMPERATURE OF AMBIENT
C     D(TST)   =TEMPERATURE OF STRUCTURE, DEG. F
C     D(VOLC)  =VOLUME OF COOLING LIQUID INSIDE EXCHANGER
C     D(MDFCL) =MASS FLOW RATE INTO EXCHANGER, COOLING LIQUID
C     D(EMASS)  =MASS OF EXCHANGER TOTAL,(WALLS,PIPES,FINS)
C     D(PP3)   =PRESSURE DROP ACROSS EXCHANGER, COOLING LIQUID
```

6.69.7 (Continued)

```
C       D(PMASS)  =PIPE MSAA INCLUDING ALL FINS,OR JUST
C         FINS IF NO PIPE
C       D(UCP)    =HEAT TRANSFER COEFF. PIPE TO COOLING LIQUID
C       NODE C IS COOLING LIQUID (JP-4)
C       NODE P IS THE PIPE INSIDE THE EXCHANGER
        RETURN
C       STEADY STATE SECTION
 2000   CONTINUE
        PUP=PUP-D1*D(LA.)*VISC(TF(L(ICON)),PUP)*
      + RHO(TF(L(ICON)),PUP)/(.028*3.2E-5)
        RETURN
 3000   CONTINUE
        KTYPE=D(KTYPE)+.001
        CTYPE=D(CTYPE)+.001
        CLA=PROP(KTYPE,3)
        RHOE=PROP(KTYPE,2)
        RHOP=PROP(KTYPE,2)
        CPPN=PROP(KTYPE,1)
        CPWN=PROP(KTYPE,1)
        CPCN=527.4
        RHOC=.025
        L1=L(1)
        L2=L(2)
        TEMPCIN=D(TEC1)
        TEMPCOUT=DT(TEC)
 3100   RHOIL=336.4*RHO(DT(TEF),P(L1))
        CMASS=D(VOLC)*RHOC
C       CMASS IS THE COOLING LIQUID MASS, JP-4 NOW, AND
C        IF A DIFFERENT LIQUID IS NEEDED CHANGES MUST BE MADE
C       TO CPCN AND RHOC TO FIT SPECIFIC NEED
        VOLF=PI*(D(IDIA))**2*D(DELTAX)/4.
        FMASS=VOLF*RHOIL
        AAA=PI*D(IDIA)**2)/4.
        DDD=D(IDIA)
        TEMP1=DT(TEM)
        ASFP=PI*D(IDIA)*D(DELTAX)
        UFOIL=UFW(AAA,DDD,ABS(D(L1)),TF(L1),P(L1))
C
        DXE=D(DELTA)/2.0
 3100   ACEW=D(WMASS)/(RHOE*D(DELTA))
        A1=DXE/(ACEW*CLA)
        R1=1./(DXF(L1)/(ACW(L1)*C(L1))+A1)
        R2=1./(DXF(L2)/(ACW(L2)*C(L2))+A1)
        R3=1./(2.0*A1)
        R4=1./(2.0*A1)
        B1=UFOIL*ASFP
        B2=ABS(D(L1))*RHOIL*CPFN
        B3=D(RMFCL)*CPCN
        B4=D(UCP)*D(ASCP)
        B5=D(UAW)*D(ASAW)
```

6.69.7 (Continued)

```
      B7=D(UCW)*D(ASCW)
      CIP=SIGMA*SHAPF*EPSION*D(ASAW)
      DCAPT=(1./RHOIL)*ABS(P(L1)-P(L2))/(CJ*CPFN)
      DCAPT2=(1./RHOC)*ABS(D(PP3))/(CJ*CPCN)
C     FLUID, WALL, COOLING LIQUID, PIPE, ARE THE NODES IN ORDER
 3300 A(1,1)=FMASS*CPFN/DELT+B1+B2
      A(1,2)=0.0
      A(1,3)=0.0
      A(2,1)=0.0
      A(2,2)=D(WMASS)*CPWN/DELT+B7+B5+R1+R2+R3+R4
      A(2,3)=-B7
      A(3,1)=0.0
      A(3,2)=-B7
      A(3,3)=CMASS*CPCN/DELT+B7+B4+B3
      A(4,1)=-B1
      A(1,4)=A(4,1)
      A(4,2)=0.0
      A(2,4)=0.0
      A(4,3)=-B4
      A(3,4)=-B4
      A(4,4)=D(PMASS)*CPPN/DELT+B4+B1
      B(4)=D(PMASS)*CPPN*DT(TEP)/DELT
      B(1)=FMASS*CPFN*DT(TEF)/DELT+B2*TF(L1)+DCAPT*B2
      B(2)=D(WMASS)*CPWN*DT(TEW)/DELT+R1*TW(L1)+R2*TW(L2)
     + +R3*TEMPCT+R4*TEMPCOT+B5*D(TA)+CIP*D(TET)-CIP*(DT(TEW)+
     + 460.)**4
      B(3)=CMASS*CPCN*DT(TEC)/DELT+B3*D(TEC1)+DCAPT2*B3
 4900 CALL SIMULT(A,3,4,IERROR)
      TF(L2)=B(1)
      TC(L1)=B(2)
      TC(L2)=B(2)
      DT(TEF)=B(1)
      DT(TEP)=B(4)
      DT(TEW)=B(2)
      DT(TEC)=B(3)
      RETURN
      END
```

6.71 SUBROUTINE TACUM71

Subroutine TACUM71 simulates a simple gas charged piston type accumulator that can be used as a system accumulator, as sketched in Figure 6.71-1. When used as a system accumulator, the initial volume of oil in the accumulator is determined by the steady state pressure. Two connections are provided, both of which are assumed to be at the same pressure. When a single connection is used, the other is blanked off automatically.

Since it is basically a passive device, its response is entirely dependent on line flow, pressure and temperature changes.

The subroutine calculates the accumulator fluid (oil) temperature, the gas temperature, and the temperature of the accumulator walls and piston.

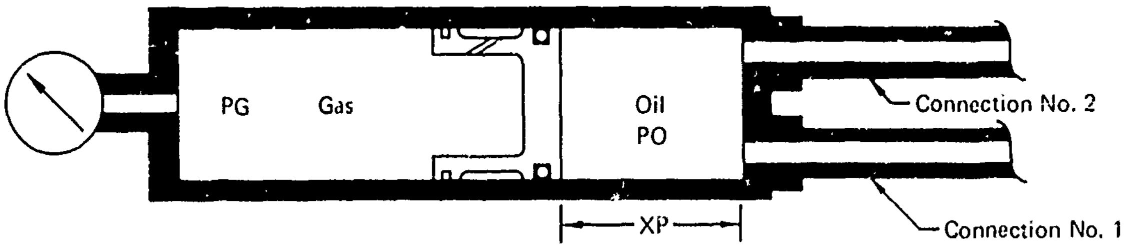


FIGURE 6.71-1
TYPE NO. 71 FREE PISTON ACCUMULATOR

GP74 0773 1

6.71.1 Math Model

The THERMAL math model for the accumulator includes heat transfer to and from two connecting line segments, one upstream and one downstream. Nine nodes are considered: three fluid nodes, one gas node, and five wall nodes (as shown in Figure 6.71-2). The temperatures of the upstream connecting line segment wall and fluid nodes are denoted by TW(L1) and TF(L1). The temperatures

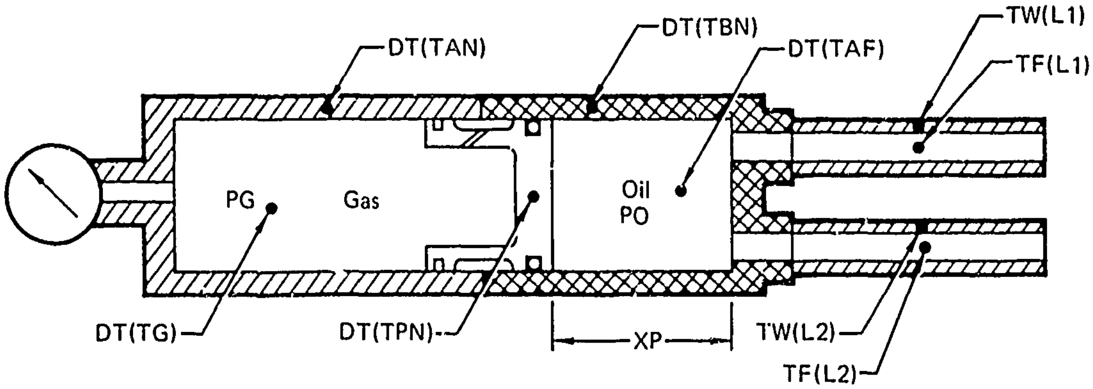


FIGURE 6.71-2
ACCUMULATOR AND CONNECTING LINE SEGMENT REPRESENTATION
GP77-0065-11

of the accumulator wall and fluid nodes are denoted by DT(TAN), DT(TBN), DT(TPN), and DT(TAF), and the temperature of the gas node is DT(TG). The temperature of the downstream connecting line segment wall and fluid nodes are TW(L2) and TF(L2). Five heat balance equations are written to solve for the five accumulator nodes temperature DT(TAN), DT(TBN), DT(TPN), DT(TG) and DT(TAF), using the accumulator and line segment material properties and dimensions, the atmosphere and structure temperature external to the accumulator, and TW(L1), TW(L2), and TF(L1). (Note: TF(L2) = DT(TAF), see assumptions). One equation for the heat balance for each of the accumulator nodes is produced. The first equation represents two modes of heat transfer relative to the accumulator gas node.

1. Heat transfer due to expansion or compression of the gas by the piston

$$DT(TG) = \frac{DT(TG)_{OLD} *DT(PG)*DT(VOLG)}{DT(OPG)*DT(OVOLG)}$$

This term calculates the new temperature of the gas just due to expansion or compression, not with its reactions with the other nodes.

2a. Convection to and from the accumulator wall node

$$B1*(DT(TAN)-DT(TG))$$

where B1 is the convection coefficient and is equal to D(UGA)*ASGA

2b. Convection to and from the accumulator piston node

$$B2*(DT(TPN)-DT(TG))$$

where B2 is the convection coefficient and is equal to D(UGA)*D(AREA)

These heat transfer terms are combined to produce the equation for the heat balance for the accumulator gas

$$DT(TG) = DT(TG)*DT(PG)*DT(VOLG)/(DT(OVOLG)*DT(OPG))$$

this new DT(TG) then becomes the $DT(TG)_{OLD}$ in the next equation

$$\frac{MCp}{DELT}(DT(TG)-DT(TG)_{OLD}) = B1*(DT(TAN)-DT(TG)) + B2*(DT(TPN)-DT(TG)) \tag{1}$$

The second equation represents three modes of heat transfer relative to the accumulator fluid (oil):

1a. Conduction to and from the upstream line segment fluid

$$R1*(TF(L1)-DT(TAF))$$

where R1 is equal to CF/(DXF(L1)/ACF(L1)+DXAF/ACAF + RMFL1*DELT/(ACAP**2*RHOIL)), the conduction coefficient for the fluid. RMFL1 equals Q(L1)*RHOIL.

1b. Conduction to and from the downstream line segment fluid node if the flow rate is negligible, or not leaving the accumulator.

$$R2*(TF(L2)-DT(TAF))$$

where R2 equals R1 except instead of L1, R2 uses L2. (Note: There may only be one connecting line with the second being closed off and consequently this term would be zero.)

2a. Convection to and from the accumulator wall node around the fluid.

$$B5*(DT(TBN)-DT(TAF))$$

where B5 is the convection coefficient and is equal to D(UFWA)*ASFB.

2b. Convection to and from the piston node

$$B4*(DT(TPN)-DT(TAF))$$

where B4 is the convection coefficient and is equal to D(UFWA)*D(AREA)

3. Heat transfer due to mass transfer into the accumulator from the upstream connecting line node

$$\dot{M}Cp*(TF(L1)-DT(TAF))$$

where $\dot{M}Cp$ is the flow rate coefficient and is equal to Q(L1)*RHOIL. If there is no fluid entering the accumulator the last term is set equal to zero.

These heat transfer terms are combined to produce the equation for the heat balance for the accumulator fluid node:

$$\frac{MCp}{DELT}*(DT(TAF)-DT(TAF)_{OLD}) = R1*(TF(L1)-DT(TAF))+R2*(TF(L2)-DT(TAF)) + \dot{M}Cp*(TR(L1)-DT(TAF))+B5(DT(TBN-DT(TAF))+B4*(DT(TPN)-DT(TAF))$$

where MCp is equal to FMASS*CPFN

The third equation represents three modes of heat transfer relative to the accumulator wall node surrounding the gas node.

1a. Conduction to and from the accumulator wall surrounding the fluid.

R4*(DT(TBN)-DT(TAN))

where R4 is the conduction coefficient and is equal to CA/(DXAA/ACAA+DXAB/ACAB)

1b. Conduction to and from the accumulator piston node

R3*(DT(TPN)-DT(TAN))

where R3 is the conduction coefficient for the wall and is equal to 1.0/(DXAP/(ACAP*CP)+DXAA/(ACAA*CA)+1.0/(CAP*ACAP))

2a. Convection to and from the gas node

B1*(DT(TG)-DT(TAN))

B1 is a convection coefficient and was defined previously.

2b. Convection to and from the external atmosphere

D1*B3*(D(TA)-DT(TAN))

where B3 is the convection coefficient equal to D(UAA)*D(ASAA) and D1 is equal to DT(VOLG)/(DT(VOLG)+DT(VOLO)), a term to represent the accumulator mass surrounding the gas.

3. Radiation exchange with the surrounding structure

$$D1*CIP*D(ASAA)*(D(TST)-(DT(TAN)+460.)^{4})$$

where CIP is equal to SIGMA*EPSION*SHAPF, the radiation coefficient, and D1 is as above.

These terms are combined to produce the equation for the heat balance for the accumulator wall node.

$$\frac{MCp}{DELT}\,(DT(TAN)-DT(TAN)_{OLD}) = R4(DT(TBN)-DT(TAN))+R3(DT(TPN)-DT(TAN))+B1*(DT(TG)-DT(TAN))+B3*(D(TA)-DT(TAN))+CIP*D(ASAA)*(D(TST)-(DT(TAN)+460)**4) \qquad (3)$$

where MCp is equal to D(AMASS)*D1*CPAN

The fourth equation represents three modes of heat transfer relative to the accumulator wall node surrounding the oil volume.

1a. Conduction with the connecting line segment wall,

R6*(TW(L1)-DT(TBN))

where R6 is the conduction coefficient for the walls equal to

1.0/(DXF(L1)/(ACW(L1)*C(L1))+D3).

D3 is equal to DXAB/(ACAB*CA).

This may be upstream if fluid is entering, or downstream if fluid is only leaving, or either if there are two connecting lines.

1b. Conduction to and from that second connecting line wall node.

R7*(TW(L2)-DT(TAN))

where R7 is a conduction coefficient equal to 1.0/(DXF(L2)/ACW(L2)*C(L2))+D3)

1c. Conduction to and from the accumulator wall node surrounding the gas.

R4*(DT(TAN)-DT(TBN))

where R4 has been defined previously.

1d. Conduction with the piston node

R5*(DT(TPN)-DT(TBN))

where R5 is the conduction coefficient between the two nodes equal to 1.0/(DXAP/(ACAP*CP)+B3+1.0/(CAP*ACAP))

2a. Convection to and from the external atmosphere

D2*B3*(D(TA)-DT(TBN))

B3 being same as defined previously and D2 is equal to DT(VOLO)/(DT(VOLO)+DT(VOLG)), a term for the amount of accumulator mass surrounding the fluid.

2b. Convection with the fluid node in the accumulator

B5(DT(TAF)-DT(TBN))

where B5 is the same as previously defined.

3. Radiation exchange with the surrounding structure

$$D2*(CP*D(ASAA)*(D(TST)-(DT(TBN)+460)^{4})$$

where all terms have been previously defined.

These six terms are combined to produce the equation for the heat balance for the accumulator wall node surrounding the fluid.

$$\frac{MCp}{DELT}\ (DT(TBA)-DT(TBN)_{OLD}) = \begin{array}{l} R6*(TW(L1)-DT(TBN))+R7(TW(L2)-DT(TBN)) \\ +R4*(DT(TAN)-DT(TBN)-DT(TBN))+R5(DT(TPN) \\ -DT(TBN))+D2*B3*(D(TA)-(DT(TBN)) \\ +B5(DT(TAF)-DT(TBN))+D2*CIP* \\ D(ASAA)*(D(TST)-(DT(TBN)+460)**4) \end{array} \qquad (4)$$

where MCp is equal to D(AMASS)*D2*CPAN

The fifth equation represents two modes of heat transfer relative to the accumulator piston node.

1a. Conduction to and from the accumulator wall node gas side

R3*(DT(TAN)-DT(TPN))

with R3 being defined previously

1b. Conduction to and from the accumulator wall node fluid side

R5(DT(TBN)-DT(TPN))

with R5 being defined previously.

2a. Convection to and from the gas node

B2*(DT(TG)-DT(TPN))

and again B2 is the same as was defined previously.

2b. Convection to and from the fluid node

B4*(DT(TAF)-DT(TPN))

with B4 being defined previously.

These terms combine to produce the equation for the heat balance for the accumulator piston node.

$$\frac{MCp}{DELT}(DT(TPN)-DT(TPN)_{OLD}) = R3*(DT(TAN)-DT(TPN))+R5*(DT(TBN)-DT(TPN)) +B2*(DT(TG)-DT(TPN))+B4*(DT(TAF)-DT(TPN)) \quad (5)$$

where MCp is equal to D(PMASS)*CPPN

A thermal model of the above heat transfer equations is shown in Figure 6.71-3. Equations (1) thru (5) are solved for the appropriate temperatures.

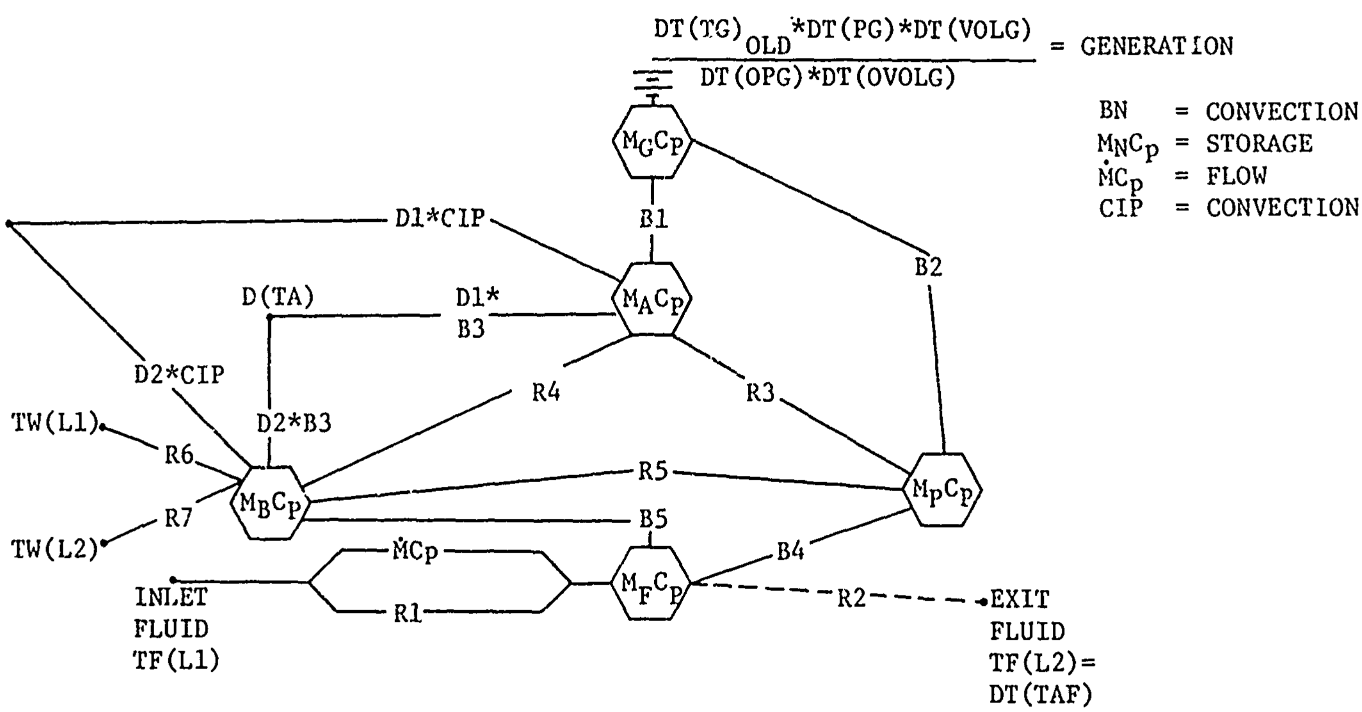


FIGURE 6.71-3

THERMAL MODEL

For the appropriate temperatures a thermal model for the accumulator is shown in Figure 6.71-3.

For entry and exit flow losses, a pressure loss term is input into the program. The term is corrected for the fluid type and operating temperature. The resulting pressure loss is

$$PUP=PUP-Q1*CORR \tag{6}$$

where Q1 = flow rate (CIS)

CORR = Entry exit flow constant (PSI/CIS)

PUP = Inlet Pressure (PSI)

CORR is the adjusted laminar flow constant determined by the following formula

$$CORR = D(LOSS)*(VISC_{OPERATING})(DENS_{OPERATING})/(VISC_{100})(DENS_{100})$$

6.71.2 Assumptions

1. The temperature of the atmosphere and structure surrounding the accumulator remain constant.
2. The temperature rise in the gas is due to compression or expansion of the gas from the fluid in the accumulator, besides the heat transferred to and from it.
3. The emissivity of the walls remain constant.
4. Complete mixing occurs in the fluid volume.
5. The interface conductance between the accumulator walls and the line walls is infinite.
6. The temperature of the fluid leaving the accumulator is equal to the fluid calculated temperature, DT(TAF).

6.71.3 Computational Methods

Section 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the forth power, and the default values are assigned.

Section 2000

The pressure loss due to fluid entry and exit is computed using equation (6) and the accumulator oil pressure is stored in DT(PO). At time zero in the program the initial gas volume is computed as

DT(VOLG) = DT(MAVOLG)-(DT(APRECH)*DT(MAVOLG))/DT(PG)

where

DT(MAVOLG) = MAX VOLUME OF GAS (IN^3)

DT(APRECH) = PRECHARGE PRESSURE (PSI)

DT(PG) = GAS PRESSURE (PSI)

DT(VOLG) = GAS VOLUME (IN^3)

Section 3000

The present oil volume is computed using a simple integration

TVOLO = DT(VOLO)+(DT(IQV)+TQV)*DT(NDELT))

where

DT(VOLO) = OLD OIL VOLUME (IN^3)

DT(TQV) = SUM OF FLOW IN AND OUT OF ACCUMULATOR (CIS)

DT(IQV) = OLD SUM OF FLOWS (CIS)

DT(NDELT) = .5*TIME STEP

TVOLO = TOTAL OIL VOLUME

TVOLO is checked to determine if the accumulator is full, empty or in its working range. With an empty accumulator the oil volume, TVOLO, is set to the minimum oil volume and the volume of gas becomes DT(MAVOLG). Similarly for a full accumulator the proper volumes are initialized.

After the volumes of oil and gas have been determined the gas pressure is computed

DT(PG)=DT(PG)+(DT(DPG)+TDPG)*DT(NDELT)

where

DT(PG) = OLD GAS PRESSURE (PSI)

TDPG = RATE OF CHANGE OF GAS PRESSURE WITH TIME (PSI/SEC)

DT(DPG) = OLD DIFFERENTIAL GAS PRESSURE (PSI/SEC)

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving through connection lines two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow direction is changed. A 5x5 matrix is loaded and the mathematical equations are solved for DT(TAF), DT(TG), DT(TAN), DT(TBN), and DT(TPN) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to special arrays TF and TC in COMMON/TRANS/.

6.71.4 Approximations

1. The shape factor is .96 (described in the technical discussion earlier).
2. The properties for the gas are for nitrogen at 100°F, and remain constant.

6.71.5 Limitations

The accumulator model may not be used as a pressure source in a hydraulic system.

6.71.6 Variable Listing

VARIABLE	DEFINITION	DIMENSION
A	Computational Array	--
AAA	Area Associated with the Inlet Flow to Accumulator	IN^2
AAMASS	Accumulator Mass - Surrounding the Gas	LB_m
ACAA	Cross Sectional Area of the accumulator wall	$IN.^2$
ACAB	Cross sectional area of the accumulator wall	$IN.^2$
ACAF	Cross sectional area of the accumulator fluid	$IN.^2$
ACAP	Cross sectional area of the accumulator piston	$IN.^2$
D(AMASS)	Accumulator Mass	LB_m
D(APRECH)	Precharge pressure adjusted to fluid temperature	PSI
D(AREA)	Piston to oil area	$IN.^2$
D(ASAA)	Surface area accumulator walls to atmosphere	$IN.^2$
ASFB,ASGA	Dummy variables	--
B( )	Computational Array	--
BMASS	Accumulator Mass Surrounding the fluid	LB_m
CA	Conducitivity of the accumulator walls	WATTS/IN.-°F
D(CAP)	Interface conductance - walls to piston	WATTS/IN.-°F
CIP	Dummy variable	--
CJ	Mechanical equivalent of heat	IN-LB_m/WATTS-SEC
CORR	Laminar flow coefficient correction term	--
CP	Thermal conductivity of the piston	WATTS/IN.-°F
CPAN	Specific heat of the accumulator walls	WATTS-SEC/LB_m-°F

6.71.6 Variable Listing (Continued)

VARIABLE	DEFINITION	DIMENSION
CPFN	Specific heat of the fluid	WATTS-SEC/LB_m-°F
CPGN	Specific heat of the gas	WATTS-SEC/LB_m-°F
CPPN	Specific heat of the piston	WATTS-SEC/LB_m-°F
DOD	Distance over which AAA Acts	IN.
D(DIA)	Inside Diameter of accumulator	IN.
D(DPG)	Differential gas pressure	PSI/SEC
DXAA	Distance from node to interface, wall to wall	IN.
DXAB	Distance from node to interface, wall to wall	IN.
DXAF	Distance from node to interface, fluid to connecting line	IN.
D XAP	Distance from node to interface, piston to wall	IN.
D1,D2,D3	Dummy Variables	--
EPSION	Emissivity factor of the walls	--
FMASS	Accumulator fluid mass	LB_m
D(CTYPE)	Gas material type	--
GMASS	Gas mass	LB_m
IERROR	Dummy variable	--
D(ITC)	Initial wall temperature	°F
D(ITF)	Initial fluid temperature	°F
D(ITG)	Initial gas temperature	°F
D(LOSS)	Entrance and exit loss coefficient	PSI/CIS
KTYPE	Dummy variable	--
DT(MAVOLG)	Maximum volume of gas	$IN.^3$
D(MAVOLO)	Maximum oil volume	$IN.^3$

6.71.6 Variable Listing (Continued)

VARIABLE	DEFINITION	DIMENSION
D(MIVOLG)	Minimum gas volume	$IN.^3$
D(MIVOLO)	Minimum oil volume	$IN.^3$
D(MTYPE)	Accumulator Wall Material Type	--
DT(NDELT)	.5 * Time Step	SEC
NTYPE	Dummy Variable	--
DT(OPG)	Old gas pressure	PSI
DT(OVOLG)	Old gas volume	$IN.^3$
DT(PG)	Gas pressure	PSI
D(PMASS)	Piston mass	LB_m
DT(PO)	Oil pressure	PSI
D(PPRES)	Gas precharge pressure	PSI
D(PTYPE)	Piston material type	--
RHOIL	Density of the oil	$LB_m/IN.^3$
RHOG	Density of the gas	$LB_m/IN.^3$
RHOP	Density of the piston	$LB_m/IN.^3$
SHAPF	Shape factor of the walls for radiation	--
SIGMA	Stefan-Boltzmann constant for radiation	$WATTS/IN.^2-°R^4$
D(TA)	Temperature of surrounding atmosphere	°F
DT(TAF)	Fluid temperature in accumulator	°F
DT(TAN)	Temperature of the wall surrounding the gas	°F
DT(TBN)	Temperature of the wall surrounding the fluid	°F
TDPG,TFO	Dummy variables	--
DT(TG)	Gas temperature	°F

6.71.6 Variable Listing (Continued)

VARIABLE	DEFINITION	DIMENSION
DT(TPN)	Piston temperature	°F
D(TST)	Temperature of the surrounding structure	°F
D(UAA)	Heat transfer coefficient, accumulator to ambient	WATTS/IN.^{2}-°F
D(UFWA)	Heat transfer coefficient, accumulator to fluid	WATTS/IN.^{2}-°F
D(UGA)	Heat transfer coefficient, accumulator to gas	WATTS/IN.^{2}-°F
DT(VOLG)	Volume of the gas	IN.^{3}
DT(VOLO)	Volume of the oil	IN.^{3}
D(WTHICK)	Accumulator wall thickness (average)	IN.

6.71.7 Subroutine Listing

```
      SUBROUTINE TACU71 (D,DT,DD,L)
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),D(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NLL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,INTPR,IFILL
      COMMON /GEOARY/XI(90),XJ(90),PLX(90),PDLEG(90),EL(90),
     1 DN,E,DI,PUP,PEOW,MODL,NLEG,NCPN,TLRN,LEGN,ICON,INV,
     2 INX,IEG,NUP(90),NOWN(90),NELEM(90),ILLCAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION A(5,5),B(5)
      INTEGER WTHICK,DIA,PMASS,HAA,ASAA,HGA,CAP,AMASS,DPG
     + ,TSC,TA,TAW,TBW,TPS,TG,TAF,VOLO,VOLG,PO,PG,APEA,PPRES
     + ,HFGA,CTYPE,PTYPE,APRLCH,OVOLG,VOLUML,DPC
C     D ARRAY VARIABLES
      DATA CTYPE/1/,GTYPE/2/,PTYPE/3/,AMASS/4/,PMASS/5/,WTHICK/6/,
     1 HFGA/7/,DIA/8/,ASAA/9/,HGA/10/,HAA/11/,CAP/12/,
     2 ASA/13/,TA/14/,ITC/15/,ITF/16/,ITG/17/,APEA/18/,AIVOLO/19/
     3 ,AVOLO/20/,AIVOLG/21/,PPRES/22/,LOSS/23/
C     DT ARRAY VARIABLES
      DATA TAW/1/,TBW/2/,TPS/3/,TG/4/,TAF/5/,VOLO/6/,VOLG/7/,
     + PO/8/,PG/9/,AVOLG/10/,OVOLG/11/,DPG/12/,APLNCH/13/
     + ,VOLUML/14/,INV/15/,DPC/16/,VOLDT/17/
      DATA SIGMA/.3196-11/,SHAPF/.96/,EPSION/1.0/,CJ/9.85/
C     CTYPE =MATERIAL TYPE OF THE CASE(CA,CPAN,RHOA)
C     PTYPE =MATERIAL TYPE OF THE PISTON(CP,CPPN,RHOP)
C     PMASS =MASS OF THE PISTON(#)
C     AMASS =MASS OF THE ACCUMULATOR(#,INCLUDING PISTON)
C     WTHICK=THICKNESS OF THE CASE WALL
C     DIA   =DIAMETER OF THE INSIDE OF THE ACCUMULATOR
C     HGA   =HEAT TRANSFER COEFF. GAS TO CASE
C     HAA   =HEAT TRANSFER COEFF. CASE TO AMBIENT
C     CAP   =INTERFACE CONDUCTANCE PISTON TO WALL
C     DPG   =PRESSURE OF THE GAS INITIAL OR PREVIOUS
C     ASAA  =SURFACE AREA CASE TO AMBIENT
C     HFGA   =HEAT TRANSFER COEFF. OIL TO WALL & PISTON
C     ASA   =PISTON TO OIL AREA
C     AIVOLO  =INIT. OIL VOLUME
C     AVOLO =MAX. OIL VOLUME
C     AIVOLG =INIT. GAS VOLUME
C     PPRES  =PRECHARGE PRESSURE
      IF(INTPR) 1000,2000,3000
 1000 CONTINUE
      L1=L(1)
C     INITIALIZING TEMPERATURES
      DT(TAW)=D(ITC)
      DT(TBW)=D(ITC)
      DT(TPS)=D(ITF)
      DT(TG)=D(ITG)
      DT(TAF)=D(ITF)
      TF(L1)=D(ITF)
```

6.71.7 (Continued)

```
      TC(L1)=D(ITC)
      D(AMASS)=D(AMASS)-D(PMASS)
      CIP=SIGMA*SHAPE*EPSION
      D(TST)=(D(TST)+460.)**4
      IF(D(UAA).LE.0.0)D(UAA)=.0069
      DT(APRECH)=D(PPRES)*(TF(L1)+460.)/520.
      DT(MAVOLG)=D(MAVOLO)-D(MIVOLO)+D(MIVOLG)
      DT(VOLUME)=D(MAVOLO)+D(MIVOLG)
      DT(OVOLG)=D(MIVOLG)
      DT(OPG)=DT(APRECH)
      DT(HDELP)=.5*DELT
      DT(IQV)=0.0
      DT(DPG)=0.0
      IF(L(2).LE.0) RETURN
      TF(L(2))=D(ITF)
      TC(L(2))=D(ITC)
      RETURN
 2000 CONTINUE
C     CORRECT FLOW CONSTANT FOR FLUID AND TEMP
      CORR=VISC(TF(L(1)),PUP)*RHO(TF(L(1)),PUP)/(.028*3.2E-5)
     + *D(LOSS)
      PUP=PUP-Q1*CORR
      IF(ICON.NE.1)RETURN
      DT(PO)=PUP
      IF(TIME.NE.0.0)RETURN
      DT(PG)=DT(PO)
      DT(VOLG)=DT(MAVOLG)-(DT(APRECH)*DT(MAVOLG))/DT(PG)
      DT(VOLO)=DT(VOLUME)-DT(VOLG)
      IF(DT(APRECH).LT.DT(PG)) RETURN
      DT(PG)=DT(APRECH)
      DT(VOLO)=D(MIVOLO)
      RETURN
 3000 L1=L(1)
      Q2=0.0
      IF(L(2).GT.0)Q2=Q(L(2))
      KTYPE=D(KTYPE)+.001
      NTYPE=D(NTYPE)+.001
      RHOG=.00001
      RHOP=PROP(NTYPE,2)
      CA=PROP(KTYPE,3)
      CP=PROP(NTYPE,3)
      CPGN=247.9
      CPAN=PROP(KTYPE,1)
      CPPN=PROP(NTYPE,1)
      RHOIL=386.4*RHO(TF(L1),DT(PO))
C     COMPUTE PRESENT OIL VOLUME
      TQV=Q(L1)+Q2
      TVOLO=DT(VOLO)+(DT(IQV)+TQV)*DT(HDELP)
      WRITE(6,900)IND,NC(IND),L(2),Q(L1),Q2,DT(IQV),DT(DPG),
     + TVOLO,DT(VOLO)
```

6.71.7 (Continued)

```
  900 FORMAT(1X,3I10,6E15.5)
      DT(VOLG)=DT(VOLGNE)-TVOLO
      IF(TVOLO.GT.D(MIVOLO))GO TO 3100
C     ACCUMULATOR IS EMPTY
      TVOLO=D(MIVOLO)
      DT(VOLG)=DT(MAVOLG)
      TOV=0.0
      TDPG=0.0
      GO TO 3200
 3100 IF(TVOLO.LT.D(MAVOLO)) GO TO 3200
C     ACCUMULATOR IS FULL
      TVOLO=D(MAVOLO)
      DT(VOLG)=D(MIVOLG)
      TOV=0.0
      TDPG=0.0
C     ACCUMULATOR IN WORKING RANGE
 3200 CONTINUE
      TDPG=DT(PG)*TOV/(D(MAVOLG)-TVOLO+D(MIVOLO))
      DT(PG)=DT(PG)+(DT(DPG)+TDPG)*DT(DELT)
 3300 CONTINUE
      DT(VOLO)=TVOLO
      DT(IOV)=TOV
      DT(DPG)=TDPG
C     HEAT TRANSFER COMPUTATIONS
      DXAF=DT(VOLO)/(2.0*D(AREA))
      DXAA=DT(VOLG)/(2.0*D(AREA))
      DXAB=DXAF
      DXAP=D(DIA)/6.0
      ACAF=D(AREA)
      ACAA=2.0*D(WTHICK)*D(DIA)
      ACAB=2.0*D(WTHICK)
      ACAP=D(PMASS)/(RHOP*D(DIA))
      ASFB=PI*D(DIA)*DT(VOLO)/D(AREA)+D(AREA)
      ASGA=PI*D(DIA)*DT(VOLG)/D(AREA)+D(AREA)
      RMFL1=ABS(D(L1))*RHOIL
      D1=DT(VOLG)/(DT(VOLG)+DT(VOLO))
      D2=DT(VOLO)/(DT(VOLO)+DT(VOLG))
      GMASS=DT(VOLG)*RHOG
      AMASS=D(AMASS)*D1
      BMASS=D(AMASS)*D2
      FMASS=DT(VOLO)*RHOIL
 3101 D3=DXAB/(ACAB*CA)
      AAA=D(AREA)/2.0
      DDD=SQRT(AAA*4./PI)
C     UFMIL=UFM(AAA,DDD,ABS(D(L1)),TF(L1),P(L1))
      R1=CF/(DXF(L1)/ACF(L1)+DXAF/ACAF+RMFL1*DELT/(ACF(L1)
     + **2*RHOIL))
      R2=0.0
      R7=0.0
      RMFL2=0.0
```

6.71.7 (Continued)


```
      IF(Q(L1).LT.0.0) R1=0.0
      IF(Q(L1).LT.0.0) RHFL1=0.0
      IF(L(2).EQ.0.0)TF(L(2))=0.0
      IF(L(2).EQ.0.0) TA(L(2))=0.0
      IF(L(2).EQ.0) GO TO 3103
      L2=L(2)
      R2=CP/(DXF(L2)/ACF(L2)+DXAF/ACAF+RHFL2*DELT/(ACF(L2)
     + **2*RHOIL))
      RHFL2=ABS(Q(L2))*RHOIL
      R7=1.0/(DXF(L2)/(ACF(L2)*C(L2))+D3)
      IF(Q(L2).LT.0.0) R2=0.0
      IF(Q(L2).LT.0.0) RHFL2=0.0
 3103 R3=1.0/(DXAP/(ACAP*CP)+DXAA/(ACAA*CA)+1.0/(CAP*ACAP))
      R4=CA/(DXAA/ACAA+DXAB/ACAB)
      R5=1.0/(DXAP/(ACAP*CP)+D3+1.0/(CAP*ACAP))
      R6=1.0/(DXF(L1)/(ACF(L1)*C(L1))+D3)
      B1=D(UGA)*AUGA
      B2=D(UGA)*D(APGA)
      B3=D(UAA)*D(ACAA)
      B4=D(UFA)*D(APFA)
      B5=D(UPGA)*ASFG
C     GAS,FLUID,A,B PISTON NODES IN ORDER
      DT(TG)=DT(TG)*DT(PG)*DT(VOLG)/(DT(OVOLG)*DT(OPG))
      A(1,1)=GMASS*CPGN/DELT+B1+B2
      A(1,2)=0.0
      A(1,3)=-B1
      A(1,4)=0.0
      A(1,5)=-B2
      B(1)=GMASS*CPGN*DT(TG)/DELT
      A(2,1)=0.0
      A(2,2)=FMASS*CPFN/DELT+B4+B5+RHFL1*CPFN+
     + R1+R2+RHFL2*CPFN
      A(2,3)=0.0
      A(2,4)=-B5
      A(2,5)=-B4
      B(2)=FMASS*CPFN*DT(TAF)/DELT+(RHFL1*CPFN+R1)*TF(L1)+
     + (R2+RHFL2*CPFN)*TF(L2)
      A(3,1)=-B1
      A(3,2)=0.0
      A(3,3)=AMASS*CPAN/DELT+B1+R3+R4+B3*D1
      A(3,4)=-R4
      A(3,5)=-R3
      B(3)=AMASS*CPAN*DT(TAA)/DELT+CIP*D(ASAA)*D1*D(PST)-
     + CIP*D(ASAA)*D1*((DT(TAA)+460.)**4)+B3*D1*D(TA)
      A(4,1)=0.0
      A(4,2)=-B5
      A(4,3)=-R4
      A(4,4)=BMASS*CPAN/DELT+B5+R4+R5+R6+R7+
     + B3*D2
      A(4,5)=-R5
```

6.71.7 (Continued)

```
      B(4)=RMASS*CPMW*DT(TMW)/DELT+R6*TW(L1)+R7*TW(L2)+
     + R3*D2*D(IA)+CIP*D(ASAA)*D2*D(TOT)-
     + CIP*D(ASAA)*D2*((DT(TMW)+460.)**4)
      A(5,1)=-R2
      A(5,2)=-R4
      A(5,3)=-R3
      A(5,4)=-R5
      A(5,5)=RMASS*CPPW/DELT+R3+R5+R2+R4
      B(5)=RMASS*CPPW*DT(TPW)/DELT
      CALL SIMULT(A,B,5,IERROR)
      DT(CVOL3)=DT(VOL3)
      DT(OPG)=DT(PG)
      TFO=TF(L1)
      DT(TG)=B(1)
      DT(TAF)=B(2)
      DT(TAW)=B(3)
      DT(TMW)=B(4)
      DT(TPW)=B(5)
      TC(L1)=B(4)
      IF(L(2).LE.0) TF(L1)=B(2)
      IF(L(2).LE.0) GO TO 4000
      TF(L2)=B(2)
      TC(L2)=B(4)
 4000 IF(R(L1).LE.0.0) TF(L1)=TFO
      RETURN
      END
```

6.81 SUBROUTINE TFILT81

TFILT81 simulates an inline non-bypass filter with no moving parts as shown in Figure 6.81.1. This subroutine calculates the filter wall temperature and the temperature of the fluid in the filter bowl.

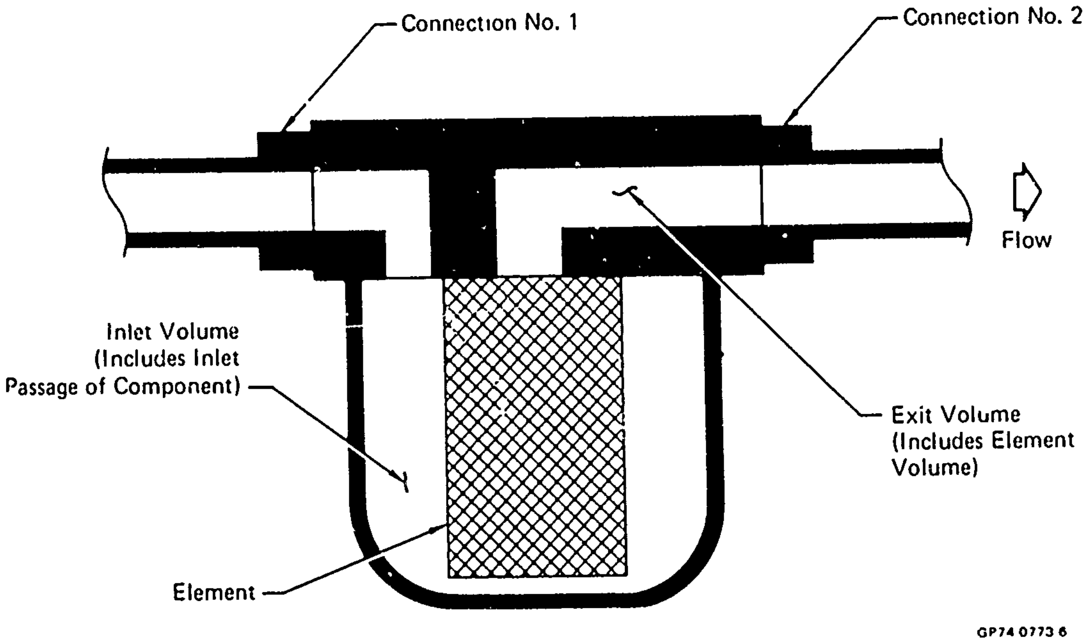


FIGURE 6.81-1
TYPE NO. 81 F-4 TYPE IN-LINE FILTER

6.81.1 Math Model

The thermal math model for the filter includes heat transfer to and from two line segments, one upstream and one downstream of the filter. Six nodes are considered: three fluid nodes and three wall nodes (as shown in Figure 6.81-2). The filter consists of two nodes: one fluid, representing all the fluid in the filter, and one wall, representing all the walls. The temperatures of the upstream and line segment wall and fluid nodes are denoted by TW(L1) and TF(L1), the temperatures of the filter wall and fluid nodes

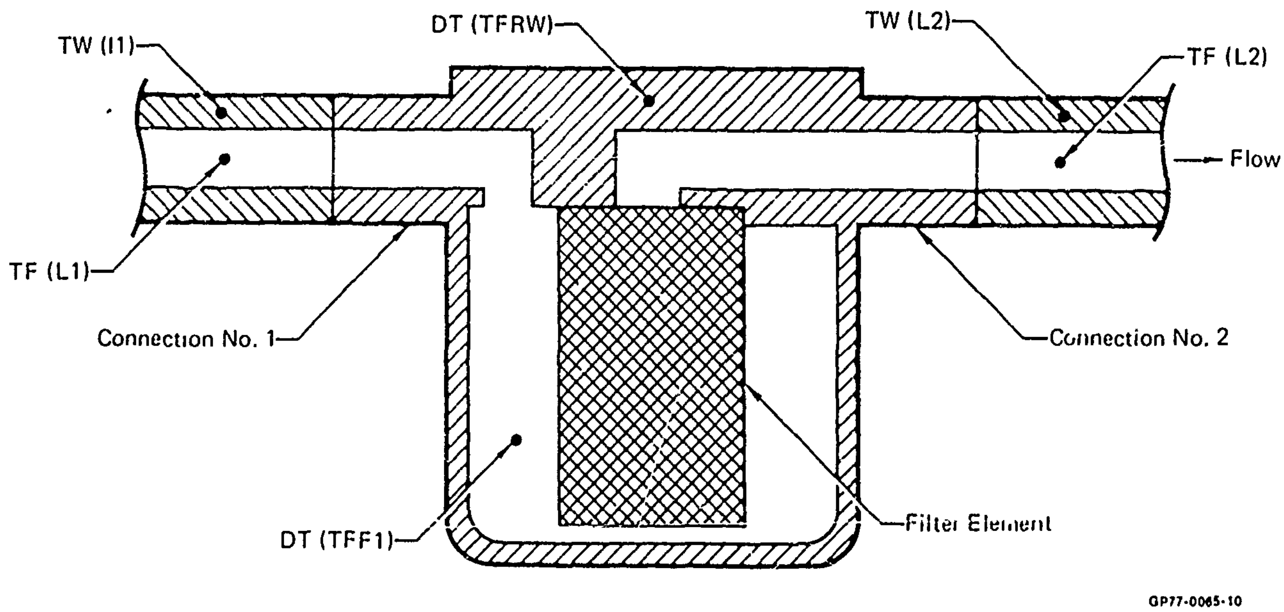


FIGURE 6.81-2
FILTER AND LINE SEGMENT NODE REPRESENTATION

Figure 6.81-2

FILTER AND LINE SEGMENT NODE REPRESENTATION

are DT(TFRW) and DT(TFF1), and the temperatures of the downstream line segment wall and fluid nodes are TW(L2) and TF(L2). Two heat balance equations are written to solve for DT(TFRW) and DT(TFF1), using the filter and line segments material properties and dimensions, the atmosphere and structure temperatures external to the filter, and TW(L1), TW(L2), and TF(L1). (Note: TF(L2) = DT(TFF1), see assumptions). One equation is a heat balance for the filter fluid node. The second equations is a heat balance for the filter wall node.

The first equation represents four modes of heat transfer relative to the filter fluid node:

1. Conduction to and from the upstream line segment fluid node

R3*(TF(L1)-DT(TFFL))

where R3 is the conduction coefficient between the fluids and is equal to CF/(DXF(L1)/ACF(L1)+DXFF/ACFF+RMFL1*DELT/(ACFF**2*RHOIL))

2. Convection to and from the filter wall node

B1*(DT(TFRW)-DT(TFF1))

where B1 is the convection coefficient between the fluid and the wall and is equal to UFWIL*D(ASFW).

3. Heat transfer due to mass transfer into the filter from the upstream line fluid segment.

$\dot{M}$Cp*(TF(L1) - DT(TFF1))

where $\dot{M}$Cp is the flow rate coefficient and is equal to Q(L1)*RHOIL*CPFN.

4. Heat addition due to a pressure drop across the filter

$\dot{M}$Cp * DCAPT

where DCAPT = (1.0/RHOIL)*(P(L1)-P(L2))/(CJ*CPFN)

Note: There may be a pressure drop across the filter and if sufficient may add heat to the fluid experiencing the pressure drop. If not an appreciable pressure drop, (100 psi or greater) this term will be negligible.

These four heat transfer terms combine to produce the equation for the heat balance for the filter fluid:

$$\frac{MCp}{DELT} * (DT(TFF1)-DT(TFF1)_{old}) = B1*(DT(TFRW)-DT(TFF1)) \qquad (1)$$
$$+ \dot{M}Cp*(TF(L1)-DT(TFF1))$$
$$+ \dot{M}Cp*DCAPT+R3*(TF(L1)-DT(TFF1))$$

where MCp is equal to FFMAS*CPFN

The second equation represents three modes of heat transfer relative to the filter wall:

1. Conduction to and from the upstream and downstream line segment walls

RI*(TW(L1)-DT(TFRW)

where R1 is the conduction coefficient and is equal to 1.0/(DXF(LI)/ACW(LI)*C(LI))+DXRW/(D(ACFW)*CW)) and I = 1 for the upstream line and 2 for the downstream line.

2a. Convection to and from the filter fluid mode

B1*(DT(TFF1)-DT(TFRW)

where B1 is defined above.

2b. Convection to and from the external atmosphere

B2*(D(TA)-DT(TFRW))

where B2 is the convection coefficient and is equal to D(UAF)*D(ASAF).

3. Radiation exchange with the surrounding structure

CIP*(D(TST)-(DT(TFRW)+460.)**4)

where CIP is the radiation coefficient and is equal to SIGMA*EPSION*SHAPF*D(ASAF).

these heat transfer modes combine to produce the equation for the heat balance of the filter wall node:

$$\frac{MCp*}{DELT}(DT(TFRW)-DT(TFRW)_{old}) = R1*(TW(L1)-DT(TFRW)) \quad (2)$$
$$+ R2*(TW(L2)-DT(TFRW))$$
$$+ B1*(DT(TFF1)-DT(TFRW))$$
$$+ B2*(D(TA)-DT(TFRW))$$
$$+ CIP*(D(TST))$$
$$-CIP*((DT(TFRW)+460)**4)$$

when MCp is equal to D(FMASS)*CPWN

Equations (1) and (2) are solved for the appropriate temperatures. A thermal model of the above heat transfer terms for the filter is shown in Figure 6.81-3.

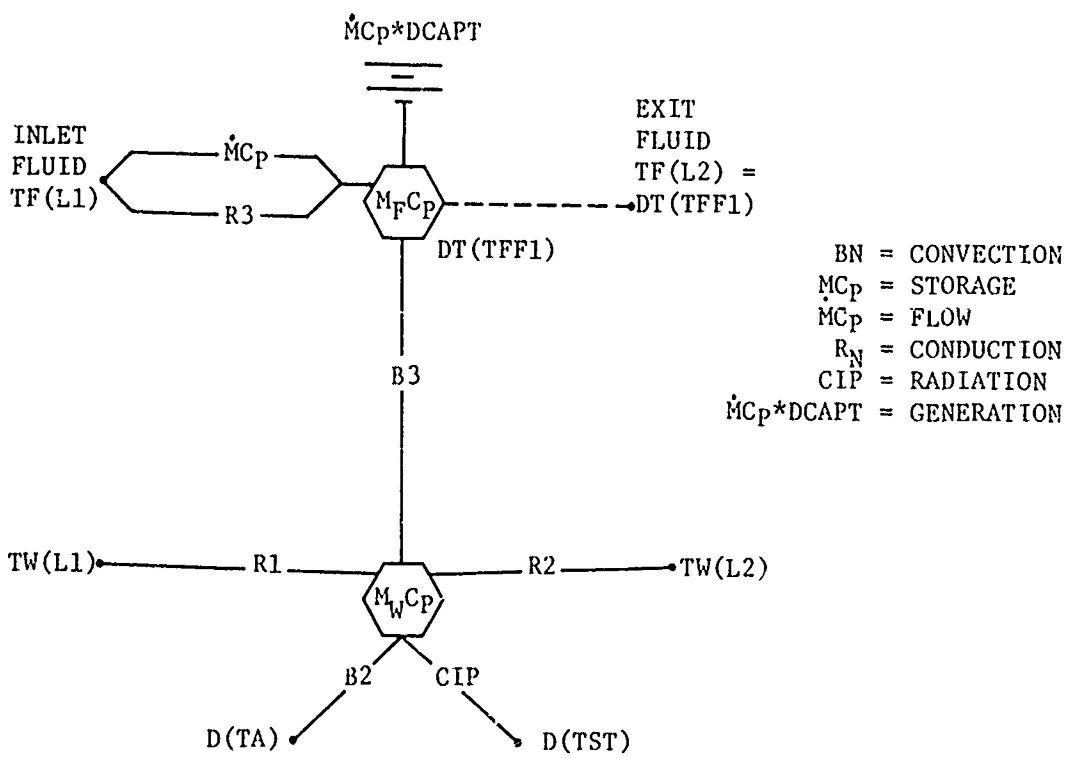


FIGURE 6.81-3

THERMAL MODEL

In the hydraulic math model a second order relationship is used to compute the filter pressure drop.

PUP = PUP-QA*QS*(COEFFL+QA*COEFFT) (4)

where PUP = upstream pressure (PSI)

QA = magnitude of flow (CIS)

QS = sign of flow

COEFFL = laminar flow coefficient (PSI/CIS)

COEFFT = turbulent flow coefficient (PSI/CIS^2)

6.81.2 Assumptions

1. The temperature of the fluid leaving the filter is equal to the filter fluid node temperature, DT(TFF1).
2. The entire filter wall is at the same temperature.
3. The temperatures of the atmosphere and structure surrounding the filter remains constant.
4. The interface conductance between the filter and line walls is infinite.
5. The emissivity of the wall material is a constant.
6. Complete fluid mixing occurs in the fluid volume.

6.81.3 Computational Methods

The subroutine executes the above discussed calculations as follows.

Section 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the forth power, and the default values are assigned.

Section 2000

The laminar flow coefficient D(CONSEL) and turbulent flow coefficient D(CONE2) are adjusted for fluid other than MIL-H-5606B and temperatures other than 100°F. Equation (4) is then solved to obtain the filter pressure prop.

Section 3000

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving thru connection line two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one.) Some coefficients are then recalculated if the flow is reassigned. A 2 x 2 matrix is loaded and the mathematical equations are solved for DT(TFF1) and DT(TFRW) and stored in the B computational array. The calculated values are assigned to their proper storage locations and this boundary conditions are assigned to arrays in COMMON/TRANS/for distribution throughout the entire program.

6.81.4 Approximations

Not applicable.

6.81.5 Limitations

Not applicable.

6.81.6 Variable Names

Variable	Description	Dimension
A( )	Computational array	--
AAA	Dummy variable	--
D(ACB)	Cross sectional area of the filter bowl	IN^2
ACFF	Cross sectional area of the fluid in the filter	IN^2
D(ACFW)	Cross sectional area of the filter walls at the connections	IN^2
D(ASAF)	External surface area of the filter	IN^2
D(ASFW)	Internal surface area of the filter	IN^2
A1, A2	Dummy variables	--
B( )	Computational array	--
CENT	Fluid viscosity	IN^2/SEC
CIP	Radiation coefficient	$WATTS/°R^4$
COEFFL	Viscosity corrected laminar flow coefficient	PSI/CIS
COEFFT	Viscosity corrected turbulent flow coefficient	PSI/CIS^2
CJ	Mechanical equivalent of heat	$FT\text{-}LB_m/WATTS\text{-}SEC$
D(CONE2)	Turbulent flow coefficient	PSI/CIS^2
D(CONSEL)	Laminar flow coefficient	PSI/CIS
CPWN	Specific heat of the filter walls	$WATTS\text{-}SEC/LB_m\text{-}°F$
CW	Thermal conductivity of the filter walls	WATTS/IN-°F
DCAPT	Heat added to fluid due to a pressure change	°F
DDD	Dummy variable	--
DENS	Fluid density	$LB_m\text{-}SEC^2/IN^4$
DXRW	Distance from wall node to interface with line segment	IN
EPSION	Emissivity factor of the walls	--
D(FMASS)	Mass of the filter walls	LB_m

Variable	Description	Dimension
FRFM	Mass of the fluid in the filter	LB_m
IERROR	Dummy variable	--
D(ITC)	Initial temperature of the filter walls	°F
D(ITF)	Initial temperature of the fluid	°F
L1,L2	Line connection addresses	--
MTYPE	Material type of the walls	--
D(PERC)	Percentage heat added to the fluid due to a pressure drop	--
RHOIL	Fluid density	LB_m/IN^3
RHOW	Density of filter walls	LB_m/IN^3
RMFL1	Mass flow rate entering filter	LB_m/SEC
RMFL2	Mass flow rate leaving filter	LB_m/SEC
R1,R2	Dummy variables	--
SHAPF	Shape factor case to surrounding structure	--
SIGMA	Stefan-Boltzman constant for radiation	WATTS /IN^2-°R^4
D(TA)	Temperature of the surrounding atmosphere	°F
TEMP1	Dummy variable	--
DT(TFF1)	Filter fluid temperature	°F
DT(TFRW)	Filter wall temperature	°F
D(TST)	Temperature of the surrounding structure	°F
D(UAF)	Heat transfer coefficient (surrounding atmosphere to filter walls)	WATTS/IN^2-°F
UFWIL	Heat transfer coefficient (fluid to filter walls)	WATTS/IN^2-°F
D(VOLUME)	Volume of fluid inside filter	IN^3

6.81.7 Subroutine Listing

```
      SUBROUTINE TFILT81 (D,DT,DD,L)
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINL,NLL
      COMMON /COMP/LTYPL(99),NC(99),KTEMP(99),IND,IENTR,INLL
      COMMON /STEADY/PN(90),QN(90),PLX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,NNODL,NLEG,NCPN,TERM,LEGN,ICON,INV,
     + INX,INZ,NUP(90),NDWN(90),NELEM(90),ILEGAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION A(2,2),B(2)
      INTEGER UAF,ASAF,FMASS,TA,TST,TFF1,TFRW,ITC,ITF,ASFW,
     + VOLUME,ACFW,ACB,CONSLL,CONE2,PERC
C     D ARRAY VARIABALES
      DATA MTYPL/1/,FMASS/2/,VOLUME/3/,ACFW/4/,ACB/5/
     + ,ASAF/6/,ASFW/7/,UAF/8/,TST/10/,TA/11/,ITF/12/,ITC/13/
     + ,CONSEL/14/,CONE2/15/,PERC/9/
      DATA SIGMA/.349E-11/,SHAPF/.93/,EPSION/.3/,CJ/8.85/
C     DT ARAAY VAARIABALLS
      DATA TFF1/1/,TFRW/2/
C     UAF   =HEAT TRANSFER COEFFICIENT CASE AWALL TO AMBIENT
C     ASAF  =SURFACE AREA CASE WALL TO AMBIENT
C     FMASS  =FILTER WALL MASS,LBS. (TOTAL WEIGHT)
C     TFF1  =TEMPERATURE OF THE FLUID
C     TFRW  =    "   "      OF THE CASSE WALL
C     VOLUME=TOTAL VOLUME OF FLUID IN CASE
C     ASFW  =SURFACE AREA CASE WALL TO INSIDE FLUID
C     TA    =TEMPERATURE OF SURROUNDING AMBIENT
C     TST   =TEMPERATURE OF     ""        STRUCTURL
C     ACFW=DISTANCE INLET TO EXIT
C     ACB=CROSS SECTIONAL ARLA OF THE INSIDE OF THE BOWL
      IF(IENTR) 1000,2000,3000
C *** 1000 SECTION
 1000 CONTINUE
      L1=L(1)
      L2=L(2)
      DT(TFF1)=D(ITF)
      DT(TFRW)=D(ITC)
      IF(D(CONSEL).LE.0.0) D(CONSEL)=.0001
      D(TST)=(D(TST)+460.)**4
      IF (D(UAF).EQ.0.0) D(UAF)=.0069
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
      TC(L1)=D(ITC)
      TC(L2)=D(ITC)
      KTYPL=D(MTYPL)+.001
      RHOW=PROP(KTYPL,2)
      CW=PROP(KTYPL,3)
      CPWJ=PROP(KTYPL,1)
      RETURN
C *** 2000 SECTION
```

```
 2000 CONTINUE
      DENS=RHO(TF(L(1)),PUP)
      CENT=VISC(TF(L(1)),PUP)
      COEFFL=CENT*DENS*D(CONSEL)/(.028*8.2E-5)
      COEFFT=CENT**.25*DENS*D(CONE2)/(.40906234*8.2E-5)
      PUP=PUP-QA*QS*(COEFFL+QA*COEFFT)
      RETURN
 3000 L1=L(1)
      L2=L(2)
      IF(Q(L1).GT.0.0) GO TO 3003
      L1=L(2)
      L2=L(1)
      RHOIL=386.4*RHO(TF(L1),P(L1))
      FFMASS=D(VOLUME)*RHOIL
 3003 AAA=D(ACB)/2.
      DDD=SQRT(AAA*4./PI)
      TEMP1=DT(TFRW)
      UFWIL=UFW(AAA,DDD,ABS(Q(L1)),TF(L1),P(L1))
      DXRW=D(VOLUME)/(2.0*D(ACFW))
      ACFF=D(ACB)/3.
      DXFF=FFMASS/(RHOIL*D(ACB)*2.)
 3033 RMFL1=ABS(Q(L1))*RHOIL
      RMFL2=ABS(Q(L2))*RHOIL
      R1=1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXRW/(D(ACFW)*CW))
      R2=1.0/(DXF(L2)/(ACW(L2)*C(L2))+DXRW/(D(ACFW)*CW))
      R3=CF/(DXF(L1)/ACF(L1)+DXFF/ACFF+RMFL1*DELT/(ACFF**2*RHOIL))
      B1=UFWIL*D(ASFW)
      B2=D(UAF)*D(ASAF)
      DCAPT=(1./RHOIL)*ABS(P(L1)-P(L2))/(CPFN*CJ)
      CIP=SIGMA*SHAPF*EPSION*D(ASAF)
C     TFF1,TFRW NODES IN ORDER
 3099 A(1,1)=FFMASS*CPFN/DELT+RMFL1*CPFN+B1+R3
      A(1,2)=-B1
      B(1)=FFMASS*CPFN*DT(TFF1)/DELT+RMFL1*CPFN*TF(L1)
     + +RMFL1*CPFN*DCAPT*D(PERC)+R3*TF(L1)
      A(2,1)=-B1
      A(2,2)=D(FMASS)*CPWN/DELT+B2+B1+R1+R2
      B(2)=D(FMASS)*CPWN*DT(TFRW)/DELT+R1*TW(L1)+R2*TW(L2)
     + +B2*D(TA)+CIP*D(TST)-CIP*((DT(TFRW)+460.)**4)
      CALL SIMULT(A,B,2,IERROR)
      TF(L2)=B(1)
      TC(L1)=B(2)
      TC(L2)=B(2)
      DT(TFF1)=B(1)
      DT(TFRW)=B(2)
      RETURN
      END
```

6.101 SUBROUTINE ACT101

TACT101 simulates a simple servo actuator with a mechanical input to the servo valve, which operates open loop, without feedback as shown in Figure 6.101-1.

A time history of valve position is inputed and a first order or straight line interpolation is used between the input points.

The valve is assumed to be a linear square port configuration, with zero lap. The width of each port slot is inputed independently, to allow the valve areas to be matched to the actuator piston areas. The initial actuator position is input, together with the external loads at the fully retracted and extended stroke positions. The load stroke curve is assumed to be linear between these positions. The steady state balancing system uses the load at the initial position to determine the pressure drop across the piston. The effects of atmospheric pressure is incorporated into the load.

The subroutine calculates two actuator fluid temperatures, two valve fluid temperatures, two actuator wall and one actuator piston temperatures, and one valve wall temperature.

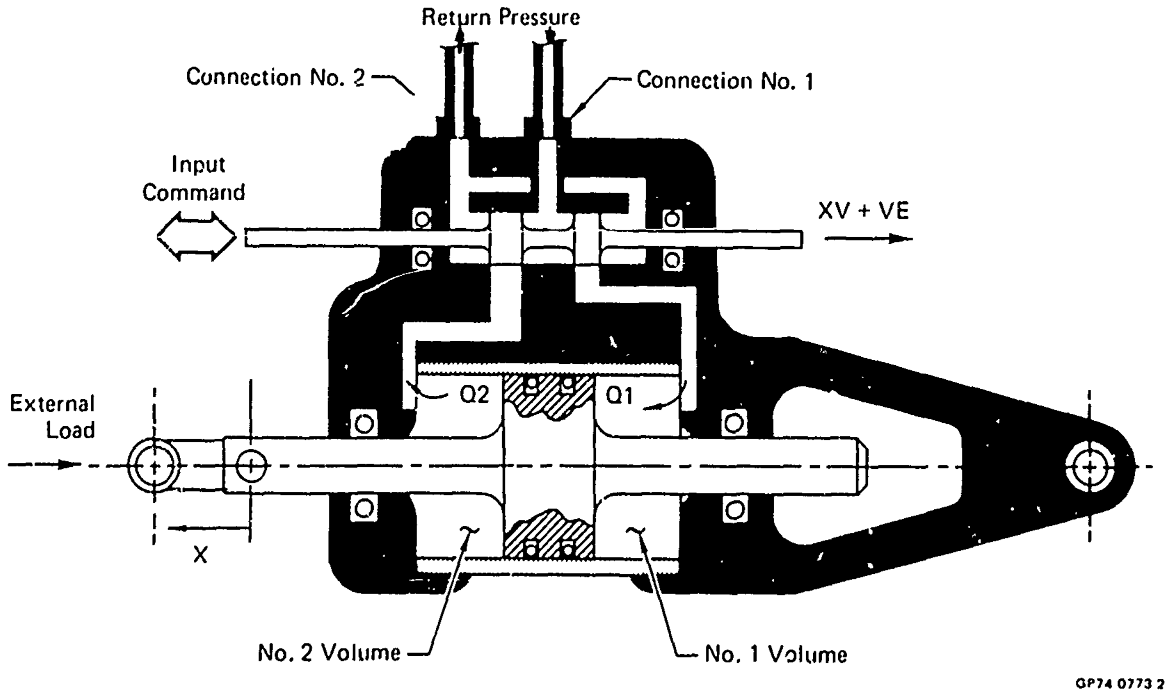


FIGURE 6.101-1
TYPE NO. 101 VALVE CONTROLLED ACTUATOR

6.101-1 Math Model

The thermal math model for the actuator includes heat transfer to and from two connecting line segments, one upstream and one downstream of the actuator valve. For the actuator valve combination there are a total of 12 nodes: six fluid nodes, five wall nodes, and one piston node, as shown in Figure 6.101-2.

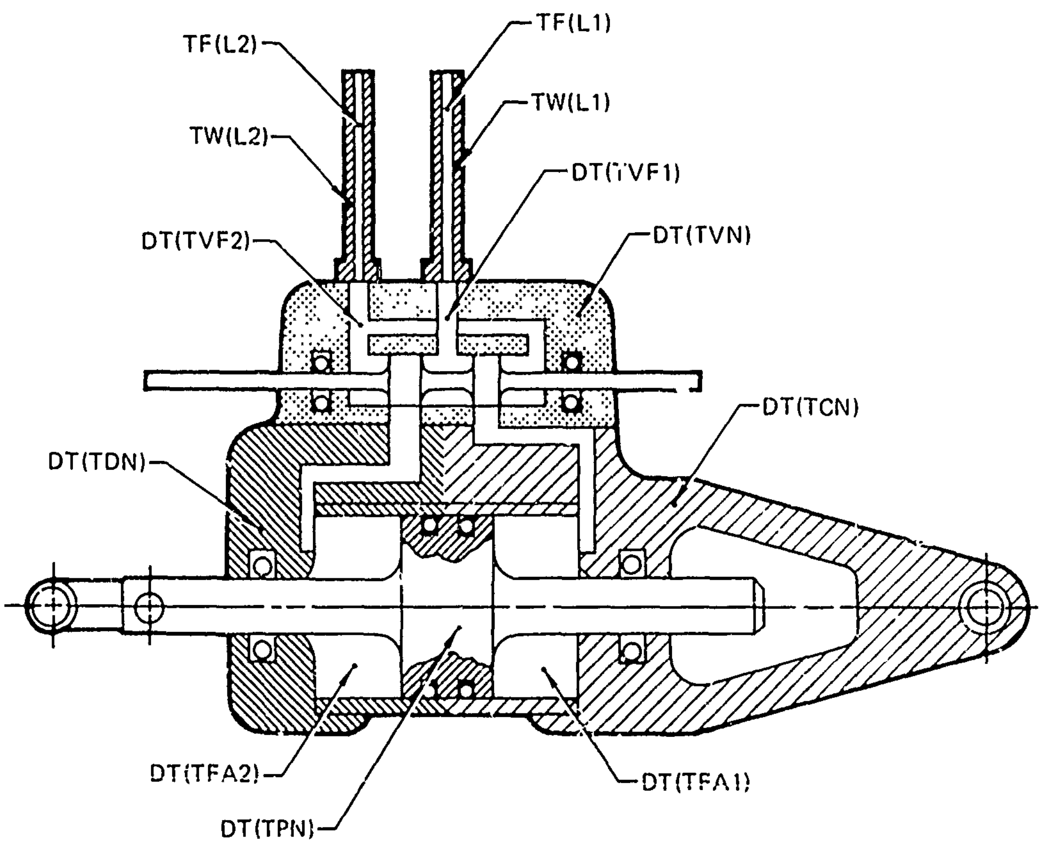


FIGURE 6.101-2
ACTUATOR, VALVE AND LINE SEGMENT NODE REPRESENTATION

GP77-0065-13

The temperatures of the upstream line segment nodes are TF(L1) and TW(L1) for the fluid and wall respectively, the temperatures of the downstream line segment nodes are TF(L2) and TW(L2) for the fluid and wall respectively. The actuator valve nodes temperatures are DT(TVF1), DT(TVF2), and DT(TVN) for two fluids and the valve wall respectively. The actuator wall temperatures are DT(TCN) for the walls around volume one, and DT(TDN) for the walls around volume two. The two actuator fluid nodes temperatures are DT(TFA1) for the fluid in volume one, and DT(TFA2) for the fluid in volume two, and the actuator pistons node temperature is designated as DT(TPN).

Eight equations are written to solve for the eight valve and actuator temperatures, using the actuator, valve and line segment material properties and dimensions, and external atmosphere and structure temperatures of the actuator and valve. The equations represent the heat transferred to and from each of the eight actuator nodes.

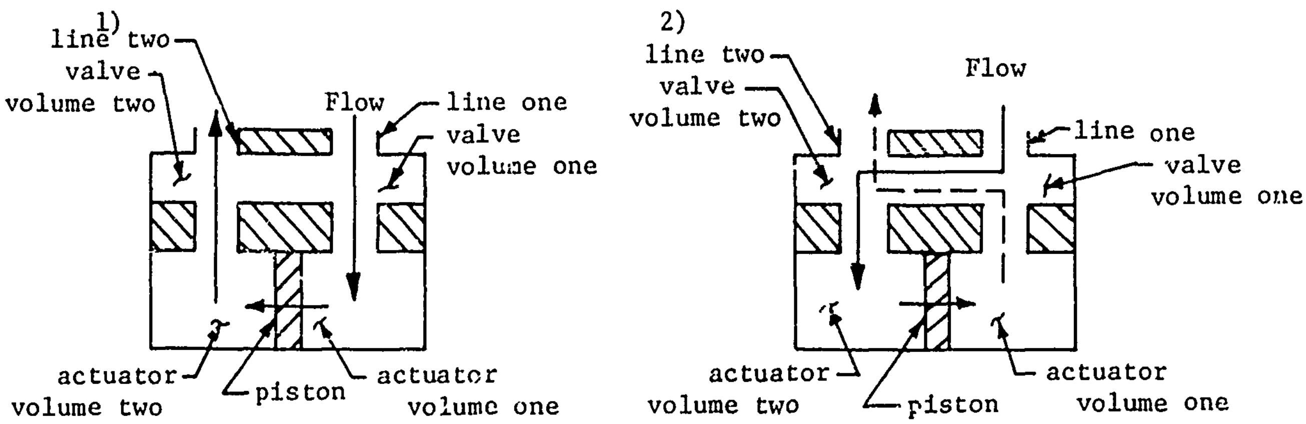


FIGURE 6.101-3

ACTUATOR WORKING SIMULATIONS

During operation of the actuator, fluid always enters connection one, or flows into valve volume one, and then has two possible paths as shown in Figure 6.101-3. It either: 1) m ay enter actuator volume one. If this happens fluid from actuator volume two then leaves actuator volume two, due to movement of the piston, and travels into valve volume two. The fluid then leaves valve volume two to line two, or 2) may enter actuator volume two moving the piston which forces fluid out of actuator volume one into valve volume two which then leaves valve volume two again to line two.

To describe the math model we shall only consider the first path. Recall that eight equations are necessary. The first equation represents four modes of heat transfer relative to the valve volume one fluid.

1) Conduction to and from the upstream line fluid node

$$\mathrm{R15}*(\mathrm{TF(L1)} - \mathrm{DT(TVF1)})$$

where R15 is the conduction coefficient for the fluids equal to CF/(DXF(L1)/ACF(L1) + DXV/ACFV + RMFL1*DELT/(ACFV**2*RHOIL)) and RMFL1 is the mass flow rate equal to Q(L1)*RHOIL

2) Heat transfer due to mass transfer of fluid into the valve from the upstream line segment

$$\dot{M}\mathrm{Cp}*(\mathrm{TF(L1)} - \mathrm{DT(TVF1)})$$

where $\dot{M}$Cp is the mass transfer term and is equal to RMFL1*CPFN

3) Convection to and from the valve wall node

$$\mathrm{B7}*(\mathrm{DT(TVN)} - \mathrm{DT(TVF1)})$$

where B7 is a convection coefficient and is equal to UFWIL*F(ASFV)/2.0

4) Heat added directly to the fluid due to a pressure drop from line one to the valve volume.

$$\dot{M}\mathrm{Cp}*\mathrm{DCAPT1}$$

where $\dot{M}$Cp is as defined previously and DCAPT1 is equal to 1.0/RHOIL* (P(L1) - DT(PP1))/ (CJ*CPFN*2.)

These terms are combined to produce the equation for heat balance for a volume one.

$$\frac{\mathrm{MCp}}{\mathrm{DELT}}\left(\mathrm{DT(TVF1)} - \mathrm{DT(TVF1)}_{\mathrm{OLD}}\right) = \mathrm{R15}*(\mathrm{TF(L1)} - \mathrm{DT(TVF1)}) + \dot{M}\mathrm{Cp}\,(\mathrm{TF(L1)} - \mathrm{DT(TVFL1)}) + \mathrm{B7}*(\mathrm{DT(TVN)} - \mathrm{DT(TVF1)}) + \dot{M}\mathrm{Cp}*\mathrm{DCAPT1} \quad (1)$$

when MCp = FMASS*CPFN

The second equation represents three modes of heat transfer relative to the actuator volume one.

1) Heat transfer due to mass transfer into the actuator volume from the valve volume one

$$\dot{M}Cp*(DT(TVF1) - DT(TFA1))$$

where $\dot{M}Cp$ is equal to Q(L1)*RHOIL*CPFN

2a) Convection to and from the actuator walls surrounding volume one B3*(DT(TCN) - DT(TFA1))

where B3 is the convection coefficient equal to

UA1C*ASA1C

2b) Convection to and from the piston node

$$B5*(DT(TPN) - DT(TFAL))$$

where B5 is the convection coefficient equal to D(UA1P)*D(AREA1).

3) Heat added directly to the fluid due to a pressure drop across the orifice into the actuator.

$$\dot{M}Cp*DCAPT1$$

$\dot{M}Cp$ is equal to RFML1*CPFN and DCAPT is the same as defined previously.

These terms are combined to produce the heat balance equation for the actuator volume one.

$$\frac{MCp}{DELT}(DT(TFA1) - DT(TFA1)_{OLD}) = \dot{M}Cp*(DT(TVF1) + DCAPT1 - DT(TFA1)) + B3*(DT(TCN)-DT(TFA1)) + B5*(DT(TPN) - DT(TFA1)) \quad (2)$$

where $\dot{M}Cp$ is equal to FMASS1(CPFN).

The third equation represents one mode of heat transfer relative to the actuator volume two.

1a) Convection to and from the actuator wall surrounding volume two

$$B4*DT(TDN) - DT(TFA2))$$

where B4 is equal to UA2D*ASA2D.

1b) Convection to and from the actuator piston node

$$B6*(DT(TPN) - DT(TFA2))$$

where B6 is equal to UA2P*D(AREA2).

These terms combine to form the heat balance equation for the actuator exit volume two.

$$\frac{MCp}{DELT} *(DT(TFA2) - DT(TFA2)_{OLD}) = B4*(DT(TDN) - DT(TFA2)) + B6* (DT(TPN) - DT(TFA2)) \qquad (3)$$

where MCp is equal to FMASS2*CPFN.

The fourth equation represents three modes of heat transfer relative to the valve volume two, the exit volume.

1) Heat transfer due to mass transfer of fluid into the valve volume from actuator volume two.

$$\dot{M}Cp*(DT(TFA2) - DT(TVF2))$$

where $\dot{M}Cp$ is equal to RMFL2*CpFN and RMFL2 is equal to Q(L2)*RHOIL.

2) Convection to and from the valve wall node

$$B7*(DT(TVN) - DT(TVF2))$$

and B7 has been defined previously.

3) Heat added directly to the fluid due to a pressure drop into the actuator volume two.

$$\dot{M}Cp*DCAPT2$$

where $\dot{M}Cp$ has just been defined and DCAPT2 is equal to

1.0/RHOIL*(DT(PP2) - P(L2))/(CJ*CPFN*2.)*2.0.

These terms combine to produce the equation for the heat transferred to and from the valve exit volume two.

$$\frac{MCp}{DELT} *(DT(TVF2) - DT(TVF2)_{OLD}) = \dot{M}Cp*(DT(TFA2) - DT(TVF2)) + B7*(DT(TVN) - DT(TVF2)) + \dot{M}Cp*DCAPT2 \qquad (4)$$

where MCP is equal to FMASS*CPFN.

The fifth equation represents three modes of heat transfer to and from the valve walls.

1a) Conduction to and from the upstream line segment wall node

R1*(TW(L1) - DT(TVN))

where R1 is the conduction coefficient for the walls equal to 1.0(DXF(L1)/(ACW(L1)*C(L1) + DXV/(ACV*CV)) with the interface conductance between the two nodes being infinite.

1b) Conduction to and from the downstream line segment wall node

R2*(TW(L2) - DT(TVN))

and R2 is equal to R1 with L1 replaced by L2

1c) Conduction to and from the actuator wall node surrounding volume one.

R3*(DT(TCN) - DT(TVN))

where R3 is equal to 1.0/(DXV/(ACV*CV) + DXC/(ACC*CC) + 1.0/(D(ACCV)*D1*D(CCV))) and D1 represents the amount of mass that surrounds the actuator volume one and is equal to DT(VOL1)/(DT(VOL1) + DT(VOL2))

1d) Conduction to and from the actuator wall node surrounding volume two, equal to DT(VOL1)/(DTVOL1) + DT(VOL2))

R4*(DT(TDN) - DT(TVN))

where R4 is equal to 1.0/(DXV/(ACV*CV) + DXD/(ACD*CC) + 1.0/(D(ACCV)*D2*D(CCV)). D2 represents the amount of mass that surrounds the actuator volume two, and is equal to

DT(VOV2)/(DT(VOL1) + DT(VOL2).

2a) Convection to and from the valve fluid in the entrance volume one.

B7*(DT(TVF1) - DT(TVN))

where B7 was defined previously.

2b) Convection to and from the valve fluid in the exit volume two.

B7*(DT(TVF2) - DT(TVN))

2c) Convection to and from the external atmosphere

A2*(D(TA) - DT(TVN))

where A2 is the convection coefficient for the walls and is equal to D(UAV)*D(ASAV).

3) Radiation exchange with the surrounding structure

$$CIP1*(D(TST) - (DT(TVN) + 460)^4)$$

where CIP1 is the radiation coefficient equal to SIGMA*SHAPF* EPSION*D(ASAV).

These terms are combined to produce the equation for the heat balance for the valve walls.

$$\frac{MCp}{DELT}\ (DT(TVN) - DT(TVN)_{OLD}) = R1*(TW(L1) - DT(TVN)) + R2*(TW(L2) - DT(TVN)) + R3*(DT(TCN) - DT(TVN)) + R4*(DT(TDN) - DT(TVN)) + B7*(DT(TFA1) + DT(TFA2) - 2*DT(TVN)) + A2*(D(TA) - DT(TVN)) + CIP1*(D(TST)) - CIP1*(DT(TVN) + 460.)**4 \qquad (5)$$

The sixth equation represents three modes of heat transfer relative to the actuator wall node surrounding volume one.

1a) Convection to and from the fluid in volume one.

B3*(DT(TFA1) - DT(TCN))

where B3 is the same as defined previouslv.

1b) Convection to and from the surrounding atmosphere

B1*(D(TA) + DT(TCN))

where B1 is equal to UAC*D(ASAC)*D1.

2a) Conduction to and from the actuator wall node that surrounds volume two

R5*(DT(TDN) - DT(TCN))

where R5 is the conduction coefficient equal to CC/(DXD/ACD + DXC/ACC).

2b) Conduction to and from the valve wall

R3*(DT(TVN) - DT(TCN))

with R3 defined previously.

2c) Conduction to and from the piston node

R9*(DT(TPN) - DT(TCN))

where R9 is equal to 1.0/(DXP/(ASCP*CP) + DXC/(ACC*CC)).

3) Radiation exchange with the surrounding structure

$$D1*C1P2*(D(TST)-(DT(TCN)+460)^4)$$

where C1P2 is the radiation coefficient equal to SIGMA*SHAPE*EPSION*D(ASAC).

The terms then combine to produce the heat balance equation for the actuator wall node around volume one.

$$\frac{MCp}{DELT} *(DT(TCN) - DT(TCN)_{OLD}) = B1*(D(TA) - DT(TCN)) +$$

$$B3*(DT(TFA1) - \qquad (6)$$

$$DT(TCN)) + R5*(DT(TDN) - DT(TCN)) + R3*(DT(TVN) -$$

$$DT(TCN)) + R9*(DT(TPN) - DT(TCN)) + D1*C1P2*(D(TST)) -$$

$$D1*CIP2*(DT(TCN) + 460)**4$$

where MCp is equal to CMASS*CPCN.

The seventh equation represents two modes of heat transfer relative to the actuator piston.

1) Convection to and from the two actuator fluids in volume one and volume two, respectively, with coefficients defend previously

$$B5*(DT(TFA1) - DT(TPN))$$

and

$$B6*(DT(TFA2) - DT(TPN)).$$

2) Conduction to and from the two actuator wall nodes surrounding volumes one and two respectively, with the terms being defined previously

$$R9*(DT(TCN) - DT(TPN)) \text{ and}$$

$$R12* (DT(TDN) - DT(TPN)).$$

These terms combine to produce the heat balance equation with the actuator piston.

$$\frac{MCp}{DELT} *(DT(TPN) - DT(TPN)_{OLD}) = B5*(DT(TFA1) - DT(TPN)) + B6*(DT(TFA2) - DT(TPN))+ R9*(DT(TCN) - DT(TPN)) + R12* (DT(TDN) - DT(TPN))$$

where MCp is equal to PMASS*CPFN.

The eighth equation represents three modes of heat transfer relative to the actuator wall node surrounding actuator volume two.

1a) Convection to and from the fluid in volume two

$$B4*(DT(TFA2) - DT(TDN))$$

where B4 is a convection coefficient equal to UA2D*ASA2D

1b) Convection to and from the surrounding atmosphere

B2*(D(TA)*DT(TDN))

where B2 is equal to D2*UAC*D(ASAC) and D2 is a variable to calculate the wall mass that surrounds volume two, equal to DT(VOL2)/(DT(VOL1) + DT(VOL2)).

2a) Conduction to and from the other actuator wall node

R5*(DT(TCN) - DT(TDN))

with R5 defined previously.

2b) Conduction to and from the valve wall mass

R4*(DT(TVN) - DT(TDN))

where R4 is the conduction coefficient between the walls equal to 1.0/(DXV/(ACV*CV) + DXD/(ACD*CC) + 1.0/(D(ACCV)*D2*D(CCV)))

2c) Conduction to and from the actuator piston

R12*(DT(TDN) - DT(TDN))

where R12 is equal to 1.0/(DXP/(ASCP*CP) + DXD/(ACD*CC))

3) Radiation exchange with the surrounding structure

$$D2*CIP2*(D(TST) - (DT(TDN) + 460)^4)$$

with these terms the same as defined previously.

These terms then combine to produce the heat balance equation for the actuator wall node

$$\frac{MCp}{DELT} *(DT(TDN) - DT(TDN)_{OLD}) = B4*(DT(TFA2) - DT(TDN)) + B2*(D(TA) - DT(TDN)) + R5*(DT(TCN) - DT(TDN)) + R4*(DT(TVN) - DT(TDN)) + CIP2*D2* D(TST) - CIP2*D2*(DT(TDN) + 460)**4$$

where MCp is equal to DMASS*CPDN. A thermal model of the above 8 equations is shown in Figure 6.101-4. Equations (1) thru (8) are solved for the appropriate temperatures.

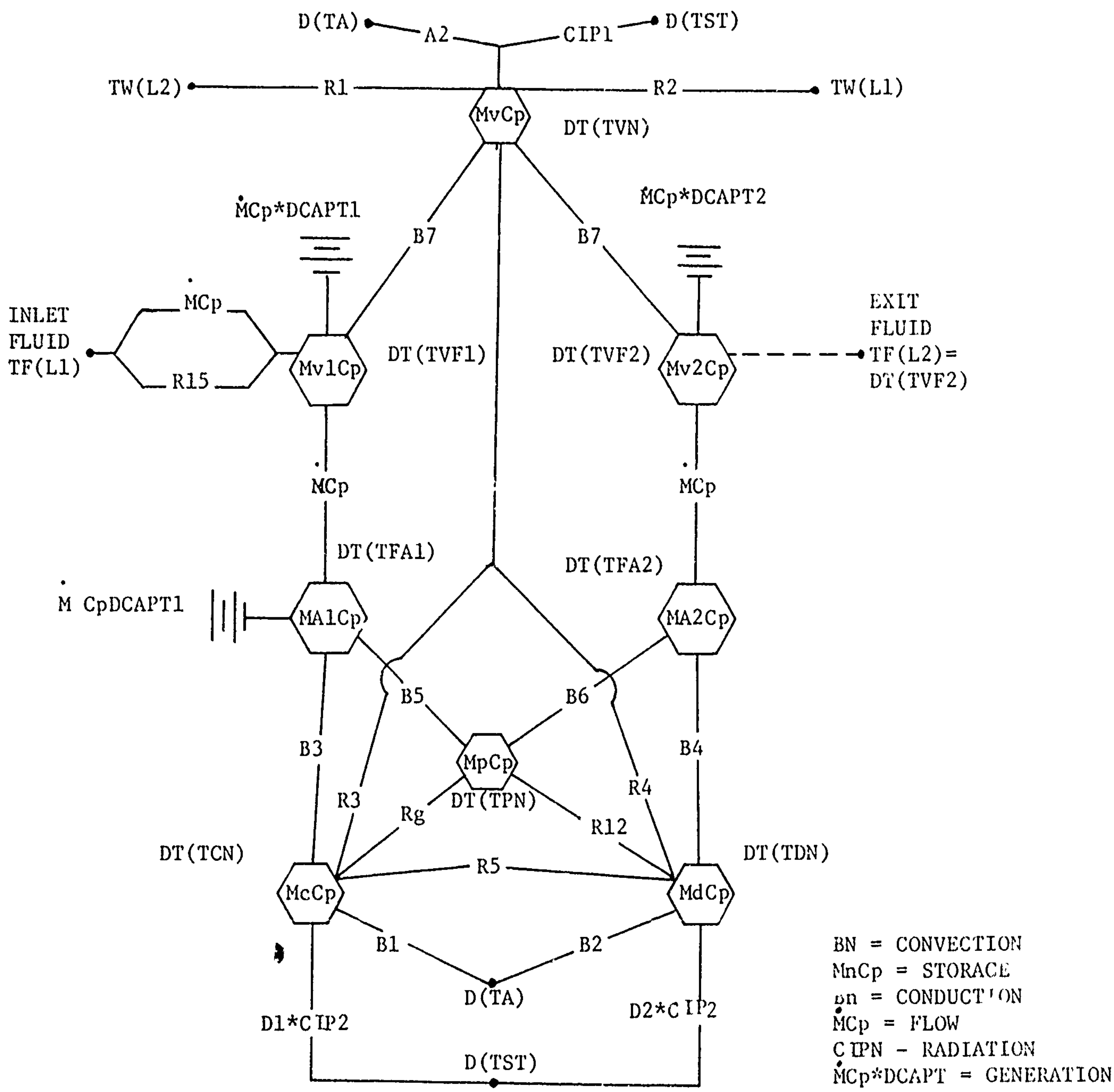


FIGURE 6.101-4
THERMAL MODEL

In the hydraulic math model the total actuator load is given by

DT(FORCE) - DT(LOADZ) + DT(LOADS)*DT(X)

where

DT(LOADZ) - LOAD AT ZERO STROKE (LB)

DT(LOADS) = LOAD/STROKE SLOPE (LB/IN)

DT(X) = ACTUATOR POSITION (IN)

The valve opening is determined from interpolation of the valve input data at the current time step. Depending on the direction of the valve movement the overboard flow at the actuator node is calculated. For an actuator that is extending

QN(N) = -DT(NCAV)*Q1*(D(AREA1) - D(AREA2))/D(AREA1) (9)

The pressure gain or loss across the piston is calculated using the force balance equation

DT(PFORCE) = (-PN(N)*(D(AREA1) - D(AREA2)))/D(AREA2) + QS*D(DAMP)/D(AREA2) (10)

If the actuator where retracting equations (9) and (10) would become

QN(N) = - DT(NCAY)*QL*(D(AREA1) - D(AREA2))/D(AREA2) (11)

and

DT(PFORCE) = (-PN(N)*(D(AREA2) - D(AREA1)) + DT(PFORCE))/D(AREA1) - (12)

QS*D(DAMP)D(AREA1)

6.101.2 Assumptions

1) The fluid exiting from the actuator valve to the connecting line is equal to DT(TVF2).

2) The interface conductance betwen the piston and the actuator walls is infinite.

3) Complete mixing occurs in all fluid volumes.

4) Piston and valve leakages are negligible.

5) The emissivity of the walls remains constant, at .3 for steel.

6) The atmosphere and structure temperatures remain constant.

6.101.3 Computational Methods

Section 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned. Compute load/stroke slope, determine valve position and compute the coefficient for the valve opening.

Section 2000

This section is called from TLEGCAL via COMPE for each connection number for each iteration. Calls are made for each iteration because the overboard flow and pressure drop across the piston head very with the flow into the actuator and the pressure in the piston cavity.

One of the cavities is required to be a system node. Which cavity it is depends on the valve position at the current time step. If NODE = 1 it is in #1 cavity, if NODE = 2 it is in #2 cavity.

The steady state section is complicated by the need to determine if the actuator is at its stroke limits, and if the flow guess is taking it toward or away from the limit.

When it is at its limits, and is being driven into the limit, a high impedance is added into the leg, and the overboard flow is set to zero. (Overboard flow is a displacement flow due to unequal areas).

The steady state calculation set up requires that connection #1 must be the last or only element in the upstream leg, and connection #2 is the first element in the downstream leg.

The upstream leg flow is used to calculate the overboard flow and piston velocity. If the valve is closed the overboard flow is set to zero.

For the upstream leg the valve impedance DT(PP1P) is added into PUP of that leg. For the downstream leg, the valve impedance PT(PP2P) is added into PUP and the constant pressure drop DT(PFORCE) across the piston is subtracted from PUP or added if it is a pressure rise.

Section 3000

This section calculates the thermal transient response of the actuator.

INTERP is called to obtain an interpolated value of valve position XV. With this value of XV, the flows into the actuator chambers are calculated.

If XV is zero, the flows are set to zero. For XV >0, Q1 is the flow from connection #1 to chamber #1, and Q2 the flow from chamber #2 to connection #2. For XV<0 the flows are reversed.

From the valve position recalculate the flow coefficients through the valve. The position of the actuator piston is computed using a simple integration.

DT(X) = DT(X) + (DT(VEL) + DT(VELO))*DELT/2.

The cylinder volumes are easily calculated as

DT(VOL1) = DT(VOL1) +(DT(x)-XO)*D(AREA1)

DT(VOL2) = DT(VOL2) -(DT(x)-XO)*D(AREA2)

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving thru connection line two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 8 x 8 matrix is loaded and the mathematical equations are solved for DT(TVF1), DT(TFA1), DT(TFA2), DT(TVFL), DT(TVN) DT(TCN), DT(TDN) and DT(TPN) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions are on distribution throughout the entire program.

6.101.4 Approximations

1) All input heat transfer and interface coefficients remain constant.

2) External temperatures all remain constant.

6.101.5 Limitations

This straight line flow characteristics of the valve and the straight line load characteristics limit the applicability of this subroutine to a simple type of actuator.

6.101.6 Variable Listing

Variable	Description	Dimension
A( )	Dummy computational array	IN^2
AAA	Dummy variable	IN^2
ACA1	Cross sectional area actuator volume one	IN^2
ACA2	Cross sectional area actuator volume two	IN^2
ACC	Cross sectional area actuator wall around volume	IN^2
D(ACCV)	Contact area between the valve and the actuator walls	IN^2
ACD	Cross sectional area actuator wall around volume two	IN^2
ACFV	Cross sectional area of the fluid in the valve	IN^2
ACV	Cross sectional area of the valve wall (contacting lines	IN^2
D(AMASS)	Mass of the actuator	LB
D(AREA1)	Surface area, piston to volume one	IN^2
D(AREA2)	Surface area, piston to volume two	IN^2
D(ASAC)	Surface area, piston to actuator	IN^2
D(ASAV)	Surface area external to valve	IN^2
ASA1C	Surface area internal to actuator wall volume one	IN^2
ASA2D	Surface area internal to actuator wall volume two	IN^2
ASCP	Contact area, piston and actuator wall volume one	IN^2
ASDP	Contact area, piston and actuator wall volume two	IN^2
D(ASFV)	Internal surface area of the valve	IN^2
ASIGN		
B1,B2,B3, B4,B5,B6,B7	Dummy variable	
CC	Thermal conductivity of the actuator mass	WATTS/IN-°F
CCV	Interface conductance between the valve and actuator	WATTS/IN^2-°F

Variable	Description	Dimension
CJ	Mechanical equivalent of heat	IN-LB_m/WATTS-SEC
CMASS	Actuator mass around volume one	LB_m
CP	Thermal conductivity of the piston	WATTS/IN-°F
CPCN	Specific heat of the actuator walls	WATTS-SEC/LB_m-°F
CPFN	Specific heat of the fluid	WATTS-SEC/LB_m-°F
CPPN	Specific heat of the piston	WATTS-SEC/LB_m °F
CPVN	Specific heat of the valve walls	WATTS-SEC/LB_m-°F
CV	Thermal conductivity of the valve walls	WATTS/IN-°F
D(DAMP)	Static seal friction	LB_f
DCAPT1	Temperature change do to a pressure drop	°F
DCAPT2	Temperature change do to a pressure drop	°F
DDD	Dummy variable	-
DELTA1	Distance from inlet of valve to volume one of actuator	IN.
DELTA2	Distance from outlet of valve to volume two of actuator	IN.
DELTA3	Distance from outlet of valve to volume one of actuator	IN.
DMASS	Actuator mass around volume two	LB.
DXC	Distance from node of actuator volume one to interface	IN.
DXD	Distance from node of actuator volume two to interface	IN.
DXF	Distance from node of fluid in valve to interface	IN.
DXP	Distance from node of piston to its interface	IN.
DXV	Distance from node of valve walls to its interface	IN.
D1,D2	Variable to determine actuator node mass	-

Variable	Description	Dimension
FMASS	Total mass of fluid in valve	LB_m
FMASS1	Mass of fluid in actuator volume one	LB_m
FMASS2	Mass of fluid in actuator volume two	LB_m
DT(FORCE)	Load on actuator	LB_f
ITC	Initial temperature of the actuator valve walls	°F
ITF	Initial temperature of the fluids	°F
KTYPE	Dummy variable	-
D(MAXST)	Maximum stroke	IN.
D(MAXL)	Load at max stroke	LB_f
D(MINL)	Load at min stroke	LB_f
D(MTYPE)	Actuator material type	-
NTYPE	Dummy variable	-
D(PERC)	Percentage heat from pressure drop added to fluid	-
DT(PFORCE)	Actuator pressure drop or rise	PSI
P(PHEIGHT)	Piston wall height	LB_m
D(PMASS)	Piston mass	LB_m
DT(PP1)	Cylinder 1 pressure	PSI
DT(PP2)	Cylinder 2 pressure	PSI
DT(PP1P)	Dummy variable	-
DT(PP2P)	Dummy variable	-
D(PTHICK)	Piston wall thickness	IN.
D(PTYPE)	Piston material type	-
RHOC	Density of the actuator material	$LB_m/IN.^3$
RHOIL	Density of the fluid	$LB_m/IN.^3$
RHOP	Density of the actuator piston	$LB_m/IN.^3$

Variable	Description	Dimension
RHOV	Density of the valve walls	$LB_m/IN.^3$
RMFL1	Mass flow rate into valve	LB_m/SEC
RMFL2	Mass flow rate leaving valve	LB_m/SEC
R1,R2,R3,R4, R9,R12,R15	Dummy variables	-
SHAPF	Shape factor walls to atmosphere	-
SIGMA	Stefan-Botzmann constant for radiation	$WATTS/IN^2-°F^4$
D(SLOTW1)	Slot width volume 1 to con #1	IN.
D(SLOTW2)	Slot width volume 1 to con #2	IN.
D(SLOTW3)	Slot width volume 2 to con #1	IN.
D(SLOTW4)	Slot width volume 2 to con #2	IN.
D(TA)	Temperature of the surrounding atmosphere	°F
DT(TCN)	Temperature of the actuator wall node surrounding volume one	°F
DT(TDN)	Temperature of the actuator wall node surrounding volume two	°F
DT(TFA1)	Temperature of the actuator fluid node in volume one	°F
DT(TFA2)	Temperature of the actuator fluid node in volume two	°F
DT(TPN)	Temperature of the piston node	°F
D(TST)	Temperature of the surrounding structure	°F
DT(TVF1)	Temperature of the fluid node in valve volume one	°F
DT(TVF2)	Temperature of the fluid node in valve volume two	°F
DT(TVN)	Temperature of the valve wall node	°F
UAC	Heat transfer coefficient actuator walls to atmosphere	$WATTS/IN^2-°F$
D(UAV)	Heat transfer coefficient valve walls to atmosphere	$WATTS/IN^2-°F$
UA1C	Heat transfer coefficient actuator fluid to walls, volume one	$WATTS/IN^2-°F$
D(UA1P)	Heat transfer coefficient actuator fluid to piston	$WATTS/IN^2-°F$

Variable	Description	Dimension
UA2D	Heat transfer coefficient actuator fluid 2 to walls, volume two	WATTS/IN^2-°F
UA2P	Heat transfer coefficient actuator fluid 1 to piston	WATTS/IN^2-°F
UFWIL	Heat transfer coefficient actuator fluid in valve to valve	WATTS/IN^2-°F
DT(VEL)	Actuator Velocity	IN/SEC
DT(VELO)	Old actuator velocity	IN/SEC
D(VMASS)	Valve wall mass	LB_m
D(VOLUME2)	Total cylinder volume of actuator	$IN.^3$
DT(VOL1)	Volume 1	$IN.^3$
DT(VOL2)	Volume 2	$IN.^3$
D(VOL3)	Valve volume	$IN.^3$
D(VTYPE)	Material type of the valve	-
DT(X)	Piston position	IN.
XV	Valve position	IN.

6.101.7 Subroutine Listing

```
      SUBROUTINE TACT101 (D,DT,DD,L)
C ***REVISED JUNE  1976 ****
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /STEADY/PN(90),QN(90),PLX(90),PDLEG(90),QL(90),
     + QA,QS,Q1,PUP,PDOWN,NNODE,NLEG,NCPN,TERM,LEGN,ICON,INV,
     + INX,INZ,NUP(90),NDWN(90),NELEM(90),ILEGAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION A(8,8),B(8)
      INTEGER AREA1,AREA2,VOL1,VOL2,SLOTW1,SLOTW2
     1,SLOTW3,SLOTW4,DAMP,X,PFORCE,FORCE,PP1P,PP2P
     2,PP1,PP2,AMASS,VEL,VELO,ASAV,ASAC,UAV,TA,TST
     4,PTHICK,PHEIGHT,CCV,ACCV,SARLA,VOLUME2,VMASS
     5,TFA1,TFA2,TVN,TCN,TDN,TPN,ASFV,VOL3,DELTA1,DELTA2,DELTA3
     6 ,PTYPE,VTYPE,TVF1,TVF2,UA1P
C     D ARRAY VARIABLES
      DATA AREA1/6/,AREA2/7/,VMASS/5/,MINST/28/,MAXST/29/,
     1 DAMP/32/,SARLA/18/,SLOTW1/33/,SLOTW2/34/,SLOTW3/35/,SLOTW4/36/,
     2 MINL/30/,MAXL/31/,INPOS/27/,MTYPE/1/,VTYPE/3/,PTYPE/2/,
     3 AMASS/4/,DELTA1/9/,DELTA2/10/,
     4 PTHICK/12/,PHEIGHT/13/,DELTA3/11/,ASAC/14/,
     5 ACCV/15/,ASAV/16/,ASFV/17/,UAV/21/,CCV/19/,PERC/26/,
     6 TST/22/,TA/23/,ITF/24/,ITC/25/,UA1P/20/,VOL3/8/,VOLUME2/37/
C     L ARRAY VARIABLES
      DATA NTAB/3/,IY/4/,NODE/5/
C     DT ARRAY VARIABLES
      DATA X/1/,VEL/2/,LOADZ/3/,LOADS/4/,PP1/5/,PP2/6/,
     1 VOL1/7/,VOL2/8/,VELO/9/,NCAV/10/,PP1P/11/,PP2P/12/,
     2 PFORCE/13/,FORCE/14/,LOADEX/15/
     3  TFA1/16/,TFA2/17/,TVN/18/,TCN/19/,TDN/20/,TPN/21/,TVF1/22/,
     4 TVF2/23/
      DATA SIGMA/.349E-11/,SHAPF/.96/,EPSION/0.3/,CJ/8.85/
C     MTYPE  IS THE MATERIAL TYPE OF THE ACTUATOR
C     VOL3  IS THE VOLUME IN THE VALVE
C     D(SARLA) =TOTAL SURFACE AREA OF BOTH FLUIDS TO ACTUATOR(VOL1+VOL2)
C     D(RHOV) IS THE COMBINED DENSITY OF THE CASE AND THE VALVE
C     D(ASAC) IS THE SURFACE AREA BETWEEN THE ACTUATOR & THE AMBIENT
C     D(ASAV) IS THE SURFACE AREA BETWEEN THE VALVE & THE AMBIENT
C     D(ACCV)  IS THE AREA OF CONTACT BETWEEN C+D & B
C     D(ASAC) IS THE SURFACE AREA OF ACTUATOR
C     D(ASAV) IS THE SURFACE AREA OF THE VALVE
C     D(VTYPE)  =VALVE MATERIAL TYPE
C     D(DELTA1)  =DISTANCE FROM CONNECTION 1 TO VOV1
C     D(DELTA2) =DISTANCE FROM CONNECTION 2 TO VOL2
C     D(DELTA3) =DISTANCE FROM CONNECTION 1 TO VOL2 OR
C         FROM CONNECTION 2 TO VOL1(AVERAGE OF BOTH OF THESE)
      IF(IENTR) 1000,2000,3000
C *** 1000 SECTION
```

6.101.7 (Continued)

```
 1000 CONTINUE
      D(VOL3)=D(VOL3)/4.0
      D(ASFV)=D(ASFV)/4.0
      D(ASAC)=D(ASAC)/2.0
      DT(TVN)=D(ITC)
      DT(TCN)=D(ITC)
      DT(TDN)=D(ITC)
      DT(TFA1)=D(ITF)
      DT(TFA2)=D(ITF)
      DT(TPN)=D(ITF)
      DT(TVF1)=D(ITF)
      DT(TVF2)=D(ITF)
      L1=L(1)
      L2=L(2)
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
      TC(L1)=D(ITC)
      TC(L2)=D(ITC)
      D(TST)=(D(TST)+460.)**4
      IF(D(UA1P).EQ.0.0) D(UA1P)=.09
      IF(D(UAV).EQ.0.0) D(UAV)=.0069
      IF(D(UAC).EQ.0.0)D(UAC)=.0069
C     ACTUATOR PARAMETER INPUT
      D(SLOTW1)=D(SLOTW1)*0.65
      D(SLOTW2)=D(SLOTW2)*0.65
      D(SLOTW3)=D(SLOTW3)*0.65
      D(SLOTW4)=D(SLOTW4)*0.65
      DT(LOADS)=(D(MAXL)-D(MINL))/(D(MAXST)-D(MINST))
      DT(LOADZ)=D(MAXL)-DT(LOADS)*D(MAXST)
      DT(FORCE)=DT(LOADZ)+DT(LOADS)*D(INPOS)
      DT(LOADEX)=ATPRES*(D(AREA1)-D(AREA2))
      DT(VELO)=0.0
      DT(VEL)=0.0
      DT(X)=D(INPOS)
      L(IY)=(L(NTAB)+7)/8
      L(IY)=41+L(IY)*8
      L(NODE)=1
      XV=D(L(IY))
      DT(NCAV)=1.0
      DT(VOL1)=D(AREA1)*D(INPOS)
      DT(VOL2)=(D(MAXST)-D(INPOS))*D(AREA2)
      DT(PFORCE)=DT(FORCE)/D(AREA2)
      IF(XV) 60,70,80
   60 DT(PP1P)=1/((D(SLOTW2)*XV)**2*2)
      DT(PP2P)=1/((D(SLOTW3)*XV)**2*2)
      L(NODE)=2
      DT(PFORCE)=DT(FORCE)/D(AREA1)
      GO TO 90
   70 DT(NCAV)=0.0
      GO TO 90
```

6.101.7 (Continued)

```
   80 DT(PP1P)=1/((D(SLOTW1)*XV)**2*2)
      DT(PP2P)=1/((D(SLOTW4)*XV)**2*2)
   90 RETURN
C *** 2000 SECTION
C     THE STEADY STATE SECTION
 2000 CONTINUE
      IF(ICON.EQ.2) GO TO 2750
      IF(ICON.NE.1) GO TO 2900
      IF(L(NODE).EQ.2) QS=-QS
      N=NDWN(INEL)
      IF(DT(NCAV).EQ.0.0)GO TO 2550
      IF(DT(X).GT.D(MINST)) GO TO 2600
      IF(QS.GT.0.0) GO TO 2650
 2550 QN(N)=0.0
      DT(PP1P)=Q1*10E6
      DT(PP2P)=Q1*10E6
      DT(VEL)=0.0
      IF(L(NODE).EQ.1) GO TO 2700
      QS=-QS
      GO TO 2850
 2600 IF(DT(X).LT.D(MAXST)) GO TO 2650
      IF(QS.GT.0.0) GO TO 2550
 2650 IF(L(NODE).EQ.2) GO TO 2800
      QN(N)=-DT(NCAV)*Q1*(D(AREA1)-D(AREA2))/D(AREA1)
      DT(VELO)=DT(VEL)
      DT(VEL)=Q1/D(AREA1)
 2700 DT(PFORCE)=(DT(FORCE)-PN(N)*(D(AREA1)-D(AREA2)))/D(AREA2)
     $ +QS*D(DAMP)/D(AREA2)
      DT(PP1)=PN(N)
      DT(PP2)=PN(N)-DT(PFORCE)
      PUP=PUP-DT(PP1P)*RHO(TF(L(ICON)),PUP)*QA*QA*QS
      RETURN
 2750 IF(DT(X).LE.D(MINST).OR.DT(X).GE.D(MAXST))GO TO 2900
      PUP=PUP-DT(PP2P)*RHO(TF(L(ICON)),PUP)*QA*QA*QS
      PDLEG(INEL)=-DT(PFORCE)
      PUP=PUP+PDLEG(INEL)
      RETURN
 2800 QS=-QS
      QN(N)=-DT(NCAV)*Q1*(D(AREA1)-D(AREA2))/D(AREA2)
      DT(VELO)=DT(VEL)
      DT(VEL)=-Q1/D(AREA2)
 2850 DT(PFORCE)=(-PN(N)*(D(AREA2)-D(AREA1))+DT(FORCE))/D(AREA1)
     $ -QS*D(DAMP)/D(AREA1)
      DT(PP2)=PN(N)
      DT(PP1)=PN(N)-DT(PFORCE)
      PUP=PUP-DT(PP1P)*RHO(TF(L(ICON)),PUP)*QA*QA*QS
      RETURN
 2900 WRITE(6,1950) IND,ICON,INEL
 1950 FORMAT(5X,7HCOMP NO,I3,20H, HAS INVALID CON NO  ,I3,
     1 11H, IN LEG NO ,I4)
```

6.101.7 (Continued)

```
      STOP 2101
 2960 PUP=Q1*10E6
      TERM=PDOWN
      DT(PP1)=PN(N)
      DT(PP2)=PN(N)
      WRITE(6,2999)
 2999 FORMAT(10X,28HTHE ACTUATOR IS BOTTOMED OU
      RETURN
C *** 3000 SECTION
 3000 CONTINUE
C     INITALIZING TEMPERATURES
      ITYPE=D(VTYPE)+.001
      NTYPE=D(PTYPE)+.001
      KTYPE=D(MTYPE)+.001
      CP=PROP(NTYPE,3)
      CW=PROP(ITYPE,3)
      CC=PROP(KTYPE,3)
      CV=PROP(KTYPE,3)
      RHOV=PROP(KTYPE,2)
      RHOC=PROP(KTYPE,2)
      RHOP=PROP(NTYPE,2)
      CPVN=PROP(ITYPE,1)
      CPPN=PROP(NTYPE,1)
      CPCN=PROP(KTYPE,1)
      PMASS=D(PTHICK)*D(AREA1)*RHOP
      DT(FORCE)=DT(LOADZ)+DT(LOADS)*DT(X)
      L(NODE)=1
      DT(NCAV)=1.0
      DT(PFORCE)=DT(FORCE)/D(AREA2)
      CALL INTERP(TIME,D(41),D(L(IY)),10,L(NTAB),XV,IERR)
      IF(ABS(XV).LE.0.001)XV=0.0
      IF(XV) 3060,3070,3080
 3060 DT(PP1P)=1/((D(SLOTW2)*XV)**2*2)
      DT(PP2P)=1/((D(SLOTW3)*XV)**2*2)
      L(NODE)=2
      DT(PFORCE)=DT(FORCE)/D(AREA1)
      GO TO 3090
 3070 DT(NCAV)=0.0
      DT(VLL)=0.0
      GO TO 3090
 3080 DT(PP1P)=1/((D(SLOTW1)*XV)**2*2)
      DT(PP2P)=1/((D(SLOTW4)*XV)**2*2)
 3090 XO=DT(X)
      DT(X)=DT(X)+(DT(VEL)+DT(VELO))*DELT/2.
      CALL XLIMIT(DT(X),DT(VEL),ASIGN,D(MINST),D(MAXST))
      IF(DT(VEL).EQ.0.0)GO TO 3099
      DT(VOL1)=DT(VOL1)+(DT(X)-XO)*D(AREA1)
      DT(VOL2)=DT(VOL2)-(DT(X)-XO)*D(AREA2)
 3099 CONTINUE
      L2=L(2)
```

6.101.7 (Continued)

```
      L1=L(1)
      DO 3001 I=1,8
      DO 3001 J=1,8
      A(I,J)=0.0
 3001 B(I)=0.0
      RHOIL=386.4*RHO(TF(L1),P(L1))
      D1=DT(VOL1)/(DT(VOL2)+DT(VOL1))
      D2=D1*DT(VOL2)/DT(VOL1)
      DXP=D(PHEIGHT)/4.0
      DXV=D(DELTA1)/2.
      DXC=DT(VOL1)/(2.*D(AREA1))
      DXD=DT(VOL2)/(2.*D(AREA1))
C      INITIALIZING COMMON FACTORS
      FMASS=D(VOL3)*RHOIL/2.
      CMASS=D(AMASS)*D1
      DMASS=D(AMASS)*D2
      FMASS1=DT(VOL1)*RHOIL
      FMASS2=DT(VOL2)*RHOIL
      ASCP=D(PTHICK)*(D(AREA1)/D(PHEIGHT))
      ASDP=ASCP
      ASA1C=D(SAREA)*D1
      ASA2D=D(SAREA)*D2
      ACA1=DT(VOL1)/D(PHEIGHT)
      ACA2=DT(VOL2)/D(PHEIGHT)
C     ACC AND ACD ARE JUST ESTIMATES OF CROSS SECTIONAL AREAS
      ACC=DT(VOL1)/D(PHEIGHT)
      ACD=DT(VOL2)/D(PHEIGHT)
      ACV=D(VMASS)/(RHOV*D(DELTA1))
      ACFV=D(VOL3)/(2.*D(DELTA1))
      AAA=D(VOL3)/(D(DELTA1)*2.)
      DDD=SQRT(AAA*4./PI)
C     ESTIMATES OF HEAT TRANSFER COEFFICIENTS
      UFWIL=UFW(AAA,DDD,Q(L1),TF(L1),P(L1))
      UA2P=D(UA1P)
      UA1C=D(UA1P)
      UA2D=D(UA1P)
      UAC=D(UAV)
      A2=D(UAV)*D(ASAV)
      RMFL1=ABS(Q(L1))*RHOIL
      CIP1=SIGMA*SHAPF*EPSION*D(ASAV)
      CIP2=SIGMA*EPSION*SHAPF*D(ASAC)
      B1=UAC*D(ASAC)*D1
      B2=B1*D2/D1
      B3=UA1C*ASA1C
      B4=UA2D*ASA2D
      B5=D(UA1P)*D(AREA1)
      B6=UA2P*D(AREA2)
      B7=UFWIL*D(ASFV)/2.
      R1=1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXV/(ACV*CV))
      R2=1.0/(DXF(L2)/(ACW(L2)*C(L2))+DXV/(ACV*CV))
```

6.101.7 (Continued)

```
      R3=1.0/(DXV/(ACV*CV)+DXC/(ACC*CC)+1./(D(ACCV)*D1*D(CCV)))
      R4=1.0/(DXV/(ACV*CV)+DXD/(ACD*CC)+1./(D(ACCV)*D2*D(CCV)))
      R5=CC/(DXD/ACD+DXC/ACC)
      R9=1.0/(DXP/(ASCP*CP)+DXC/(ACC*CC))
      R12=1.0/(DXP/(ASCP*CP)+DXD/(ACD*CC))
      R15=CF/(DXF(L1)/ACF(L1)+DXV/ACFV+RMFL1*DELT
     + /(ACFV**2*RHOIL))
      IF(XV.EQ.0.0) RMFL1=0.0
      RMFL2=ABS(Q(L2))*RHOIL
      IF(XV.EQ.0.0) RMFL2=0.0
      RHOIL=386.4*RHO(DT(TVF1),P(L1))
      IF(XV.GT.0.0) GO TO 3030
      DCAPT1=0.0
      DCAPT2=0.0
      IF(XV.EQ.0.0) GO TO 3066
      DCAPT1=(1./RHOIL)*(P(L1)-DT(PP2))/(CJ*CPFN*2.)
      DCAPT2=(1./RHOIL)*(DT(PP1)-P(L2))/(CJ*CPFN*2.)
      A(5,5)=RMFL1*CPFN
      A(5,1)=-RMFL1*CPFN
      A(2,4)=-RMFL2*CPFN
      B(5)=RMFL1*CPFN*DCAPT1
      GO TO 3066
 3030 DCAPT1=1./RHOIL*(P(L1)-DT(PP1))/(CJ*CPFN*2.)
      DCAPT2=1./RHOIL*(DT(PP2)-P(L2))/(CJ*CPFN*2.)
      A(2,5)=-RMFL2*CPFN
      A(4,4)=RMFL1*CPFN
      A(4,1)=-RMFL1*CPFN
      B(4)=RMFL1*CPFN*DCAPT1
 3066 CONTINUE
      A(1,1)=FMASS*CPFN/DELT+B7+R15+RMFL1*CPFN
      B(1)=FMASS*CPFN*DT(TVF1)/DELT+RMFL1*CPFN*(TF(L1)+DCAPT1)+
     + R15*TF(L1)
      A(1,3)=-B7
      A(2,2)=FMASS*CPFN/DELT+B7+RMFL2*CPFN
      B(2)=FMASS*CPFN*DT(TVF2)/DELT+RMFL2*CPFN*DCAPT2*2.
      A(2,3)=-B7
      A(3,1)=-B7
      A(3,2)=-B7
      A(3,3)=D(VMASS)*CPVN/DELT+R1+R2+A2+2.*B7
      B(3)=D(VMASS)*CPVN*DT(TVN)/DELT+R1*TW(L1)+R2*TW(L2)*A2*D(TA)
     + +CIP1*D(TST)-CIP1*(DT(TVN)+460.)**4
      A(3,6)=-R3
      A(3,7)=-R4
       A(4,4)=FMASS1*CPFN/DELT+A(4,4)+B3+B5
      A(4,6)=-B3
      A(4,8)=-B5
      B(4)=FMASS 1*CPFN*DT(TFA1)/DELT+B(4)
      A(5,5)=FMASS2*CPFN/DELT+B4+B6+A(5,5)
      A(5,7)=-B4
      A(5,8)=-B6
```

6.101.7 (Continued)

```
B(5)=FMASS2*CPFN*DT(TFA2)/DELT+B(5)
A(6,3)=-R3
A(6,4)=-B3
A(6,6)=CMASS*CPCN/DELT+R3+B3+R5+B1+R9
A(6,7)=-R5
A(6,8)=-R9
B(6)=CMASS*CPCN*DT(TCN)/DELT+B1*D(TA)+CIP2*D1*D(TST)
+ -CIP2*D1*(DT(TCN)+460.)**4
A(7,3)=-R4
A(7,5)=-B4
A(7,6)=-R5
A(7,7)=DMASS*CPCN/DELT+R5+B4+R4+R12+B2
A(7,8)=-R12
B(7)=DMASS*CPCN*DT(TDN)/DELT+B2*D(TA)+CIP2*D2*D(TST)
+ -CIP2*D2*(DT(TDN)+460.)**4
A(8,4)=-B5
A(8,5)=-B6
A(8,6)=-R9
A(8,7)=-R12
A(8,8)=PMASS*CPPN/DELT+R12+R9+B5+B6
B(8)=PMASS*CPPN*DT(TPN)/DELT
CALL SIMULT(A,B,8,IERROR)
TF(L2)=B(2)
DT(TVF1)=B(1)
DT(TVF2)=B(2)
DT(TVN)=B(3)
DT(TCN)=B(6)
DT(TDN)=B(7)
DT(TPN)=B(8)
DT(TFA1)=B(4)
DT(TFA2)=B(5)
TC(L1)=B(3)
TC(L2)=B(3)
RETURN
END
```

6.102 SUBROUTINE ACT102

ACT102 simulates a basic utility actuator shown in Figure 6.102-1. The subroutine allows the input of piston rod loads at zero and maximum stroke. Straight line interpolation is used between these two loads.

The subroutine calculates the actuator and piston wall temperatures, and actuator fluid temperatures in Volume 1 and Volume 2.

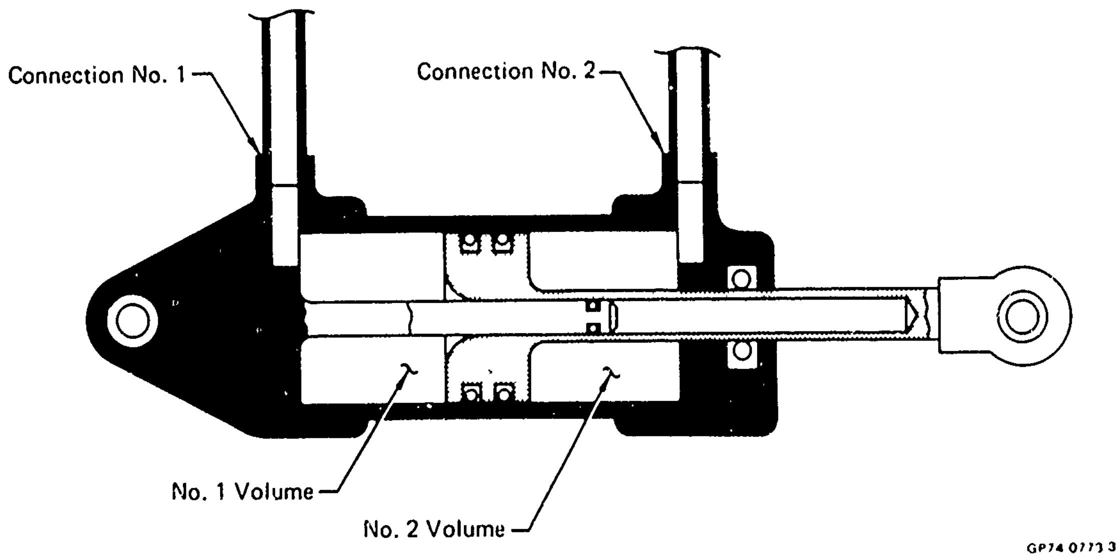


FIGURE 6.102-1
TYPE NO. 102 UTILITY ACTUATOR

6.102.1 Math Model

The thermal math model for the actuator includes heat transfer to and from two connecting line segments, one upstream and one downstream. Nine nodes are considered: four fluid nodes, four wall nodes, and one piston node (as shown in Figure 6.102-2).

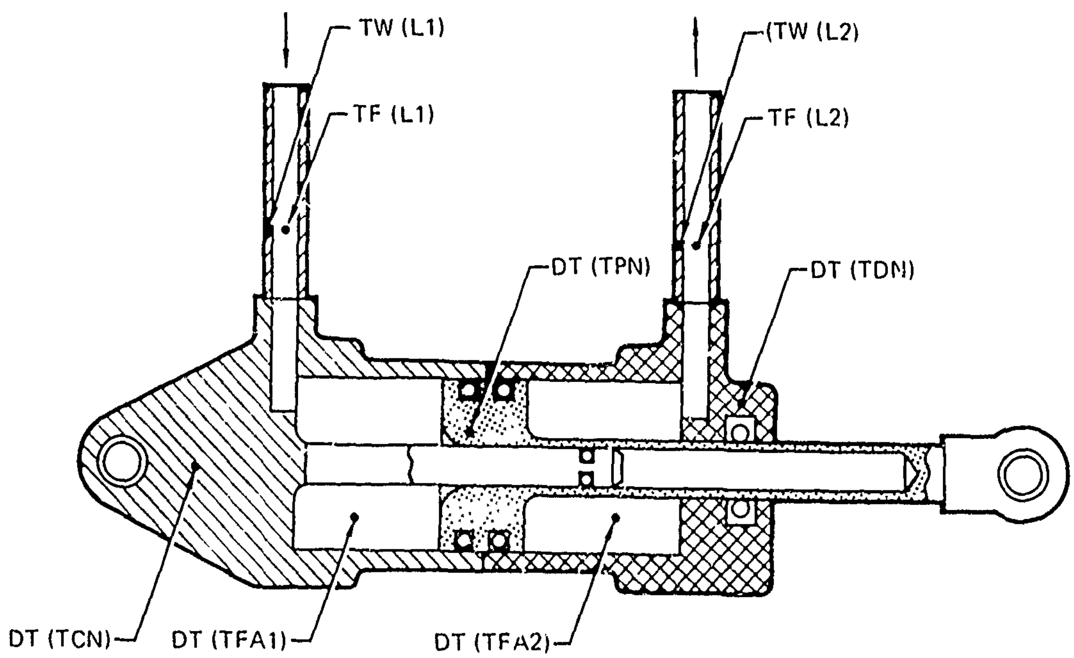


FIGURE 6.102-2
ACTUATOR AND CONNECTOR NODE REPRESENTATION

GP77-0065-6

The temperatures of the upstream line segment, wall and fluid nodes are denoted by TW(L1), TF(L1), the temperatures of the actuator wall and fluid nodes are DT(TCN), DT(TDN), DT(TFA1), DT(TFA2), and the temperature of the piston node is denoted by DT(TPN). The downstream line segment wall

and fluid node temperatures are TW(L2) and TF(L2). This identifies piston travel from left to right. Note that the piston can also travel from right to left.
Then L1 and L2 would be reversed and the discussion to follow would require reversing nomenclature.

Five heat balance equations are written to solve for the five actuator temperatures just stated, using the actuator and line material properties and dimensions, the atmospheric and structure temperatures external to the actuator, and the temperatures of the line segment nodes, TW(L1), TF(L1), and TW(L2). (Note: TF(L2) = DT(TFA2), see assumptions).

The first equation represents four modes of heat transfer relative to the actuator fluid node in volume one (entrance volume).

1. Heat transfer due to mass transfer into the actuator from the upstream line segment

$\dot{m}$Cp*(TF(L1) - DT(TFA1))

where $\dot{m}$Cp is the volume flow rate coefficient and is equal to

Q(L1) * RHOIL * CPFN.

2. Conduction to and from the upstream line segment fluid node

R1 * (TF(L1) - DT(TFA1))

where R1 is the conduction coefficient and is equal to CF/(DXF(L1)/ ACF(L1) + DXA1/ACA1 + ABS(Q(L1) * RHOIL*DELT/(ACF(L2)**2*RHOIL).

3a. Convection to and from the actuator wall node

B3 * (DT(TCN) - DT(TFA1))

where B3 is a convection coefficient equal to UA1C * ASA1C.

3b. Convection to and from the piston node

B5 * (DT(TPN) - DT(TFA1))

where B5 is the convection coefficient and is equal to

D(UA1P) * ASA1P

4. Heat addition due to a pressure drop experienced by the fluid as it flows thru the actuator orifice. If the flow is into volume one, the heat added is

A1 * DCAPT1

where A1 is equal to Q(L1)*RHOIL*CPFN and DCAPT1 is equal to 1.0/RHOIL*(P(L1)-DT(P1))/(CJ*CPFN).

These heat transfer terms are then combined to produce the equation for the heat balance for the actuator volume one fluid node.

$$\frac{MCp}{DELT} * (DT(TFA1) - DT(TFA1)_{OLD}) = \dot{M}Cp * (TF(L1)-DT(TFA1)) + R1*$$
$$(TF(L1)-DT(TFA1))+B3*(DT(TCN)-DT(TFA1) + B5*(DT(TPN)-DT(TFA1))$$
$$+ A1*DCAPT1$$

The second equation represents two modes of heat transfer relative to the actuator fluid in volume two.

1a. Convection to and from the actuator wall node

B4 * (DT(TDN) - DT(TFA2))

where B4 is the convection coefficient for the fluid equal to UA2D * ASA2D.

1b. Convection to and from the piston node

B6 * (DT(TPN) - DT(TFA2))

where again, B6 is a convection coefficient and equal to UA2P * ASA2P.

2. Heat added directly to the fluid due to a pressure drop across an actuator orifice, (the exit).

A2 * DCAPT2

where DCAPT2 is equal to 1.0/RHOIL*(DT(P2)-P(L2))/(CJ*CPFN) and A2 is equal to RMF (L1)*CPFN.

These terms are then combined to produce the equation for the heat balance for the actuator volume two fluid node.

$$\frac{MCp}{DELT} * (DT(TFA2) - DT(TFA2)_{OLD}) = B4 * (DT(TDN) - DT(TFA2)) + B5 * (DT(TPN) - DT(TFA2)) + A2 * DCAPT2 \qquad (2)$$

The third equation represents three modes of heat transfer relative to the actuator wall surrounding volume one.

1a. Conduction to and from the upstream connecting line wall node

R2 * (TW(L1) - DT(TCN))

where R2 is the conduction coefficient for the walls equal to 1.0/(DXF(L1)/(ACW(L1)*C(L1))+DXC/(ACC*CC))

1b. Conduction to and from the actuator wall node surrounding volume two.

R8 * (DT(TDN) - DT(TCN))

where R8 is the conduction coefficient equal to CC/(DXD/ACD + DXC/ACC).

1c. Conduction to and from the piston node.

R9 * (DT(TPN) - DT(TCN))

where R9 is the conduction coefficient between the two nodes equal to 1.0/(DXP/(CP*ACP) + DXC/(ACC * CC)), where the interface conductance between the two nodes is infinite.

2a. Convection to and from the actuator fluid node in volume one.

B3 * (DT(TFA1) - DT(TCN))

and B3 is as defined previously.

2b. Convection to and from the external atmosphere

B1 * (D(TA) - DT(TCN))

where B1 is a convection coefficient for the actuator wall equal to D(UAC) * D(ASAC) * DVOL1. DVOL1 is a coefficient to calculate the wall mass around volume one equal to DT(VOLUME1)/(DT(VOLUME1) + DT(VOLUME2)).

3. Radiation exchange with the surrounding structure

$D3 * (D(TST) - (DT(TCN) + 460.)^4)$

where D3 is a radiation coefficient equal to SIGMA * EPSION * SHAPF * D(ASAC) * DVOL1

These terms combine to produce the equation for the heat balance for the actuator wall node surrounding volume one.

$$\frac{MCp}{DELT} * (DT(TCN) - DT(TCN)_{OLD}) = R2*(TW(L1)-DT(TCN))+R8*(DT(TDN) - DT(TCN))+R9*(DT(TPN)-DT(TCN))+B3*(D(TA)-DT(TCN))+D3*D(TST)-D3*(DT(TCN)+460.)**4 \quad (3)$$

where MCp is equal to CMASS * CPCN and CMASS is equal to D(AMASS) * DVOL1.

The fourth equation represents three modes of heat transfer the actuator wall surrounding volume two.

1a. Conduction to and from the downstream line segment wall node.

R3 * (TW(L2) - DT(TDN))

where R3 is a conduction coefficient equal to 1.0/(DXF(L2)/(C(L2)*ACW(L2)) + DXD/(ACD * CD)).

1b. Conduction to and from the actuator wall node surrounding volume one.

R8 * (DT(TCN) - DT(TDN))

with R8 defined previously.

1c. Conduction to and from the piston node.

R12 * (DT(TDN) - DT(TDN))

where R12 is a conduction coefficient equal to 1.0/(DXP/(ACP*CP) + DXD/(ACD * CD)) with the interface conductance being infinite.

2a. Convection with the actuator fluid node in volume two.

B4 * (DT(TFA2) - DT(TDN))

where B4 is the convection coefficient between the two nodes equal to UA2D * ASA2D.

2b. Convection to and from the external atmosphere

B2 * (D(TA) - DT(TDN))

where B2 is the convection coefficient equal to B1 * DVOL2/DVOL1 with the terms defined previously and DVOL2 is a coefficient to calculate the wall mass around volume two, equal to DT(VOLUME2)/ (DT(VOLUME1) + DT(VOLUME2)).

3. Radiation exchange with the surrounding structure.

$$D4 *(D(TST) - (DT(TDN) + 460.)^{4})$$

where D4 is a radiation coefficient equal to D3*DVOL2/DVOL1, with these terms defined previously.

These terms then combine to produce the equation for the heat balance for the actuator wall node surrounding volume 2.

$$\frac{MCp}{DELT}(DT(TDN)-DT(TDN)_{OLD})=R3*(TW(L2)-DT(TDN))+R8*(DT(TCN)-DT(TDN)) +B4*(DT(TFA2)-DT(TDN))+B2*(D(TA) -DT(TDN))+D4*D(TST)-D4*(DT(TDN)+460.) **4 \qquad (4)$$

where MCp is equal to DMASS*CPDN and DMASS is equal to D(AMASS)* DVOL2.

The fifth equation represents two modes of heat transfer to and from the piston node.

1. Conduction to and from the two actuator wall nodes, one and two respectively.

   a. R9 * (DT(TCN) - DT(TPN)) and

   b. R12 *(DT(TDN) - DT(TPN))

   where R9 and R12 are the same as defined previously.

2. Convection to and from the two actuator fluid nodes, one and two respectively.

   a. B5 * (DT(TFA1) - DT(TDN)) and

   b. B6 * (DT(TFA2) - DT(TDN))

   and B5 and B6 are the same as defined previously.

These terms then combine to produce the equation for the heat balance for the actuator piston node

$$\frac{MCp}{DELT} * (DT(TDN)-DT(TDN)_{OLD}) = R9 * (DT(TCN)-DT(TDN)+R12* (DT(TDN)-DT(TPN))+B5*(DT(TFA1) - DT(TDN))+B6*(DT(TFA2)-DT(TDN)) \qquad (5)$$

where MCp is equal to PMASS * CPPN.

A thermal model of the above heat transfer terms for the Actuator is shown in Figure 6.102-3. Equations (1) thru (5) are solved for the appropriate temperatures.

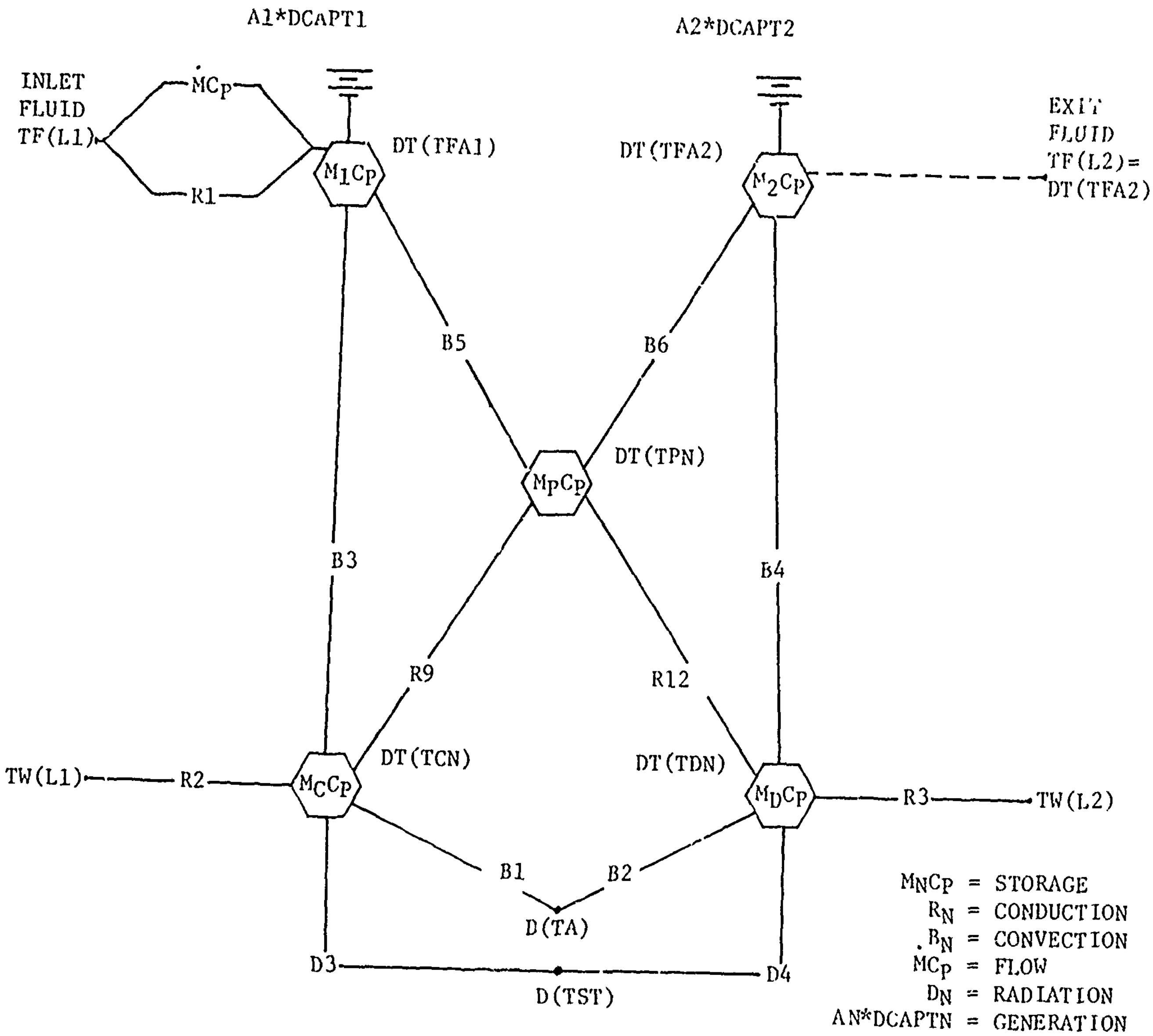


FIGURE 6.102-3

THERMAL MODEL

In this hydraulic math model, the external force due to atmospheric pressure is

ATRESS*(D(AREA1)-D(AREA2))

This value is subsequently used as part of the actuator load for any piston position.

The total actuator load

DT(LOADEX) = DT(LOADZ) + DT(LOADS)*DT(X) + (D(AREA1) - D(AREA2))
*ATPRESS

A sign convention is established such that flow into the Volume 1 chamber and resulting piston velocity are positive.

A system node, N, is established in the Volume 1 end of the actuator.

The simple unbalanced actuator represents a flow and pressure discontinuity. In this case, the flow out of the actuator is proportional to the flow in, but does not equal it. The pseudo overboard flow is the difference between the inlet and outlet flows. This flow is added or subtracted to the flow at the pressure node depending on the direction of motion of the piston. The associated pressure gradient is applied to the leg connected downstream of the actuator.

Overboard flow at the actuator node is calculated using the piston area ratio times the inlet actuator flow

QN(N) = (-1.)*QA*QS*(D(AREA1)-D(AREA2))/D(AREA1)

The pressure gain or loss across the piston is calculated using the force balance equation

DT(DELTP) = LS*(PN(N)*(D(AREA1) - D(AREA2)) - DT(LOADEX) - QS*
D(DAMP))/D(AREA2)

6.102.2 Assumptions

1. The temperature of the fluid leaving the actuator is equal to the fluid node temperature calculated, DT(TFA2).
2. Each entire actuator wall is at the same temperature.
3. The interface conductance between the actuator walls and the line walls is infinite.
4. The emissivity of the wall materials is a constant.
5. The atmospheric and structure temperatures remain constant.
6. Complete fluid mixing occur in the fluid volume.

6.102.3 Computational Methods

SECTION 1000

The fluid and wall temperatures are initialized, the external structure temperature is changed from degrees Farenheit to Rankine and raised to the fourth power, and the default values are assigned.

SECTION 2000 - The entry first determines whether connection No. 1 is attached to an upstream or downstream line. This establishes the actuator steady state mode of operation. If entry is made using connection No. 2, leg pressure gain (or loss) and leg laminar constant are updated. Pressure at connection No. 2 is also calculated and stored. If entry is made using connection No. 1, tests are performed to verify that the piston is free to move as prescribed by the flow guess.

If the piston is on a stop and the flow guess is such that motion would be into the stop, the node overboard flow is set to zero and DT(DELTP) is set to a very large number.

If the piston is free to move, the overboard flow, piston velocity, ΔP across piston and pressure at connection No. 2 are calculated.

SECTION 3000

The position of the actuator piston is computed via a simple integration

DT(X) = DT(X) + (DT(VEL) + DT(VELO)) * DELT/2.

The cylinder volumes are easily calculated as

DT(VOL1) = DT(VOL1) + (DT(X)-XO)*D(AREA1)

DT(VOL2) = DT(VOL2)-(DT(X)-XO)*D(AREA2)

Property values are assigned. Dimensions and coefficients are calculated. The flow direction is determined. (The program is set up with the flow entering connection line one (L1) and leaving thru connection line two (L2). During the calculation the flow direction is checked. If the flow has reversed flow direction, the program reassigns connection numbers so that the flow still enters connection line one). Some coefficients are then recalculated if the flow is reassigned. A 5x5 matrix is loaded and the mathematical equations are solved for DT(TFA1), DT(TFA2), DT(TCN), DT(TDN) and DT(TPN) and stored in the B computational array. The calculated values are assigned to their proper storage locations and the boundary conditions are assigned to arrays (TC and TF) in common /TRANS/.

TC(L1) = B(3)

6.102.4 Approximations

(a) Emissivity of the actuator is .3, which is the emissivity of steel.

(b) Areas and distances are approximated.

6.102.5 Limitations

Not applicable.

6.102.6 Variable Listing

Variables	Description	Dimensions
A( )	Computational array	--
ACA1	Cross sectional area of fluid 1	$IN.^2$
ACA2	Cross sectional area of fluid 2	$IN.^2$
ACC	Cross sectional area of case 1	$IN.^2$
ACD	Cross sectional area of case 2	$IN.^2$
ACP	Cross sectional area of the piston	$IN.^2$
D(AMASS)	Actuator mass	LB_m
D(AREA1)	Volume 1 piston area	$IN.^2$
D(AREA2)	Volume 2 piston area	$IN.^2$
D(ASAC)	Surface area external to actuator	$IN.^2$
ASA1C	Surface area fluid 1 to case 1	$IN.^2$
ASA1P	Surface area fluid 1 to piston	$IN.^2$
ASA2D	Surface area fluid 2 to case 2	$IN.^2$
ASA2P	Surface area fluid 2 to piston	$IN.^2$
ASCD	Contact area between two actuator wall nodes	$IN.^2$
ASCP	Surface area case, Volume 1 to piston	$IN.^2$
ASDP	Surface area case, Volume 2 to piston	$IN.^2$
D(ASFA)	Total internal surface area, fluid to actuator	$IN.^2$
ASIGN	Dummy variable	--
D(ATHICK)	Actuator wall thickness	IN.
A1	Dummy variable	--
A2	Dummy variable	--
B( )	Computational array	--
BUZ	Dummy variable	--

Variables	Description	Dimensions
B1,B2,B3,B4,B5,B6	Dummy variables	--
CC	Thermal conductivity of the actuator wall surrounding volume one	WATTS/IN-°F
CD	Thermal conductivity of the actuator wall surrounding volume two	WATTS/IN=°F
CJ	Mechanical equivalent of heat	IN-LB_m/WATTS-SEC
CMASS	Node C mass	LB_m
CP	Thermal conductivity of the piston	WATTS/IN.-°F
CPCN	Specific heat of the wall mass around volume one	WATTS-SEC/LB_m-°F
CPDN	Specific heat of the wall mass around volume two	WATTS-SEC/LBm-°F
CPPN	Specific heat of the piston	WATTS-SEC/LB_m-°F
D(DAMP)	Dynamic friction	LB_f
DCAPT1	Temperature change due to pressure drop	°F
DCAPT2	Temperature change due to pressure drop	°F
D(DELTP)	Actuator pressure drop	PSI
D(DIA)	Piston rod diameter	IN.
DMASS	Node D mass, around volume two	LB_m
DVOL1	Dummy variable	--
DVOL2	Dummy variable	--
DXA1	Distance, node to interface, Volume 1 to Line 1	IN.
DXA2	Distance, node to interface, Volume 2 to Line 2	IN.
DXC	Distance, node to interface, node C to interface with D	IN.
DXD	Distance, node to interface, node D to interface with C	IN.
DXP	Distance, node to interface, piston to case	IN.

Variables	Description	Dimensions
D3, D4	Dummy variables	--
EPSION	Emissivity factor of the case	--
FA1M	Fluid in Volume 1, mass	LB_m
FA2M	Fluid in Volume 2, mass	LB_m
IR, IS	Dummy variables	--
D(ITC)	Initial temperature of the case	°F
D(ITF)	Initial temperature of the fluid	°F
KTYPE	Dummy variable	--
DT(LOADS)	Load/stroke slope	LB/IN
DT(LOAD7)	External load at piston stroke	LB.
D(MAXL)	Load at full stroke	LB.
D(MAXST)	Maximum actuator stroke	IN.
D(MINST)	Minimum actuator stroke	IN.
D(MINL)	Load at minimum stroke	LB.
D(MTYPE)	Actuator material type	--
NTYPE	Dummy variable	--
D(PHEIGHT)	Piston height	IN.
PMASS	Piston mass	LB_m
D(PTHICK)	Piston thickness	IN.
D(PTYPE)	Piston material type	--
RHOC	Actual metal density	LB_m/IN^3
RHOIL	Fluid density	LB_m/IN^3
RHOP	Piston metal density	LB_m/IN^3
RMS	Fummy variable	--
RQ	Dummy variable	--

Variable	Description	Dimensions
R1,R2,R3,R6,R12	Dummy variables	--
SHAPF	Shape factor case to structure	
SIGMA	Stefan-Boltzmann radiation constant	WATTS/IN^3-°F
D(TA)	Temperature of the surrounding ambient	°F
DT(TCN)	Actuator wall temperature, Volume 1	°F
DT(TDN)	Actuator wall temperature, Volume 2	°F
DT(TFA1)	Fluid temperature, Volume 1	°F
DT(TFA2)	Fluid temperature, Volume 2	°F
DT(TPN)	Piston temperature	°F
D(TST)	Temperature of the surrounding structure	°F
D(UAC)	Heat transfer coefficient external to case C	WATTS/IN^2-°F
UAD	Heat transfer coefficient node D to atmosphere	WATTS/IN^2-°F
UA1C	Heat transfer coefficient fluid 1 to case C	WATTS/IN^2-°F
D(UA1P)	Heat transfer coefficient fluid 1 to piston	WATTS/IN^2-°F
UA2D	Heat transfer coefficient fluid 2 to case D	WATTS/IN^2-°F
UA2P	Heat transfer coefficient fluid 2 to piston	WATTS/IN^2-°F
DT(VEL)	Actuator velocity	IN/SEC
DT(VELO)	Old velocity	IN/SEC
DT(VOLUME1)	Actuator Volume 1	IN^3
DT(VOLUME2)	Actuator Volume 2	IN^3
D(VOL1)	Volume 1 (initially)	IN^3
D(VOL2)	Volume 2 (initially)	IN^3
DT(X)	Actuator position	IN.
XO	Old actuator position	IN.

For variables in common refer to Paragraph 3.3.

6.102.7 Subroutine Listing

```
      SUBROUTINE TACT102 (D,DT,DD,L)
C *** REVISED AUG 1976
      DIMENSION D(1),DT(1),DD(1),L(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INEL
      COMMON /STEADY/PN(90),QN(90),PEX(90),PDLEG(90),QL(90),
     1 QA,QS,Q1,PUP,PDOWN,NNODE,NLEG,NCPN,TERM,LEGN,ICON,INV,
     2 INX,INZ,NUP(90),NDWN(90),NELEM(90),ILEGAD(90),ILEG(1000)
      COMMON /FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      DIMENSION RMF(6),A(5,5),B(5)
      INTEGER VOL1,VOL2,ASAC,UAC,DIA,PHEIGHT,TA,TST,TCN,TDN,
     1 TPN,TFA1,TFA2,ITC,ITF,PTHICK,AMASS,ATHICK,ASFA,VELO,
     2 AREA1,AREA2,P1,P2,MTYPE,PTYPE,UA1P,DAMP,X,
     3 ZQ,VEL,DELTP,VOLUME1,VOLUME2
C     D ARRAY VARIABLES
      DATA VOL1/1/,VOL2/2/,AREA1/3/,AREA2/4/,MTYPE/5/,PTYPE/6/
     1 ,AMASS/7/,ATHICK/8/,ASFA/9/,PHEIGHT/10/,PTHICK/11/,DIA/12/
     2 ,ASAC/13/,UAC/14/,UA1P/15/,TST/16/,TA/17/,ITF/18/,ITC/19/
     3 ,MINST/20/,MAXST/21/,DAMP/22/,MINL/23/,MAXL/24/,INPOS/25/
C     DT ARRAY VARIABLES
      DATA TCN/1/,TDN/2/,TPN/3/,TFA1/4/,TFA2/5/,P1/6/,
     1 P2/7/,X/8/,VEL/9/,LOADZ/10/,LOADS/11/,ZQ/13/
     2 ,LOADEX/12/,INENT/14/,DELTP/15/,VOLUME1/16/,VOLUME2/17/
     3 ,VELO/18/
      DATA SIGMA/.349E-11/,SHAPE/.96/,EPSION/.3/,CJ/8.85/
      IF(IENTR) 1000,2000,3000
C     ATHICK  =ACTUATOR WALL THICKNESS
C     AMASS   =ACTUATOR MASS, INCLUDING PISTON
C     ASFA    =SURFACE AREA FLUID TO ACTUATOR, INSIDE
C     DELCS  =DIMENSIONS OF THE RESTRICTOR
C     PTHICK =THICKNESS OF THE PISTON
C     DIA    =DIAMETER OF THE PISTON
C     UAC    =HEAT TRANSFER COEFF. CASSE TO AMBIENT
C     VEL    =VELOCITY OF THE PISTON
C *** 1000 SECTION
 1000 CONTINUE
      L1=L(1)
      L2=L(2)
C     INITALIZING TEMPERATURES
      TC(L1)=D(ITC)
      TC(L2)=D(ITC)
      TF(L1)=D(ITF)
      TF(L2)=D(ITF)
      DT(TCN)=D(ITC)
      DT(TDN)=D(ITC)
      DT(TPN)=D(ITF)
      DT(TFA1)=D(ITF)
      DT(TFA2)=D(ITF)
      D(TST)=(D(TST)+460.)**4
```

```
      KTYPL=D(MTYPL)+.001
      NTYPL=D(PTYPL)+.001
      CP=PROP(NTYPL,3)
      CC=PROP(KTYPL,3)
      CPPN=PROP(NTYPL,1)
      CPCN=PROP(KTYPL,1)
      RHOP=PROP(NTYPL,2)
      RHOC=PROP(KTYPL,2)
      CD=CC
      CPDN=CPCN
      DT(VEL)=0.0
      IF(D(UA1P).EQ.0.0) D(UA1P)=.0063
      IF (D(UAC).EQ.0.0) D(UAC)=.0053
      D(AMASS)=D(AMASS)-D(AREA1)*D(PTHICK)*RHOP
      D(ASAC)=D(ASAC)/2.0
      L(3)=1
      DT(VOLUME1)=D(VOL1)
      DT(VOLUME2)=D(VOL2)
      IF(L(1)/2.NE.(L(1)+1)/2) L(3)=-1
      L(4)=-1
      IF(L(2)/2.NE.(L(2)+1)/2) L(4)=1
      DT(DELTP)=0.0
      DT(X)=D(INPOS)
      DT(LOADS)=(D(MAXL)-D(MINL))/(D(MAXST)-D(MINST))
      DT(LOADZ)=D(MAXL)-DT(LOADS)*D(MAXST)
      DT(LOADEX)=DT(LOADS)*DT(X)+DT(LOADZ)+(D(AREA1)-D(AREA2))
     + *ATPRES
      RETURN
C *** 2000 SECTION
 2000 CONTINUE
      LS=L(2+ICON)
      N=NDWN(INEL)
      IF(LS.GT.0) GO TO 2510
      IF(INX.NE.1.AND.ICON.EQ.1) GO TO 2900
      N=NUP(INEL)
 2510 IF(ICON.EQ.2) GO TO 2850
      IF(DT(X).GT.D(MINST)) GO TO 2600
      IF(DT(LOADEX).GE.0.0) GO TO 2550
      IF(QS.GT.0.0) GO TO 2650
 2550 QN(N)=0.0
      DT(LOADEX)=0.0
      DT(VEL)=0.0
      DT(DELTP)=Q1*10E6
      GO TO 2700
 2600 IF(DT(X).LT.D(MAXST)) GO TO 2650
      IF(DT(LOADEX).LE.0.0) GO TO 2550
      IF(QS.GT.0.0) GO TO 2550
 2650 CONTINUE
      DT(VELO)=DT(VEL)
      QN(N)=(-1.)*Q1*(D(AREA1)-D(AREA2))/D(AREA1)
```

```
      DT(VEL)=Q1/D(AREA1)
      DT(DELTP)=(PN(N)*(D(AREA1)-D(AREA2))-DT(LOADEX)-
     + QS*D(DAMP))/D(AREA2)
 2700 DT(P1)=PN(N)
      DT(P2)=PN(N)+DT(DELTP)
      RETURN
 2850 IF(INX.EQ.1.AND.LS.EQ.-1) GO TO 2900
      IF(DT(X).GE.D(MAXST).OR.DT(X).LE.D(MINST))GO TO 2950
      PDLEG(INEL)=DT(DELTP)*LS
      PUP=PUP+DT(DELTP)*LS
      RETURN
 2950 PUP=+Q1*10E6
      TERM=PDOWN
      DT(P2)=PN(N)
      RETURN
 2900 STOP
 3000 CONTINUE
      KTYPE=D(MTYPE)+.001
      NTYPE=D(PTYPE)+.001
      CP=PROP(NTYPE,3)
      CC=PROP(KTYPE,3)
      CPPN=PROP(NTYPE,1)
      CPCN=PROP(KTYPE,1)
      RHOP=PROP(NTYPE,2)
      RHOC=PROP(KTYPE,2)
      CD=CC
      CPDN=CPCN
      DT(LOADEX)=DT(LOADS)*DT(X)+DT(LOADZ)+(D(AREA1)-D(AREA2))
     + *ATPRES
      DT(X)=DT(X)+(DT(VLL)+DT(VELO))*DELT/2.
      CALL XLIMIT(DT(X),DT(VEL),ASIGN,D(MINST),D(MAXST))
      IF(DT(VEL).EQ.0.0) GO TO 3001
      DT(VOLUME1)=DT(VOLUME1)+DT(X)*D(AREA1)
      DT(VOLUME2)=DT(VOLUME2)-DT(X)*D(AREA2)
 3001 L2=L(2)
      L1=L(1)
      RHOIL=386.4*RHO(TF(L1),P(L1))
      DXA1=D(PHEIGHT)/1.33
      DXA2=DXA1
      DXP=D(PHEIGHT)/4.0
      ASA2P=D(AREA2)
      ASA1P=D(AREA1)
      ASCP=D(PTHICK)*(D(AREA1)/D(PHEIGHT))
      ASDP=ASCP
C     ACF(L1)=CROSS SECTIONAL AREA BETWEEN FLUID A1&L1
C     ACC&ACD=ESTIMATES OF CROSS SECTIONAL AREAS, ALSO
C     CONTACT AREA BETWEEN C&D
 3003 BUZ=(D(AMASS)/RHOC)/(D(AREA1)/DT(VOLUME1)+D(AREA2)/DT(VOLUME2))
      ACC=BUZ
      ACD=BUZ
```

6.102.7 (Continued)

```
      UA2P=D(UA1P)
      UA1C=D(UA1P)
      UA2D=D(UA1P)
      UAD=D(UAC)
      DVOL1=DT(VOLUME1)/(DT(VOLUME1)+DT(VOLUME2))
      DVOL2=DVOL1*DT(VOLUME2)/DT(VOLUME1)
 3006 CMASS=D(AMASS)*DT(VOLUME1)/(DT(VOLUME1)+DT(VOLUME2))
      DMASS=D(AMASS)*DT(VOLUME2)/(DT(VOLUME1)+DT(VOLUME2))
      PMASS=(D(AREA1)*D(PTHICK)+(D(AREA2)-D(AREA1))*D(DIA)
     + /4.0)*RHOP
      FA1M=DT(VOLUME1)*RHOIL
      FA2M=DT(VOLUME2)*RHOIL
      RD=Q(L1)
      ACP=PMASS/(RHOP*D(PHEIGHT))
      ASA1C=D(ASFA)*DT(VOLUME1)/(DT(VOLUME1)+DT(VOLUME2))
      ASA2D=D(ASFA)*DT(VOLUME2)/(DT(VOLUME1)+DT(VOLUME2))
      ASCD=ACC
      DXC=DT(VOLUME1)/(2.0*D(AREA1))
      DXD=DT(VOLUME2)/(2.0*D(AREA2))
      ACA1=DT(VOLUME1)/D(PHEIGHT)
      ACA2=DT(VOLUME2)/D(PHEIGHT)
C     ESTIMATES OF HEAT TRANSFER COEFF.
      QLEAK=0.0
      RMF(L1)=Q(L1)*RHOIL
      A2=ABS(RMF(L1))*CPFN
      R1=CF/(DXF(L1)/ACF(L1)+DXA1/ACA1+ABS(RMF(L1))*DELT/(ACF(L1)**2
     + *RHOIL))
      IF(Q(L1).LT.0.0) RMF(L1)=0.0
      RMF(L2)=Q(L2)*RHOIL
      R6=CF/((DXF(L2)/ACF(L2))+DXA2/ACA2+ABS(RMF(L2))*DELT
     + /(ACF(L2)**2*RHOIL))
      IF(Q(L2).LT.0.0) RMF(L2)=0.0
      D3=SIGMA*EPSION*SHAPE*D(ASAC)*DVOL1
      D4=D3*DVOL2/DVOL1
      A1=CPFN/TIME
 3033 B1=D(UAC)*D(ASAC)*DVOL1
      B2=B1*DVOL2/DVOL1
      B3=UA1C*ASA1C
      B4=UA2D*ASA2D
      B5=D(UA1P)*ASA1P
      B6=UA2P*ASA2P
      R2=1.0/(DXF(L1)/(C(L1)*ACW(L1))+DXC/(ACC*CC))
      R3=1.0/(DXF(L2)/(C(L2)*ACW(L2))+DXD/(ACD*CD))
      R8=CC/(DXD/ACD+DXC/ACC)
      R9=1.0/(DXP/(CP*ACP)+DXC/(ACC*CC))
      R12=1./(DXP/(ACP*CP)+DXD/(ACD*CD))
      IF(Q(L1).LT.0.0) R1=0.0
      IF(Q(L1).GT.0.0) R6=0.0
C     CALCULATING TEMPERATURE DISTRIBUTION
      IS=L1
```

6.102.7 (Continued)

```
      IR=L2
      DCAPT1=(1./RHOIL)*ABS(P(L1)-DT(P1))/(CJ*CPFN)
      DCAPT2=(1./RHOIL)*ABS(DT(P2)-P(L2))/(CJ*CPFN)
      IF(Q(L1).GE.0.0) GO TO 3100
      L2=L1
      L1=L(2)
      IS=L2
      IR=L1
      DCAPT2=(1./RHOIL)*ABS(P(L1)-DT(P2))/(CJ*CPFN)
      DCAPT1=(1./RHOIL)*ABS(DT(P1)-P(L2))/(CJ*CPFN)
 3100 CONTINUE
C     A1,A2,C,D,P NODES IN ORDER
 3200 A(1,1)=(FA1M*CPFN)/DELT+R1+RMF(IS)*CPFN+B5+B3
      A(1,2)=0.0
      A(1,3)=-B3
      A(1,4)=0.0
      A(1,5)=-B5
      B(1)=FA1M/DELT*DT(TFA1)*CPFN+(R1+RMF(IS)*CPFN)*TF(L1)
     + +A2*DCAPT1
      A(2,1)=0.0
      A(2,2)=(FA2M)*CPFN/DELT+R6+RMF(IR)*CPFN+B6+B4
      A(2,3)=0.0
      A(2,4)=-B4
      A(2,5)=-B6
      B(2)=CPFN*FA2M*DT(TFA2)/DELT+(R6+RMF(IR)*CPFN)*TF(L1)
     + +A2*DCAPT2
      A(3,1)=-B3
      A(3,2)=0.0
      A(3,3)=(CMASS)*(CPCN/DELT)+R9+B3+B1+R8+R2
      A(3,4)=-R8
      A(3,5)=-R9
      B(3)=(CMASS)*DT(TCN)*CPCN/DELT+B1*D(TA)+D3
     + *D(TST)-D3*(DT(TCN)+460.)**4+R2*TW(IS)
      A(4,1)=0.0
      A(4,2)=-B4
      A(4,3)=-R3
      A(4,4)=(DMASS)*(CPDN/DELT)+R8+R12+B4+B2+R3
      A(4,5)=-R12
      B(4)=(DMASS)*(CPDN/DELT)*DT(TDN)+B2*D(TA)+D4*D(TST)
     + -D4*(DT(TDN)+460.)**4+R3*TW(IR)
      A(5,1)=-B5
      A(5,2)=-B6
      A(5,3)=-R9
      A(5,4)=-R12
      A(5,5)=PMASS*CPPN/DELT+R9+B5+R12+B6
      B(5)=PMASS*CPPN*DT(TPN)/DELT
      CALL SIMULT(A,B,5,IERROR)
      IF(RQ.LE.0.0) GO TO 3250
      TC(L1)=B(3)
      TC(L2)=B(4)
```

6.102.7 (Continued)

```
      TF(L2)=B(2)
      GO TO 3300
3250  TF(L2)=B(1)
      TC(L1)=B(4)
      TC(L2)=B(3)
3300  DT(TCN)=B(3)
      DT(TDN)=B(4)
      DT(TFN)=B(5)
      DT(TFA1)=B(1)
      DT(TFA2)=B(2)
      RETURN
      END
```

7.0 OUTPUT SUBROUTINES

The output subroutines comprising, TSTORE, TGRAPH and SCALED are currently dedicated to producing print plots of the data calculated by the program.

Current options allow maximum or minimum calculated values to be substituted for plot values in event these max or min values occurred between plot intervals. This assures that the max or min values calculated are reflected in the output plots. Another option allows tabulation of all calculated values for each plot variable

7.1 SUBROUTINE TSTORE

Subroutine TSTORE, which is called by THYTR, reads output requirements and stores data required for output plots, and prints an index of all the plots.

7.1.1 Math Model

Not applicable.

7.1.2 Assumptions

Not applicable.

7.1.3 Computation Methods

Section 1000

Section 1000 reads in all the plot information for line and component plots.

Section 3000

This section first performs a test to determine if the current time step is also a plot time, if so line or component data is stored. If it is not time to store but the MAX/MIN option has been exercised, tests are made to determine if the current calculated value is less than or greater than (depending on which option was exercised) the previous value stored, if so the stored value is replaced by the current calculated value. If the LIST option has been exercised every calculated plot variable is printed. Once all or a max of 101 points have been stored, TGRAPH is called to plot the points. A test is then performed to determine if more than 101 points are to be plotted, if so the additional points (up to 101) are calculated and stored as before. TGRAPH is again called to plot these points. This procedure is repeated until all points have been plotted.

7.1.4 Approximations

Not applicable.

7.1.5 Limitations

Not applicable.

7.1.6 Variable Listing

Variable	Description		Dimension
I	Counter		-
INDEX	Line Number Associated with Pressure and/or Flow Plots		-
IPLT	Number of Plots Required along line INDEX		-
IPTS	Dummy Variable		-
J	Counter		-
LIST	Input Integer Value 0 (No List) of 1 (List of all Points		-
LPT	Coded Input	1 = Pressure 2 = Flow 3 = Component Temperature 4 = Fluid Temperature 5 = Wall Temperature	-
M	Counter		-
MXTREM	Dummy Variable		-
N	Counter		-
NABSQ	Input Integer	0 = Normal Graphs 1 = Prints Magnitude	-
NISTEP	Counter		-
NLPLTC	Number of Line Plot Points		-
NOGRAF	Not Used		-
NOMSG	Not Used		
NOSTOP	Not Used		-
NPT	Dummy Variable		-

7.1.6 (Continued)

Variable	Description	Dimension
NXTREM	Input Integer Value 0 (Normal Plot), +1 (Plot with Max Values) or -1 (Plot with Min Values)	-
N1	Counter	-
Y	Dummy Variable	-
YY( )	Array Used to Store Line Positions of Required Plots	-

7.1.7 Subroutine Listing

```
      SUBROUTINE TSTORE
C**** REVISED AUGUST 5, 1975 ****
      COMMON DUM(3500),VSTORE(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NLL
      COMMON /LIMIT/MNLINE,MNEL,MNLEG,MNNODE,MNPLOT,MNLPTS,MDS
      COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
     + NLSEG(150)
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR,INLL
      COMMON /PLOT/TITLE(20),PLTDEL,NPTS,IPOINT,ISTEP,TFINAL,NLPLT(61,3)
     + ,NABSQ,NTOPL,NTOLPL
      DIMENSION YY(10),DD(1400),ITITLE(40),IN(40),IY(40),IIC(40),IC(40),
     1IITITLE(40),ICHAR(12),ICI(40)
      COMMON/COMPD/D(4500),L(1500),LE(99,4)
      EQUIVALENCE(DD(1),D(1))
      DATA ICHAR/4HLINE,4HUPS ,4HCOMP,4HVAR ,2H P,2H Q,2HTC,
     +2HTF,2HTW,2H  ,4HDNS ,1H /
      IF(IENTR) 1000, 1000,3000
 1000 CONTINUE
      IPT=0
      NISTEP=0
      ITER=0
      IPTS=NPTS
      IF(NPTS.GT.101) NPTS=101
C
      NTOLPL=0
      READ(5,103) NLPLTC,NTELPL,NXTREM,LIST,NOSTOP,NOMSG,NOGRAF,NABSQ
  103 FORMAT(8I5)
      IF(NLPLTC.EQ.0) GO TO 142
      DO 140 I=1,NLPLTC
      READ(5,109) INDEX,IPLT,(IY(M),M=1,IPLT)
  109 FORMAT(16I5)
      DO 130 M=1,IPLT
      LPT=IY(M)
      LOC=2*INDEX-1
      IF(LPT.LE.0)LOC=2*INDEX
      NTOLPL=NTOLPL+1
C     LOC - ADDRESS OF VARIABLE IN P,Q,TC,TW OR TF ARRAY
C     INDEX - LINE NUMBER
C     LPT - CODED INPUT
      NLPLT(NTOLPL,2) = INDEX
      NLPLT(NTOLPL,3) = LPT
      NLPLT(NTOLPL,1) = LOC
  130 CONTINUE
  140 CONTINUE
  142 CONTINUE
      NTOPL=NTOLPL+NTELPL
      IF(NTOLPL.GT.MNPLOT) NTOLPL=MNPLOT
C
      IF(NTELPL.EQ.0) GO TO 144
```

7.1.7 (Continued)

```
      READ(5,143) ((NLPLT(I+NTOLPL,2),NLPLT(I+NTOLPL,3)),I=1,NTELPL)
  143 FORMAT(16I5)
  144 CONTINUE
      IF(NTOLPL+NTELPL.GT.MNPLOT) NTELPL=MNPLOT-NTOLPL
      IF(NTOLPL+NTELPL.NE.NTOPL) WRITE(6,520)
      NTOPL=NTOLPL+NTELPL
      IF(NTELPL.EQ.0) GO TO 3000
      LPT=NTOLPL+1
      DO 1200 I=LPT,NTOPL
      NPT=NLPLT(I,2)
      N  =NLPLT(I,3)
      IF(N) 1150,1180,1160
 1150 N1=-LE(NPT,3)+N+1
      GO TO 1170
 1160 N1=LE(NPT,2)+N-1
 1170 NLPLT(I,1)=N1
      GO TO 1200
 1180 NLPLT(I,1)=1
 1200 CONTINUE
C
 3000 CONTINUE
C
      IF(ISTEP.EQ.NISTEP) GO TO 2010
      IF(NXTREM.EQ.0.AND.LIST.EQ.0) RETURN
      MXTREM=NXTREM
      GO TO 2020
 2005 NISTEP=NISTEP-IPOINT
 2010 NXTREM=0
      IPT=IPT+1
      NISTEP=NISTEP+IPOINT
      VSTORE(IPT)=TIME
 2020 NPT=IPT
      N1=0
      DO 2200 I=1,NTOPL
      NPT=NPT+NPTS
      N=NLPLT(I,1)
      NN=IABS(NLPLT(I,3))
      IF(I.GT.NTOLPL) GO TO 2050
      GO TO(2901,2902,2903,2904,2905)NN
 2901 Y=P(N)
      GO TO 2080
 2902 Y=Q(N)
      IF(NLPLT(I,3).GT.0.0) Y=-Q(N)
      GO TO 2080
 2903 Y=TC(N)
      GO TO 2080
 2904 Y=TF(N)
      GO TO 2080
 2905 Y=TW(N)
      GO TO 2080
```

7.1.7 (Continued)

```
 2050 IF(N) 2060,2150,2070
 2060 Y=DD(-N)
      GO TO 2080
 2070 Y=D(N)
 2080 IF(MXTREM)2090,2085,2100
 2085 IF(ISTEP+IPOINT.EQ.NISTEP) GO TO 2110
      GO TO 2130
 2090 IF(VSTORE(NPT).GT.Y) GO TO 2110
      GO TO 2120
 2100 IF(VSTORE(NPT).GE.Y) GO TO 2120
 2110 VSTORE(NPT)=Y
 2120 IF(LIST.EQ.0) GO TO 2200
      IF(I.EQ.1) WRITE(6,2211) TIME
 2130 N1=N1+1
      YY(N1)=Y
      IF(N1.NE.10) GO TO 2200
      WRITE(6,2210) YY
      N1=0
      GO TO 2200
 2150 WRITE(6,2220) I
      Y=TIME
      GO TO 2080
 2200 CONTINUE
      IF(N1*LIST.NE.0)WRITE(6,2210) (YY(I),I=1,N1)
      IF(IPT.NE.NPTS) RETURN
      IF(ITER.EQ.1)GO TO 2550
      WRITE(6,1601)
      WRITE(6,1603)
      IF(MXTREM) 5,12,10
    5 WRITE(6,1606)
      GO TO 1220
   10 WRITE(6,1607)
      GO TO 1220
   12 WRITE(6,1608)
      GO TO 1220
 1220 WRITE(6,1603)
      JJ=0
      II=NTOPL
      III=0
      JIJ=1
      DO 1300 I=1,II
      J=I
      JJ=JJ+1
   15 N1=NLPLT(I,1)
      NPT=NLPLT(I,2)
      N=NLPLT(I,3)
      IF(I.GT.NTOLPL) GO TO 2500
      ITITLE(JJ)=ICHAR(1)
      IN(JJ)=NPT
      IITITLE(JJ)=ICHAR(2)
```

7.1.7 (Continued)

```
      IF(N.LT.0)IITITLE(JJ)=ICHAR(11)
      LL=IABS(N)
      IC(JJ)=ICHAR(4+LL)
      IIC(JJ)=ICHAR(10)
      ICI(JJ)=ICHAR(12)
      GO TO 1700
 2500 ITITLE(JJ)=ICHAR(3)
      IITITLE(JJ)=ICHAR(4)
      IN(JJ)=NPT
      ICI(JJ)=ICHAR(12)
      IY(JJ)=N
      IC(JJ)=ICHAR(10)
      IIC(JJ)=ICHAR(10)
 1700 IF(JJ.LT.10.AND.I.LT.NTOPL) GO TO 1300
      III=III+JJ
      WRITE(6,1500)((ICI(JJJ),(JJJ)),JJJ=JIG,III)
      JIG=JIG+10
      IF(I.GT.NTOLPL) GO TO 75
      WRITE(6,1600) ((ITITLE(JJJ),IN(JJJ),IIC(JJJ)),JJJ=1,JJ)
      WRITE(6,1699)((IITITLE(JJJ),IC(JJJ)),JJJ=1,JJ)
      GO TO 16
   75 WRITE(6,1604) ((ITITLE(JJJ),IN(JJJ),IC(JJJ)),JJJ=1,JJ)
      WRITE(6,1604) ((IITITLE(JJJ),IY(JJJ),IIC(JJJ)),JJJ=1,JJ)
   16 WRITE(6,1605)
      JJ=0
 1300 CONTINUE
      WRITE(6,1602)
      WRITE(6,1603)
      ITLR=1
 2550 CALL TGRAPH
      IF(IPTS-NPTS) 2300,2350,2310
 2300 NPTS=IPTS
      IPT=0
      GO TO 2005
 2310 NPTS=101
      IPTS=IPTS-100
      IPT=0
      GO TO 2005
 2350 CONTINUE
      RETURN
  520 FORMAT(5X,42HTOO MANY PLOTS REQUESTED MAX NUMBER IS 60     )
 2210 FORMAT(5X,10E12.5)
 2211 FORMAT(//,5X,25HDATA CALCULATED AT TIME =,F8.4)
 2220 FORMAT(5X,45HVALUE OF N IN 2000 SECTION OF COMP IS ZERO I=   ,I5)
 1500 FORMAT(4X,10(A1,6HGRAPH ,I4,1H ))
 1600 FORMAT(5X,10(A4,2H  ,I4,A2))
 1699 FORMAT(5X,10(A4,4H    ,A2,2H  ))
 1601 FORMAT(1H1,42X,35HVARIABLES SELECTED FOR OUTPUT PLOTS)
 1602 FORMAT(1H1,53X,13HHYTRAN OUTPUT)
 1603 FORMAT(1H0)
```

7.1.7 (Continued)

```
 1604 FORMAT(5X,10(A4,I6,A2))
 1605 FORMAT(1H )
 1606 FORMAT(28X,71HVALUES PLOTTED REPRESENT MINIMUM VALUES CALCULATED I
     1N THE TIME INTERVAL)
 1607 FORMAT(28X,71HVALUES PLOTTED REPRESENT MAXIMUM VALUES CALCULATED I
     2N THE TIME INTERVAL)
 1608 FORMAT(21X,75HVALUES PLOTTED REPRESENT THE ACTUAL VALUES CALCULATE
     3D AT EACH PLOT INTERVAL)
      END
```

7.2 SUBROUTINE TGRAPH

Subroutine TGRAPH produces print plots of the output data stored in VSTORE ( ).

Most computers will have their own version of this subroutine which could be used if necessary. However, since the plotted output is such an integral part of HYTTHA, this subroutine has been added to avoid the problems involved in changing from one computer to another.

7.2.1 Theory - Not applicable.

7.2.2 Assumptions - Not applicable.

7.2.3 Limitations

The program is executed one for each plot, up to the total number of plots NTOPL. The DO 901 J = 1, NTOPL controls this loop.

The first section which sets the X scale, is only executed on the first pass, when J = 1.

The program currently uses VSTORE (1) as XMIN and VSTORE (NPTS) as XMAX.

In the second section a DO loop is used to find the maximum and minimum values of the Y data to be plotted, using the functions AMAX1 (YMAY, VSTORE, (I+IADD)) and AMIN1 (YMIN, VSTORE (I+IADD)).

With the maximum and minimum values established, a check is made to see if they are equal, if they are, 25 is added to YMAX, and YMIN is set at 50 less than that, to avoid a fruitless search for a suitable scale.

Subroutine SCALED is then called to obtain a preferred scale for the Y axis, and returns with values for YMAX and YMIN.

The next section finds the type of plot and sets the plot character, P, Q, T or C and the data to be written at the bottom of the output plot.

The routine then starts the output plot section by going to the top of a new page, and proceeds to plot the output data, line by line until the plot is complete.

At the bottom of the plot a descriptive line is written which gives the line number and distance along the line for line pressure or flow plots or the variable number and the component number if it is a component data plot.

The next printed line is the title of the run, which was inputted on the first data card.

When all the plots have been completed, a listing of the titles is provided on the final page of the program output.

7.2.4 Approximations

Not applicable.

7.2.5 Limitations

The basic limitation of a print plot is the number of points that can be plotted on a single page graph and the resulting inaccuracy in reading the graph. To an extent these limitations can be over come by use of the MAX/MIN and LIST options noted in Section 8.0 of Volume I of this report.

7.2.6 Variable Listing

Variable	Description	Dimensions
AVS	Absolute Value of VS	-
DIST	Distance of Plot Point Down a Line	IN
I	Counter	-
IADD	Address J*NPTS	-
ICHAR	Plot Character	-
ICHAR( )	X and Y Axis Write Characters	-
ISP	Counter	-
ISPACE( )	Temporary Variable for Writing X and Y Axis Scales	-
ITEST	Counter	-
J	Counter Indicating Plot Number	-
L	Dummy Variable	-
LINE	Integer Counter for Plot Line Number	-
NABSQ	Integer Value 1 or 0 Used as Indicator	-
NCHAR	Dummy Variable Representing Plot Character	-
NVAR	Dummy Variable Representing Point at which Line Plot is Taken or Component Number	-
SP	Column Number Nearest to the Ith Value of X- Variable	-

7.2.6 (Continued)

Variable	Description	Dimensions
VS	Dummy Variable	-
XAX	Temporary Variable for Writing X Axis Scale Values	-
XDELTA	Distance Between Stored Points on X Axis	-
XMAX	Last (Largest) X Axis Value	-
XMIN	First (Lowest) X Axis Value	-
XSCALE	X Scale Range	-
Y	Temporary Variable (Y Axis Scale Value)	-
YDELTA	Distance Between Stored Points on the Y Axis	-
YLAST	Last Y Axis Scale Value	-
YLO	Lowest Value in Search Range	-
YMAX	Maximum Value to be Plotted	-
YMIN	Minimum Value to be Plotted	-
YUP	Highest Value in Search Range	-

7.2.7 Subroutine Listing

```
      SUBROUTINE TGRAPH
C**** REVISED AUGUST 5, 1975 ****
      COMMON DUM(3500),VSTORE(1)
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NLL
      COMMON /LIMIT/MNLINE,MNLL,MNLEG,MNNODE,MNPLOT,MNLPTS,MDS
      COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
     + NLSEG(150)
      COMMON /COMP/LTYPE(99),NC(99),KTEMP(99),IND,IENTR
      COMMON /PLOT/TITLE(20),PLTDEL,NPTS,IPOINT,ISTEP,TFINAL,NLPLT(61,3)
     + ,NABSQ,NTOPL,NTOLPL
      DIMENSION ISPACE(101),ISTR(2),XAX(6),ICHART(9)
      DATA ICHART/1HO,1HP,1HC,1HI,1H-,1H+,1H ,1H*,1HT/
      DATA ITEST,XSCALE/0,0.0/
      DATA ISTR/10H UPSTREAM ,10HDOWNSTREAM/
C-----BEGIN OUTER LOOP. FIND X PARAMETERS ON FIRST PASS ONLY
    1 DO 901 J=1, NTOPL
      IADD=J*NPTS
      IF(J.NE.1)GO TO 2
      XMAX=VSTORE(NPTS)
      XMIN=VSTORE(1)
      ITEST=ITEST+1
      IF(ITEST.NE.1) XMAX=XMIN+XSCALE
      XSCALE=XMAX-XMIN
C     CALL SCALED(XMAX,XMIN)
      XDELTA=(XMAX-XMIN)/100.
C-----FIND Y PARAMETERS
    2 YMAX=VSTORE(1+IADD)
      YMIN=YMAX
      DO 902 I=2, NPTS
      YMAX=AMAX1(YMAX,VSTORE(I+IADD))
  902 YMIN=AMIN1(YMIN,VSTORE(I+IADD))
      NABSQ=0
      IF(J.GT.NTOLPL.OR.NLPLT(J,1).GE.0) GO TO 905
      IF(NABSQ.EQ.0) GO TO 905
      IF(YMIN.GT.0) GO TO 905
      NABSQ=1
      IF(ABS(YMIN).GT.YMAX) YMAX=ABS(YMIN)
  905 IF(YMAX.NE.YMIN)GO TO 9020
      YMAX=YMAX + 25.
      YMIN=YMAX - 50.
      GO TO 9025
 9020 AMAX = (YMAX+YMIN)*.001
      IF((YMAX-YMIN).GT.AMAX) GO TO 9025
      YMAX = YMAX+AMAX
      YMIN = YMIN-AMAX
 9025 CALL SCALED(YMAX,YMIN)
      YDELTA=(YMAX-YMIN)/50.
C-----FIND LINE/COMPONENT NUMBER, TYPE OF PLOT, OUTPUT DATA
      L=NLPLT(J,2)
```

7.2.7 (Continued)

```
      IF (J.GT.NTOLPL) GO TO 5
      INJ=IABS(NLPLT(J,3))
      NVAR=NLPLT(J,3)
      GO TO(4,3,730,730,730)INJ
    3 ICHAR=ICHART(1)
      GO TO 6
    4 ICHAR=ICHART(2)
      GO TO 6
  730 ICHAR=ICHART(9)
      GO TO 6
    5 ICHAR=ICHART(3)
C-----GO TO TOP OF NEXT PAGE
    6 WRITE(6,501)
C-----LOOP FOR EACH PLOT LINE.
      Y=YMAX + YDELTA
    7 DO 907 LINE=1, 51
      YLAST=Y
      Y=Y-YDELTA
      YUP=Y+YDELTA/2.
      YLO=Y-YDELTA/2.
C-----FIRST + LAST CHAR. ON LINE = *I*
      ISPACE(1)=ICHART(4)
      ISPACE(101)=ICHART(4)
C-----FIRST + LAST LINES ALL *-*, EXCEPT *+* IN 11,21,31,41,...,81,+91
      IF(LINE.NE.1 .AND. LINE.NE.51)GO TO 11
    9 DO 909 ISP=2,100
      IF((ISP-1).EQ.(ISP-1)/10*10)GO TO 10
      ISPACE(ISP)=ICHART(5)
      GO TO 909
   10 ISPACE(ISP)=ICHART(6)
  909 CONTINUE
      GO TO 14
C-----INITIALIZE COL. 2-100 ON LINES 2-50 TO * *, OR *+----+-* IF AXIS
   11 IF(Y.LE.0. .AND.YLAST.GT.0.)GO TO 13
   12 DO 912 ISP=2, 100
  912 ISPACE(ISP)=ICHART(7)
      GO TO 14
   13 DO 913 ISP=2,100
      ISPACE(ISP)=ICHART(5)
  913 IF((ISP-1).EQ.(ISP-1)/10*10)ISPACE(ISP)=ICHART(6)
C---- SEARCH Y-VALUE ARRAY FOR THOSE IN RANGE YLO.LT.VALUE.GE.YUP
   14 DO 914 I=1, NPTS
      VS=VSTORE(I+IADD)
      NCHAR=ICHAR
      IF(VS.GT.YLO .AND. VS.LE.YUP)GO TO 145
      IF(NABSC.NE.1) GO TO 914
      AVS=ABS(VS)
      IF(AVS.LT.YLO.OR.AVS.GT.YUP) GO TO 914
      NCHAR=ICHART(8)
C-----FIND COLUMN NUMBER NEAREST TO I-TH VALUE OF X-VARIABLE WHEN SCALED
```

```
  145 SP=(VSTORE(I)-XMIN)/XDELTA + 1
      IF(SP-AINT(SP).GT.0.50) SP=SP + 0.50
      ISP=SP
C-----CHECK ISP. IF LT 0 OR GT 102, ERROR; IF 0, ADD 1; IF 102, SUBT. 1
      IF(ISP) 914, 15, 16
   15 ISP=1
      GO TO 18
   16 IF(ISP-102)18,17,914
   17 ISP=101
   18 ISPACE(ISP)=NCHAR
  914 CONTINUE
C-----LINES 1,11,21,31,41,+51 HAVE Y-VALUES; THESE LINES, PLUS LINES 6,
C     16,26,... ALSO HAVE *+* IN COL. 1+101 IF EMPTY
      IF((LINE-1).NE.(LINE-1)/5*5)GO TO 19
      IF(ISPACE(1).NE.ICHAR) ISPACE(1)=ICHART(6)
      IF(ISPACE(101).NE.ICHAR)ISPACE(101)=ICHART(6)
      IF((LINE-1).NE.(LINE-1)/10*10)GO TO 19
C-----WRITE OUT PLOT LINE, CONTINUE
      WRITE(6,602)Y, ISPACE
      GO TO 907
   19 WRITE(6,603)ISPACE
  907 CONTINUE
C-----CALCULATE + PRINT X-AXIS VALUES
   20 DO 920 I=1, 6
  920 XAX(I)=XMIN + (I-1)*20.*XDELTA
      WRITE(6,604) XAX
C-----WRITE LOWER TITLES + VALUES, REENTER OUTER LOOP
      IF (J.GT.NTOLPL) GO TO 23
      II=ISTR(1)
      IF(NVAR.LT.0)II=ISTR(2)
      NN=IABS(NVAR)
      GO TO(711,712,713,714,715)NN
  711 WRITE(6,606)J,II,L
      GO TO 900
  712 WRITE(6,605)J,II,L
      GO TO 900
  713 WRITE(6,612)J,II,L
      GO TO 900
  714 WRITE(6,613)J,II,L
      GO TO 900
  715 WRITE(6,614)J,II,L
      GO TO 900
   23 WRITE(6,607)J,NLPLT(J,3),L
  900 CONTINUE
      WRITE(6,608)TITLE
  901 CONTINUE
 1000 WRITE(6,601)
      WRITE(6,610)
      WRITE(6,611)
      IF(TIME.LT.TFINAL-DELT)RETURN
```

BEST AVAILABLE COPY

```
      DO 1250 J=1,NTOPL
      L=NLPLT(J,3)
      IF(J.GT.NTOLPL) GO TO 50
      II=ISTR(1)
      IF(L.LT.0)II=ISTR(2)
      NN=IABS(L)
      GO TO(701,702,703,704,705)NN
  701 WRITE(6,606)J,II,NLPLT(J,2)
      GO TO 1250
  702 WRITE(6,605)J,II,NLPLT(J,2)
      GO TO 1250
  703 WRITE(6,612)J,II,NLPLT(J,2)
      GO TO 1250
  704 WRITE(6,613)J,II,NLPLT(J,2)
      GO TO 1250
  705 WRITE(6,614)J,II,NLPLT(J,2)
      GO TO 1250
   50 WRITE(6,607)J,NLPLT(J,3),NLPLT(J,2)
 1250 CONTINUE
  601 FORMAT(1H1)
  602 FORMAT(1X,15X,F12.4,1X,101A1)
  603 FORMAT(1X,28X,101A1)
  604 FORMAT(1X,23X,5(F9.3,11X),F9.3)
  605 FORMAT(1X,28X,6HGRAPH ,I3,1X,33H FLOW (CU.IN/SEC) VS. TIME (SEC.)
     +,A10,16H OF LINE NUMBER ,I5)
  606 FORMAT(1X,28X,6HGRAPH ,I3,1X,33H  PRESSURE (PSIA) VS. TIME (SEC.)
     +,A10,16H OF LINE NUMBER ,I5)
  612 FORMAT(1X,28X,6HGRAPH ,I3,1X,47H  COMPONENT TEMPERATURE (DEG.F) VS
     +. TIME (SEC.),A10,12H OF LINE NO.,I5)
  613 FORMAT(1X,28X,6HGRAPH ,I3,1X,43H  FLUID TEMPERATURE (DEG.F) VS. TI
     +ME (SEC.),A10,12H OF LINE NO.,I5)
  614 FORMAT(1X,28X,6HGRAPH ,I3,1X,42H  WALL TEMPERATURE (DEG.F) VS. TIM
     +E (SEC.),A10,12H OF LINE NO.,I5)
  607 FORMAT(1X,28X,6HGRAPH ,I3,1X,18H  VARIABLE NUMBER ,I3,21H OF COMPO
     +NENT NUMBER ,I3,38H VS. TIME (SEC.).  THE VARIABLE IS ---  )
  608 FORMAT(1X,28X,20A4)
  610 FORMAT(1H0,65X,27HHYTRAN PROGRAM OUTPUT PLOTS)
  611 FORMAT(1H0)
      RETURN
      END
```

7.3 SUBROUTINE SCALED

The subroutine SCALED is used by TGRAPH to obtain a preferred scale for the X and Y axis of the print plot graphs.

The number of divisions on the X axis = 100 and the number of divisions on the Y axis = 50, a preferred scale system was chosen which would give a difference between RMAX and RMIN of either 1.0*10**N, 2.0*10**N or 5.0*10**N where N can be +ve or -ve.

The graph data MAX and MIN is centered between RMAX and RMIN unless either RMAX and RMIN can be set to zero.

An overriding requirement is that the scales should be at some reasonable number for easy reading hence with a range of 5000, the MIN can be set at intervals of 500, or range/10. This sometimes leads to a larger scale being used than would expected from the actual range.

The goal however was graphical readability and scalability without the need to resort to a calculator to find the value of a point, and in meeting this goal we have payed some penalty in the size of the actual graph.

7.3.1 Theory

Not applicable.

7.3.2 Assumptions

Not applicable.

7.3.3 Computation

See subroutine listing.

7.3.4 Approximation

Not applicable.

7.3.5 Limitations

In its present form SCALED gives inconsistent answers for small values of RMAX and RMIN, and is not currently used to scale the X axis.

7.3.6 Variable Listing

Name	Description	Dimension
AMAX	Maximum value to be plotted	-
AMIN	Minimum value to be plotted	-
IBOT	Variable used to calculate Y axis scale values.	-
IEMAX	Variable used to calculate Y axis scale values	-
IEXP	Variable used to calculate Y axis scale values	-
ITOP	Variable used to calculate Y axis scale values	-
J	Integer counter	-
MANT	Variable used to calculate Y axis scale values	-
RANGE	Range of values to be plotted	-
RMAX	Maximum Y axis scale value	-
RMIN	Minimum Y axis scale value	-
SCALE(-)	Scale factors for Y axis	-

7.3.7 Subroutine Listing

```
      SUBROUTINE SCALED(RMAX,RMIN)
      DIMENSION SCALE(6)
      DATA SCALE/.5,1.,2.,5.,10.,20./
C-----FIND THE RANGE OF VALUES *RANGE*, AND PLACE ACTUAL MAX AND MIN
C-----POINTS IN *AMAX* AND *AMIN*
      RANGE=RMAX-RMIN
      AMAX=RMAX
      AMIN=RMIN
C-----FIND AN INTEGER EXPONENT *IEXP* AND BASE *MANT* SUCH THAT THE
C-----VALUE OF MANT**IEXP IS .GE. RANGE
      IEXP=ALOG10(RANGE)
      MANT=RANGE/10.**IEXP
      IF(RANGE.GT.MANT*10.**IEXP)MANT=MANT+1
C-----USING MANT, SELECT ONE OF THE PREFERRED SCALES
      IF(MANT.GT.10)GO TO 70
      IF(MANT.LT.1) GO TO 70
      GO TO(80,90,100,100,100,110,110,110,110,70), MANT
   70 MANT=1
      IEXP=IEXP+1
   80 J=2
      GO TO 120
   90 J=3
      GO TO 120
  100 J=4
      GO TO 120
  110 J=5
C-----SET *IEMAX* EQUAL TO THE EXPONENT OF 10. CORRESPONDING TO RMAX
  120 IF(RMAX.EQ.0.) GO TO 121
      IEMAX=ALOG10(ABS(RMAX))
C-----USE AMAX AND IEMAX TO FIND A POSSIBLE MAXIMUM VALUE FOR THE
C-----SCALE.  PLACE THE VALUE IN RMAX, AND COMPARE WITH THE ACTUAL
C-----MAXIMUM POINT.
      RMAX=INT(ABS(AMAX)/10.**IEMAX)*10.**IEMAX*SIGN(1.0,AMAX)
  121 IF(RMAX.GE.AMAX)GO TO 130
C-----IF RMAX IS .LT. ACTUAL MAX POINT, INCREASE IT BY 5
C-----PERCENT AND RECHECK--REPEAT AS NECESSARY
      RMAX=RMAX+.05*SCALE(J)*10.**IEXP
      GO TO 121
C-----SET THE SCALE'S MINIMUM BY SUBTRACTING SCALE(J)**IEXP FROM RMAX
C-----IF THE ACTUAL MINIMUM *AMIN* LIES WITHIN THE RANGE NOW DEFINED
C-----BY RMAX AND RMIN, CONTINUE.
  130 RMIN=RMAX-SCALE(J)*10.**IEXP
      IF(RMIN.LE.AMIN)GO TO 150
C-----GO TO THE NEXT LARGEST SCALE, RECALCULATE RMIN, AND RECHECK
      J=J+1
      IF(J.LT.5.5)GO TO 130
      J=1
      IEXP=IEXP+1
      GO TO 130
C-----IF THE SCALE'S MIN IS .LT. ZERO, BUT THE ACTUAL MIN IS POSITIVE,
```

7.3.7 (Continued)

```
C-----SHIFT THE SCALE UP SO THAT THE SCALE BEGINS AT ZERO
  150 IF(RMIN*AMIN.GT.0.)GO TO 170
      RMIN=0.
  160 RMAX=SCALE(J)*10.**IEXP
C-----DUE TO THE SHIFT, IT MAY BE POSSIBLE TO DECREASE THE SCALE TO
C-----THE NEXT SMALLEST SIZE
      IF(AMAX.GT.1.000001*SCALE(J-1)*10.**IEXP) RETURN
      J=J-1
      IF(J.GT.1.5) GO TO 160
      J=4
      IEXP=IEXP-1
      GO TO 160
C-----IF RMIN IS POSITIVE AND NEAR ZERO, SHIFT THE SCALE DOWN TO 0. MIN
  170 IF(RMIN.LT.0.)GO TO 175
      IF(RMIN.GT..1*RMAX)GO TO 180
      RMIN=0.
      RMAX=SCALE(J)*10.**IEXP
C-----IF THE SHIFT DOWN CAUSES RMAX TO LIE BELOW THE ACTUAL MAX,
C-----INCREASE THE SCALE RANGE TO THQ NEXT LARGEST
      IF(RMAX.LT.AMAX)RMAX=SCALE(J+1)*10.**IEXP
      RETURN
C-----IF RMAX IS NEGATIVE AND NEAR ZERO, SHIFT THE SCALE UP TO 0. MAX
 175  IF(RMAX.GE.0.) GO TO 180
      IF(-RMAX.GT.-0.1*RMIN) GO TO 180
      RMAX=0.
      RMIN=-SCALE(J)*10.**IEXP
C-----IF THE SHIFT UP CAUSES RMIN TO LIE ABOVE THE ACTUAL MIN,
C-----INCREASE THE SCALE RANGE TO THE NEXT LARGEST
      IF(RMIN.GT.AMIN) RMIN=-SCALE(J+1)*10 **IEXP
      RETURN
C-----CENTER THE SCALE ABOUT THE ACTUAL RANGE OF POINTS
  180 ITOP=(RMAX-AMAX)/(.05*SCALE(J)*10.**IEXP)
      IBOT=(AMIN-RMIN)/(.05*SCALE(J)*10.**IEXP)
      IDIF=(ITOP-IBOT)/2
      IF(IDIF.LE.0) RETURN
      RMIN=RMIN-IDIF*.05*SCALE(J)*10.**IEXP
      RMAX=RMIN+SCALE(J)*10.**IEXP
      RETURN
      END
```

8.0 UTILITY SUBROUTINES

The utility subroutines have been added to avoid some of the annoying problems encountered when a program is transferred to another system which may have similar but incompatible library routines.

INTERP and DISER1 both of which started as library routines, have been modified to cut running costs wherever possible.

A skilled user can probably replace INTERP, DISER1, SIMULT, XLIMIT, and TGAUSS with local library routines and operate efficiently.

LAGRAN, UFW and FRIC are specialized subprograms developed for use in HYTTHA. The LAGRAN subroutine not only interpolates data points but computes viscosity and a viscosity pressure correction factor. Function UFW provides a heat transfer coefficient, and FRIC give laminar and turbulent flow coefficients.

8.1 INTERP SUBROUTINE

The INTERP subroutine provides interpolation for continuous or discontinuous functions of the form Y = f(X). INTERP is a shortened version of a MCAUTO library functional subroutine named DISCOT.

INTERP uses two other subroutines, DISER1 and LAGRAN, to derive the dependent variable from tabulated data input by the programmer or already existing in the program subroutine. Subroutine DISER1 gives the data points around the X variable. Lagrange's interpolation formula is used in the subroutine LAGRAN to obtain a Y value. For an X value lying outside the range of the tabulated data, the Y value will be extrapolated. Fluid viscosities are calculated using a modified Walther equation (Reference 9.5).

8.1.1 Solution Method. The INTERP subroutine provides the necessary control parameters to DISER1 and LAGRAN to yield a dependent variable. The subroutine arguments are:

Subroutine INTERP (X, TABX, TABY, NC, NY, Y, IND)

where:

X - Argument of function Y = f(X)

TABX - X array of independent variables in ascending order

TABY - Y array of dependent variables in ascending order

NC - Control word

Tens Digit - Degree of interpolation

Units Digit - = 1 Walther equation

= 0 LAGRAN interpolation

NY - Number of data points in the Y array

Y - Dependent variable

IND - Error indicator

0 = Normal interpolation

1 = Extrapolation outside range of data points.

8.1.2 Assumptions. Not applicable

8.1.3 Computations. The degree of interpolation will be decoded from the control word NC in the INTERP subroutine argument and passed to DISER1. The error indicator IND is set to zero. On finding the data point closest to the X value from DISER1, it is entered into the LAGRAN subroutine argument. If the modified Walther equation is to be used for a viscosity calculation, IDX will be set equal to -1.

8.1.4 Approximations. Not applicable

8.1.5 Limitations. The X and Y data points must be entered in an ascending order. When tabulating a discontinuous function the independent variable (X) at the point of discontinuity is repeated, i.e.,

X_1, X_2, X_3, X_3, X_4, X_5

Y_1. Y_2, Y_3, Y_4. Y_5, Y_6

Thus for discontinuous functions there must be K + 1 points above and below the discontinuity, where K is the degree of interpolation.

8.1.6 INTERP Variable Names.

Variables	Description	Dimensions
IDX	Degree of interpolation	-
IND	Solution indicator	-
	= 0 Normal interpolation	
	= 1 Extrapolation outside of data range	
NC	Control word	-
NPX	Dummy array	-
NPX1	Location of data point X, Y for interpolation	-
NY	Number of Y data points	-
TABX	X array of data points	-
TABY	Y array of data points	-
X, XA	Independent variable	-
Y	Dependent variable	-

8.1.7 Subroutine Listing

```
SUBROUTINE INTERP(X,TABX,TABY,NC,NY,Y,IND)
DIMENSION TABX(1),TABY(1),NPX(8)
IDX=(NC-(NC/100)*100)/10
IND=0
XA=X
CALL DISER1(XA,TABX,1,NY,IDX,NPX,IND)
NPX1=NPX(1)
IF((NC-IDX*10).EQ.1)IDX=-1
IF((NC-IDX*10).EQ.2)IDX=-2
CALL LAGRAN(XA,TABX(NPX1),TABY(NPX1),IDX+1,Y)
RETURN
END
```

8.2 DISER1 SUBROUTINE

The subroutine DISER1 will return the array location of the lower bound value of the interval in which the independent variable lies. DISER1 is a modification of a MCAUTO library subroutine named DISSER.

The arguments for the DISER1 subroutine are as follows:

Subroutine DISER1 (XA, TAB, I, NX, ID, NPX, IND)

XA - Independent variable

TAB - X array

I - Tabulated data location

NX - Number of points in the independent array

ID - Degree of interpolation

NPX - Location of lower bound for data point XA, in the TAB array

IND - Indicator

8.2.1 Solution Method. Not applicable

8.2.2 Assumptions. Not applicable

8.2.3 Computations. On entry of the independent variable, XA, and the tabulated data form the TABX array, DISER1 will find the tabulated data values that bound XA, and return the smaller one to the calling program. If XA were to lie outside the lower end of the data, DISER1 would return the first data point as the lower bound. Should XA lie outside the upper tabulated value, the second from the last data point location will be returned by DISER1.

8.2.4 Approximations. Not applicable

8.2.5 Limitations. Not applicable

8.2.6 DISER1 Variable Names

Variable	Description	Dimensions
IND	Solution indicator	-
I, ID, IT, J, NLOC, NLOW, NPB, NPT, NPU, NPX, NUPP, NX, NXX	Integer counters	-
TAB	Array of independent variables	-
XA	Independent variable	-

8.2.7 Subroutine Listing

```
      [illegible] ([illegible],I,[illegible],I[illegible])
      [illegible] (I)
      IF([illegible](I))71,72,72
 71   I[illegible]=I[illegible]+1
      [illegible]=I
      [illegible]
 72   [illegible]=[illegible](I)
      [illegible]=I
      [illegible]
 73   J=I+[illegible]-1
      IF([illegible](J))1,77,7[illegible]
 7[illegible]  I[illegible]=I[illegible]+1
      [illegible]=J-I[illegible]
      [illegible]
 77   [illegible]=[illegible](J)
      [illegible]=J-I[illegible]
      [illegible]
 1    [illegible]=I[illegible]+1
      [illegible]=[illegible]/2
      [illegible]=[illegible]
      IF([illegible])4,5,1[illegible]
 4    I[illegible]=[illegible]-1
      [illegible] 1
 5    [illegible]=I
      [illegible]
 1[illegible]  [illegible]=I+[illegible]
      [illegible]=I+[illegible]-([illegible]+1)
      IF([illegible])1[illegible],1[illegible],11
 11   [illegible]=[illegible]/2+I
      IF([illegible]([illegible]))12,17,1[illegible]
 12   [illegible]=[illegible]/4
      IF([illegible]([illegible]))1[illegible],17,17
 13   [illegible]=[illegible]+[illegible]/4
      IF([illegible]([illegible]))1[illegible],17,17
 1[illegible]  [illegible]=[illegible]/4
      [illegible] 1[illegible]
 17   [illegible]=[illegible]
 1[illegible]  [illegible] II=[illegible]
      [illegible]
      I[illegible]([illegible](II)-[illegible])1[illegible],2[illegible],2[illegible]
 1[illegible]  [illegible]
      [illegible]=[illegible]-[illegible]+1
      [illegible]
 20   [illegible]=[illegible]-[illegible]
      [illegible]
      [illegible]
```

8.3 LAGRAN SUBROUTINE

The LAGRAN subroutine will interpolate or extrapolate a data point from two known tabulated values. In addition, LA^RAN will calculate viscosity using a modified Walther equation. The LAGRAN subroutine arguments are:

Subroutine LAGRAN (XA, X, Y, N, ANS)

XA - Independent variable

X - X array

Y - Y array

N - Degree of interpolation

ANS - Dependent varaible

8.3.1 Math Model. LAGRANGES interpolation equation is used in this subroutine to calculate the dependent variable. The LAGRANGE formula is:

$$P(x) = \sum_{i=0}^{m} L_i(x)\, y_i \qquad (1)$$

Where:

$L_i(x)$ is the Lagrange multiplier function.

$$L_i(x) = \frac{(x-x_0)\,(x-x_1)\cdots(x-x_{i-1})\,(x-x_{i+1})\cdots(x-x_n)}{(x_i-x_0)\,(x_i-x_1)\cdots(x_i-x_{i-1})\,(x_i-x_{i+1})\cdots(x_i-x_n)} \qquad (2)$$

The LAGRANGE equation will generate a polynomial between two data points. The degree of the polynominal will be that specified by the index value N. The dependent variable will be returned as ANS in the subroutine argument.

A modified version of the Walther equation taken from Reference 9.6 is used in the calculation of viscosity. The ASTM charts are based on this equation.

$$\text{LOG}\,[\text{LOG}\,(\nu+c)\,] = A\ \text{LOG}\ {}^{\circ}R + B \qquad (3)$$

Where:

c = a constant

°R = Temperature, °RANKINE

ν = Viscosity, cSt

A,B = Constants for each fluid

LOG = Log to the base 10

The ASTM chart expresses c as a constant varying from 0.75 at 0.4 cSt to 0.6 at 1.5 cSt and above.

8.3.2 <u>Assumptions</u>. The Lagrangian equation generated by the subroutine will only use the data points around the dependent variable to generate a polynomial for interpolation. The last or first set of two data points will be used for extrapolation. The equation used to determine the viscosity uses a constant factor that is applicable to viscosity values of 2 centistokes or more.

8.3.3 <u>Computation</u>. The procedure LAGRAN will perform whether it be interpolation or the viscosity calculation, will always be recognized by testing the N argument in the subroutine statement. If N is equal to zero, then the viscosity will be calculated using the modified Walther equation. Otherwise N will specify the degree of interpolation to be used by the Lagrange formula. Both results will be returned to the calling program through the variable named ANS. The LAGRAN interpolation is a direct application of equation (1) to the given data.

Before evaluating the viscosity equation (3) for the viscosity value at XA temperature, the constants A and B must be calculated. They are solved using the data points that surround the dependent variable, or the first or last set of two data points if the dependent variable lies outside the range of the tabulated data. With the constants calculated for this fluid the viscosity can be computed from Equation (3).

8.3.4 <u>Approximations</u>. In the viscosity calculation, 0.6 was used as a constant factor for all ranges of viscosity. See reference 9.6 for a more thorough discussion.

8.3.5 Limitations. Since the LAGRANGE method only uses two data points to interpolate it can become inaccurate for remotely spaced tabulated data points. Any degree of interpolation greater than two can lead to erroneous results.

For the viscosity equation, any computed value of viscosity less than 2 centistokes cannot be considered accurate, and should be weighed in the final results.

8.3.6 LAGRAN Subroutine Variable Names

Variable	Description	Dimensions
A	Constant for viscosity	-
ANS	Dependent variable	-
B	Constant for viscosity	-
I,J	Integer counters	-
N	Method of solution	-
	N = 0 Viscosity calculation	
	N > 0 Degree of interpolation	
PROD	Lagrange partial product	-
P1	LOG LOG of (Y (1) + C)	cSt
P2	LOG LOG of (Y (2) + C)	cSt
T1	LOG of T (1)	°R
T2	LOG of T (2)	°R
X	X-array	-
XA	Independent variable	-
Y	Y-array	-

8.3.7 Subroutine Listing

```
      SUBROUTINE LAGRAN(XA,X,Y,N,ANS)
      DIMENSION X(1),Y(1)
      IF(N.EQ.-1) GO TO 20
      IF(N.EQ.0)GO TO 10
      SUM=0.0
      DO 3 I=1,N
      PROD=Y(I)
      DO 2 J=1,N
      A=X(I)-X(J)
      IF(A) 1,2,1
 1    B=(XA-X(J))/A
      PROD=PROD*B
 2    CONTINUE
 3    SUM=SUM+PROD
      ANS=SUM
      RETURN
C     VISCOSITY CALCULATION
   10 CONTINUE
      A1=0.
      IF(Y(1).LE.2.)A1=EXP(-1.47-1.84*Y(1)-.51*Y(1)**2)
      A2=0.
      IF(Y(2).LE.2.)A2=EXP(-1.47-1.84*Y(2)-.51*Y(2)**2)
      P1=ALOG10(ALOG10(Y(1)+.7+A1))
      P2=ALOG10(ALOG10(Y(2)+.7+A2))
      T1=ALOG10(X(1)+460.)
      T2=ALOG10(X(2)+460.)
      B=(P1-P2)/(T2-T1)
      A=P1+B*T1
      Z=10**(10**(A-B*ALOG10(XA+460.)))
      IF(Z.LE.2.7)GO TO 11
      ANS=Z-.7
      RETURN
   11 ANS=(Z-.7)-EXP(-.7487-3.295*(Z-.7)+.6119*(Z-.7)**2
     +-.3193*(Z-.7)**3)
      RETURN
C     FLUIDL CALCULATION
   20 CONTINUE
      P1=ALOG10(ALOG10(Y(1)+.6))
      P2=ALOG10(ALOG10(Y(2)+.6))
      T1=ALOG10(X(1)+460.)
      T2=ALOG10(X(2)+460.)
      B=(P1-P2)/(T2-T1)
      A=P1+B*T1
      V0=10**(10**(A-B*ALOG10(XA+460.)))-.6
      T5=10**((.125989+A)/B)
      T1000=10**((-.477159+A)/B)
      S=ALOG10((T5)/100.+1.)-ALOG10((T1000)/100.+1)
      DELX=65.10979*S
      S=6.65/DELX
      ALPHA=3.23523-11.3886*S+13.1735*S*S-4.8881*S*S*S
```

8.3.7 (Continued)

```
[illegible]PS=-5.33125+1[illegible].[illegible]521*S-23.9448*S*S+10.155*S*S*S
CHI=3.35152-13.1273*S+17.1712*S*S-7.6551*S*S*S
[illegible] =[illegible]+[illegible]*[illegible]10([illegible])+CHI*(ALOG10(V0))**2
IF([illegible].[illegible].[illegible].)[illegible]=0
[illegible]
[illegible]
```

8.4 SIMULT SUBROUTINE

SIMULT is a Fortran library subroutine (Reference 9.7) that solves systems of N linear algebraic equations with N unknowns. SIMULT employs Gaussian elimination and positioning for size using the largest pivotal divisor as the solution process.

8.4.1 Solution Method - A system of linear equations may be written:

$$
\begin{array}{lllll}
a_{11}x_1 & + a_{12}x_2 & + \cdot\cdot + a_{1n}\, x_m & = b_1 & \\
a_{21}x_1 & + a_{22}x_2 & + \cdot\cdot + a_{2n}\, x_m & = b_2 & \\
\cdot & \cdot & \cdot\cdot \quad \cdot & \cdot & \qquad (1) \\
\cdot & \cdot & \cdot\cdot \quad \cdot & \cdot & \\
a_{m1}x_1 & + a_{m2}x_2 & + \cdot\cdot + a_{mm}x_n & = b_m &
\end{array}
$$

Rewriting in matrix form:

$$
\begin{bmatrix} a_{11} & a_{12} & \cdot\cdot & a_{1m} \\ a_{21} & a_{22} & \cdot\cdot & a_{2m} \\ \cdot & \cdot & \cdot\cdot & \cdot \\ \cdot & \cdot & \cdot\cdot & \cdot \\ a_{m1} & a_{m2} & & a_{mm} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_m \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_m \end{bmatrix} \qquad (2)
$$

Equation (2) may be further simplified by writing:

$$AX = B \qquad (3)$$

where

A = M * M Matrix of coefficients

B = M Matrix of Constants

X = M Matrix of M unknowns in the system.

The solution of a set of simultaneous linear equations as in (1) is by Gaussian elimination using pivoting. Each stage of elimination consists of interchanging rows when necessary to avoid division by zero or small elements. The forward solution to obtain variable M is done in M stages. The back solution for the other variables is calculated by successive substitutions. Final solution values are developed in matrix B, with variable 1 in B(1), variable 2 in B(2),..., variable M in B(M). If no pivot can be found exceeding a tolerance of 0.0, the matrix is considered singular. The arguments for SIMULT are as follows:

Subroutine SIMULT (CALC1, CALC2, M, J)

where:

CALC1 = A

CALC2 = B

M = number of equations

J = solution indicator

J = 1 when no solution can be found - equations are singular

J = 0 for a normal solution

Both the original CALC1 and CALC2 Matrices are destroyed in the computation. The answers are returned through the CALC2 Matrix.

8.4.2 Assumptions - The basic assumption used in the solution of simultaneous linear equations involves the ability to actually linearize the complex mathematical system that is being described. If this can reasonably be done then a set of equations as in (1) may be written and solved.

8.4.3 Computation - Gaussian elimination and positioning for size using the largest pivotal divisor is used. Positioning for size or pivoting will ordinarily reduce some of the roundoff errors in the solution and may actually allow some ill-conditioned systems to be solved. See Appendix D SSFAN Technical Manual (MDC A3059 Vol II) for a more thorough discussion of this method.

8.4.4 Approximations - The approximations are inherent in the use of the Gaussian elimination procedure as described in Appendix D of the SSFAN Technical Manual.

8.4.5 Limitations - If no equation in the set (1) is a linear combination of the others, the system of equations is said to be linearly independent and a unique solution exists for the unknowns. A system of equations are homogeneous if each b_i in B (EQN 2) is equal to zero. The Gaussian elimination method will provide a unique solution to equation (3) when the corresponding homogenous system has only the solution X = 0. Both systems AX = B and AX = 0 as well as the coefficient matrix A are then termed non-singular. When AX = 0 has solutions other than zero, the two systems and matrix A are termed singular. This results in AX = B either having no solution or an infinite number of solutions.

8.4.6 SIMULT Subroutine Variable Names

Variables	Description	Dimensions
A	N*N Matrix of Coefficients	--
B	N matrix of constants	--
BIGA	Largest element	--
IA,IB,IC,IJ IMAX,IQS,IT IX,IXJ,IXJX, I1,I2,J,JJ,JJX, JX,JY,K,NY	Integer counters	--
N	Number of unknowns	--
SAVE	Temporary storage location	--
TOL	Tolerance	--

8.4.7 Subroutine Listing

```
      SUBROUTINE SIMULT(A,B,N,KS)
C**** REVISED JANUARY 3,1974 ****
      DIMENSION A(1),B(1)
      TOL=0.0
      KS=0
      JJ=-N
      DO 65 J=1,N
      JY=J+1
      JJ=JJ+N+1
      BIGA=0
      IT=JJ-J
      DO 30 I=J,N
      IJ=IT+I
      IF(ABS(BIGA)-ABS(A(IJ)))20,30,30
   20 BIGA=A(IJ)
      IMAX=I
   30 CONTINUE
      IF(ABS(BIGA)-TOL)35,35,40
   35 KS=1
      RETURN
   40 I1=J+N*(J-2)
      IT=IMAX-J
      DO 50 K=J,N
      I1=I1+N
      I2=I1+IT
      SAVE=A(I1)
      A(I1)=A(I2)
      A(I2)=SAVE
   50 A(I1)=A(I1)/BIGA
      SAVE=B(IMAX)
      B(IMAX)=B(J)
      B(J)=SAVE/BIGA
      IF(J-N)55,70,55
   55 IQS=N*(J-1)
      DO 65 IX=JY,N
      IXJ=IQS+IX
      IT=J-IX
      DO 60 JX=JY,N
      IXJX=N*(JX-1)+IX
      JJX=IXJX+IT
   60 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
   65 B(IX)=B(IX)-(B(J)*A(IXJ))
   70 NY=N-1
      IT=N*N
      DO 80 J=1,NY
      IA=IT-J
      IB=N-J
      IC=N
      DO 80 K=1,J
      B(IB)=B(IB)-A(IA)*B(IC)
      IA=IA-N
   80 IC=IC-1
      RETURN
      END
```

8.5 TGAUSS SUBROUTINE

TGAUSS is a subroutine for in-core solution of large, sparse systems of linear equations (Reference 9.8). The subprogram employs minimum row minimum column elimination. A limited number of zeros is stored and trivial arithmetic is used to preserve computer storage and to reduce the time required for solution. TGAUSS is used in conjunction with TCALC to obtain the system flows and pressures.

8.5.1 Solution Method

Excellent discussions on the Gauss-Jordan elimination technique can be found in many numerical analysis textbooks. Briefly the method is based on the three elementary row operations:

1. Interchange of any two rows.
2. Multiplication of a row by a scalar.
3. Addition of a multiple of one row to another row.

For example by applying a sequence of row transformations to a system of simultaneous equations

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m = b_2$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mm}x_m = b_m$$

Expressed in augmented form

$$\left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdot & a_{1m} & b_1 \\ a_{21} & a_{22} & \cdot & a_{2m} & b_2 \\ a_{m1} & a_{m2} & \cdot & a_{mm} & b_m \end{array}\right]$$

Yields

$$\left[I \,\vdots\, X \right]$$

where I is the identity matrix and X is the solution. The Gauss elimination technique used in TGAUSS requires elimination of only the elements in the upper or lower triangular partition of the array which is followed by a back substitution to obtain the solution.

Two arrays are generated that contain the number of non-zero elements in each row (IRENT()) and the number of non-zero elements in each column (ICENT()) of an N x N array. These arrays are updated each time an element is eliminated or generated, so that the current row and column count are available for pivot selection.

In TGAUSS the IRENT array is searched to find the row with the least number of non-zero coefficients that has not been previously selected as the pivotal row. Should two or more rows satisfy this criteria, the row with the smallest row index is selected. Next the ICENT array is searched to select the column with least number of entries. In the event that two or more columns contain the same number of elements, the column with the smallest index is selected as the pivotal column.

Each row-column selection is thus used in the back substitution to obtain the solution.

8.5.2 Assumptions - Not applicable.

8.5.3 Computation - Upon entry into TGAUSS the number of non-zero elements in each column and row of the M x M solution matrix is stored in ICENT( ) and IRENT( ). At this point the remainder of the program is contained within three nested loops. The outer loop selects a new pivotal element on each pass. The pivot element is stored in the order array for future use during subsequent iterations in the TCALL program. This is a time saving device to eliminate the necessity of selecting the same sequence of pivot elements on each iteration. Once the pivotal element has been selected, the pivotal row is normalized by dividing the row by the pivotal element. Since the pivotal element is normalized, it is set to one as a precaution against round-off errors.

The second loop is entered, which involves a row-by-row search for rows containing elements in the pivotal column. If the number of entries in the pivotal column has been reduced to one entry, there is no need to continue and the program selects a new pivotal element. Also if the pivotal row is selected all further tests are bypassed and the next row is selected since operations on the pivotal row are not permitted.

Finally, the inner loop is a column-by-column search of each row to determine if the row contains the pivotal element. At this point, there are three alternatives available:

1. If the column index in the row being searched is less than the pivotal column, it is necessary to continue searching the row.

2. If the column index is greater than the pivotal column, the row does not contain the pivotal column and a new row must be selected.

3. If the column index is equal to the pivotal column, the row contains the pivotal element and the row can be operated on by the pivotal row.

If the conditions in 3 are met, the pivotal row is multiplied by the negative of the element in the pivotal column of the row being operated on. Then the two rows are added. The element being eliminated is simply dropped from consideration by moving all entries to its right one space to the left. All elements remaining in the row are compared to ZTEST to see if any elements other than the element in the pivotal column were eliminated. If so, the row was further compressed to eliminate the zero entry from the row. Finally, the row is tested to see if the row count is zero which indicates a singularity. If a singularity is encountered an error message is printed:

* SINGULAR MATRIX-NO SOLUTION*

If a singularity is not encountered, the program continues looping until a pivotal element has been selected from each row and column, at this point, the solution vector is stored in the CALC2 array in a scrambled order. The solution is then unscrambled and stored in numerical order in the first column of the A( ) array.

8.5.4 Approximations - In situations where it is known that an operation will result in a zero or a one, the arithmetic operation is bypassed and the element simply set to zero or one.

8.5.5 Limitations - TGAUSS is set up to solve only sparse symmetric systems of linear equations.

8.5.6 TGAUSS Variable Listing

Variable	Description	Dimension
A( )	Matrix of coefficients	--
ATEST	Dummy variable	--
CALC2( )	NU matrix of constants	--
IC,II,IK	Dummy variables	--
ICENT( )	Array containing number of non-zero elements in each column of A( )	--
IORDER( )	Array giving pivot selection based on min-row min column criteria	--
IRENT( )	Array containing number of non-zero elements in each row of A( )	--
ITER	Iteration count	--
IX,IY,J,JKL, JKOP,JKPI,LKJ, NAA,NK	Dummy variables	--
NU	Number of equations	--
OPROW	Dummy variable	--

Variable	Description	Dimension
PIVCOL	Pivot column	--
PIVROW	Pivot row	--
X,ZTEST	Dummy variables	--

8.5.7 Subroutine Listing

```
      SUBROUTINE TGAUSS(NU,ITER)
      DOUBLE PRECISION C,X,A,CALC2
      INTEGER PIVROW,PIVCOL,OPROW
      COMMON G(90),CALC2(55),JPCOL(55,20),A(55,20)
      COMMON/ICC/ICOL(55,20),JRENT(55),JCENT(55)
      DIMENSION ICENT(100),IRENT(100),IORDER(100,3)
      ZTEST=0.0
      NAA=20
C      BUILD IRENT AND ICENT
      DO 99 I=1,NU
      IRENT(I)=JRENT(I)
      ICENT(I)=JCENT(I)
 99   CONTINUE
C     WRITE(6,9010)(IRENT(I),I=1,100)
C     WRITE(6,9010)(ICENT(I),I=1,100)
 9010 FORMAT(1X,20I6)
C     IF(ITER.GT.1)WRITE(6,9020)((IORDER(I,J),J=1,3),I=1,NU)
 9020 FORMAT(1X,6I10)
      DO 66 LKJ=1,NU
      IF(ITER.NE.1) GO TO 432
      IK=100000
      DO 103 I=1,NU
      IF(IRENT(I).GE.IK.OR.IRENT(I).LE.0)GO TO 103
      PIVROW=I
      IK=IRENT(I)
 103  CONTINUE
      IORDER(LKJ,1)=PIVROW
      IK=100000
      IC=IRENT(PIVROW)
      DO 104 I=1,IC
      II=ICOL(PIVROW,I)
      IF(ICENT(II).GE.IK.OR.ICENT(II).LE.0)GO TO 104
      PIVCOL=II
      IK=ICENT(II)
      IY=I
 104  CONTINUE
      IORDER(LKJ,2)=PIVCOL
      IORDER(LKJ,3)=IY
      GO TO 450
 432  PIVROW=IORDER(LKJ,1)
      PIVCOL=IORDER(LKJ,2)
      IY=IORDER(LKJ,3)
 450  X=A(PIVROW,IY)
      IC=IRENT(PIVROW)
      DO 5 J=1,IC
 5    A(PIVROW,J)=A(PIVROW,J)/X
      A(PIVROW,IY)=1.0
      CALC2(PIVROW)=CALC2(PIVROW)/X
      DO 106 I=1,NU
      IF(ICENT(PIVCOL).EQ.1)GO TO 107
```

8.5.7 (Continued)

```
      IF(I.EQ.PIVROW) GO TO 106
      IC=IABS(IRENT(I))
      DO 105 J=1,IC
      IF(ICOL(I,J)-PIVCOL) 105,77,106
 77   OPROW=I
      JKOP=1
      JKPI=1
      C=-A(OPROW,J)
      X=CALC2(PIVROW)*C+CALC2(OPROW)
       CALC2(OPROW)=X
 79   CONTINUE
      IF(ICOL(PIVROW,JKPI).EQ.0) GO TO 106
      IF(ICOL(OPROW,JKOP).EQ.0) GO TO 80
      IF(ICOL(PIVROW,JKPI)-ICOL(OPROW,JKOP)) 80,81,82
 80   IRENT(I)=IRENT(I)+1
      IF(IRENT(I).LE.0) IRENT(I)=IRENT(I)-2
      II=IABS(IRENT(I))
      IF(II.GT.NAA)WRITE(6,9000)II
9000  FORMAT(10X,*EXCEEDED MAX COLUMN NUMBER*,I10)
      IF(II.GT.NAA)STOP
      JKL=JKOP+1
 90   CONTINUE
      IX=II-1
      A(OPROW,II)=A(OPROW,IX)
      ICOL(OPROW,II)=ICOL(OPROW,IX)
      II=IX
      IF(II.GE.JKL) GO TO 90
      X=A(PIVROW,JKPI)*C
      A(OPROW,JKOP)=X
      ICOL(OPROW,JKOP)=ICOL(PIVROW,JKPI)
      IX=ICOL(OPROW,JKOP)
      ICENT(IX)=ICENT(IX)+1
      GO TO 83
 81   IX=ICOL(OPROW,JKOP)
      IF(IX.EQ.PIVCOL) GO TO 11
      X=A(PIVROW,JKPI)*C+A(OPROW,JKOP)
      A(OPROW,JKOP)=X
      ATEST=DABS(X)-ZTEST
      IF(ATEST.GT.0.0)GO TO 83
 11   ICENT(IX)=ICENT(IX)-1
      IRENT(OPROW)=IRENT(OPROW)-1
      IF(IRENT(OPROW))140,141,142
 141  CONTINUE
      WRITE(6,9030)
9030  FORMAT(10X,*SINGULAR MATRIX-NO SOLUTION*)
      STOP
 140  CONTINUE
      IRENT(OPROW)=IRENT(OPROW)+2
 142  IX=IABS(IRENT(OPROW))
      DO 181 NK=JKOP,IX
```

8.5.7 (Continued)

```
      A(I,NK)=A(I,NK+1)
 181  ICOL(I,NK)=ICOL(I,NK+1)
      IX=IX+1
      ICOL(I,IX)=0
      JKPI=JKPI+1
      GO TO 79
 83   JKPI=JKPI+1
 82   JKOP=JKOP+1
      GO TO 79
 105  CONTINUE
 106  CONTINUE
 107  CONTINUE
      IRENT(PIVROW)=-IRENT(PIVROW)
      ICENT(PIVCOL)=-ICENT(PIVCOL)
 66   CONTINUE
      DO 350 I=1,NU
      II=ICOL(I,1)
 350  A(II,1)=CALC2(I)
C     WRITE(6,9040)(CALC2(I),I=1,NU)
 9040 FORMAT(1X,10E12.5)
      RETURN
      END
```

8.6 FUNCTION UFW

Function UFW is a heat transfer coefficient calculation subroutine. The heat transfer coefficient between a wall and a fluid is calculated based on the volume flow rate, pressure, temperature, cross sectional area of the fluid and the distance over which the fluid flows.

8.6.1 Math Model - The function routine UFW is called by several of the subroutines listed previously. The function is called from each subroutine in the form

```
UFWIL = UFW(AAA,DDD,Q(LI),TF(LI),P(LI))
```

AAA is equal to the cross sectional area of the fluid at the location where the heat transfer coefficient is of interest. DDD is the diameter of the orifice at the previous cross section. Q(LI) is the volume flow rate also at that section; TF(LI) is the temperature of the fluid and P(LI) is the pressure in the fluid at that point.

Once UFW is called from a subroutine the function UFW is executed and calculates the heat transfer coefficient as follows:

The Reynolds number is calculated

```
REN = DDD*ABS(FLOW)/(VISCIL*AAA)
```

where flow is equal to Q(LI), the volume flow rate, and VISCIL is the fluid viscosity.

The Prandtt number is calculated

```
PRN = VISCIL*386.4*RHOIL(TEMP,PRESS)*CPFN/CF
```

where 386.4 is just a conversion factor for the density from Hydraulic units to thermal units of lb./in.^{3}.

A check on the Reynolds number is made to see if turbulent or laminar flow exists.

```
IF(REN.LT. 1200.) go to 1000
```

If laminar or less than 1200 the heat transfer coefficient is calculated

UFW = 4.364*CF/DDD

But if turbulent then the heat transfer coefficient is calculated as

UFW=0.0118*(PRN**.3)*(REN**.9)*CF/DDD

The heat transfer coefficient is then returned to the subroutine which made the call.

8.6.2 Variable Listing

Variable	Description	Dimension
AAA	Fluid cross sectional area	$IN.^2$
DDD	Diameter of the cross sectional area	IN.
FLOW	Volume flow rate in the considered section	CIS
PRESS	Fluid pressure	PSI
PRN	Prandtl number	--
REN	Reynolds number	--
UFW	Heat transfer coefficient, fluid to wall	$WATTS/IN^2$-°F
VISCIL	Fluid viscosity	IN^2/SEC.

8.6.3 <u>Subroutine Listing</u>

```
      FUNCTION UFW(AAA,DDD,FLOW,TEMP PRESS)
C  **********  CACULATE HEAT TRANSFER COEFFICIENT  *********
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON/LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),NLSEG(150)
      COMMON/FLUID/ATPRES,CF,CPFN,FTEMP,PROP(13,3)
      VISCIL=VISC(TEMP,PRESS)
      REN=DDD*ABS(FLOW)/(VISCIL*AAA)
      PRN=VISCIL*386.4*RHO(TEMP,PRESS)*CPFN/CF
      IF(REN.LT.1200.) GO TO 1000
      UFW=0.0118*(PRN**.3)*(REN**.9)*CF/DDD
      RETURN
 1000 UFW=4.364*CF/DDD
      RETURN
      END
```

8.7 SUBROUTINE XLIMIT

XLIMIT is a utility subroutine which provides the calling program with information to determine if a limit has been reached. The subroutine is typically used for components with mechanical movement and returns position and velocity data.

8.7.1 MATH MODEL

Not applicable.

8.7.2 ASSUMPTIONS

Not applicable.

8.7.3 COMPUTATION METHOD

Minimum (POSMIN) and maximum (POSMAX) limits are input along with the current values of position (POS) and velocity (VEL) from the calling program. Initially the sign is set to zero and the position is compared against POSMAX.

If POS is greater than or equal to POSMAX, the position is set to POSMAX and ASIGN is set to 1. Should VEL be greater than zero it is zeroed and a return is made to the calling program.

If POS is less than POSMAX it is checked against POSMIN. When POS is less than or equal to POSMIN, POS is set to POSMIN, ASIGN equals -1 and the velocity is zeroed if it is less than zero.

Should POS be greater than POSMIN a return is made to the calling program without any position or velocity changes.

8.7.4 APPROXIMATIONS

Not applicable.

8.7.5 LIMITATIONS

Not applicable.

8.7.6 VARIABLE NAMES

Variable	Description
ASIGN	Sign (-1, 0, 1)
POS	Position
POSMIN	Minimum Position
POSMAX	Maximum Position
VEL	Velocity

8.7.7 Subroutine Listing

```
      SUBROUTINE XLIMIT(POS,VEL,ASIGN,POSMIN,POSMAX)
      ASIGN=0.0
      IF(POS-POSMAX) 20,10,10
   10 POS=POSMAX
      ASIGN=1.0
      IF(VEL.GT.0.0) GO TO 40
      GO TO 50
   20 IF(POS-POSMIN) 30,30,50
   30 POS=POSMIN
      ASIGN=-1.0
      IF(VEL.GE.0.0) GO TO 50
   40 VEL=0.0
   50 RETURN
      END
```

8.8 FUNCTION FRIC

FRIC is a function subroutine that is used to calculate steady state flow-pressure drop coefficients for laminar and turbulent flows.

8.8.1 Math Model - Steady state pressure drops are computed using the Darcy-Weisbach equation. For laminar flow

```
FRIC = 128./PI*VISC( )*RHO( )/(PARM(KNEL,2)**4)
       *(PARM(KNEL,1) + PARM(KNEL,3))
```

For turbulent flow

```
FRIC = .213*RHO( )*(VISC( )**.25)/(PARM(KNEL,2)**4.75)
       *(PARM(KNEL,1)+PARM(KNEL,3))
```

where

PARM(KNEL,1) = Line length (IN)

PARM(KNEL,2) = inside line diameter (IN)

PARM(KNEL,3) = equivalent line length (IN)

8.8.2 Assumptions -

1. Transition from laminar to turbulent flow is assumed to occur at a Reynolds number of 1200. Flows having a Reynolds number greater than 1200 are considered turbulent while flows having a Reynolds of 1200 or less are assumed laminar.

2. The friction factors used are based on circular cross-section, smooth I.D., drawn tubing.

8.8.3 Computation Methods - Function FRIC arguments are

KNEL - Line number

TEMP - Line temperature (°F)

PRESS - Line pressure (PSI)

The function returns either a laminar or turbulent coefficient for use in a flow-pressure drop equation.

8.8.4 Approximations - Pressure drops calculated for flows in the Reynolds number range of 1200 to 3000 are approximate since a transition flow equation was not developed. The turbulent equation is used in this range.

8.8.5 Limitations - FRIC should not be used to calculate pressure drops across non-circular cross section passages or across rough I.D. tubing.

8.8.6 Subroutine Listing

```
      FUNCTION FRIC(KNEL,TEMP,PRESS)
C *** REVISED AUGUST 15, 1976 ***
      COMMON /TRANS/P(300),Q(300),C(300),TC(300),TW(300),TF(300),
     + ACF(300),ACW(300),DXF(300),TIME,DELT,PI,NLINE,NEL
      COMMON /LINE/PARM(150,4),TLW(2000),TLF(2000),LSTART(150),
     + NLSEG(150)
      ENTRY FRICL
      FRIC=128./PI*VISC(TEMP,PRESS)*RHO(TEMP,PRESS)/(PARM(KNEL,2)**4.)
     + *(PARM(KNEL,1)+PARM(KNEL,3))
      RETURN
      ENTRY FRICT
      FRIC=.213*RHO(TEMP,PRESS)*(VISC(TEMP,PRESS)**.25)/(PARM(KNEL,2)
     + **4.75)*(PARM(KNEL,1)+PARM(KNEL,3))
      RETURN
      END
```

9.0 REFERENCES

1. Kreith, Frank, "Principals of Heat Transfer" 3rd Edition, Intext Press Inc., pg. 244

2. Memo E242-659, 17 March 1976, HEATRAN Procedure for Computing Temperature Rise of Hydraulic System Oil Passing through Pump and Servo Valve/Actuator

3. Kays, W. M., "Convective Heat and Mass Transfer", McGraw Hill, 1966

4. Fermi, Enrico "Thermodynamics", Dover Publications, 1936

5. Streeter, Victor L., "Fluid Mechanics," McGraw Hill, 1951

6. Cambel, A,B., Jennings, B.H., "Gas Dynamics", McGraw Hill, 1958

7. Report GH20-0205-4 "System/360 Scientific Subroutine Package - Version III, Programmer's Manual," IBM Corp., August 1970

8. Key, John E., "Computer Program for Solution of Large, Sparse, Unsymmetric Systems of Linear Equations," International Journal for Numerical Methods in Engineering, Vol. 6, Pgs. 497-509 (1973).