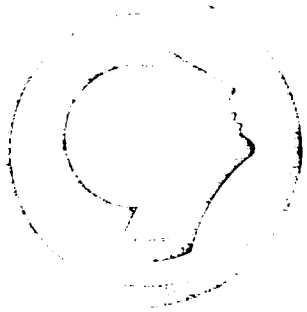
D.O. 123
NAS 8-36955



Users Manual for Program ADMIT

Admittance and Pressure Transfer Function

Developed for Use on a PC Computer

rch Institute
niversity of Alabama in Huntsville

105

Users Manual for Program ADMIT

Admittance and Pressure Transfer Function

Developed for Use on a PC Computer

bur C. Armstrong

J 1992

Table of Contents

1.0 Introduction

The piping in a liquid rocket can assume complex configurations due to multiple tanks, multiple engines, and structures that must be piped around. The capability to handle some of these complex configurations have been incorporated into the ADMIT code. The capability to modify the input on line has been implemented.

The configurations allowed include multiple tanks, multiple engines, the splitting of a pipe into unequal segments going to different (or the same) engines. This program will handle the following type elements

Straight pipes
Bends
Inline accumulators
Tuned stub accumulators
Helmholtz resonators
Parallel resonators
Pumps
Split pipes
Multiple tanks
Multiple engines

2.0 Input Description

ADMIT uses the following files: ENG.RLN, LOX.RLN, and FUEL.RLN. All files are in free format, therefore each of the following records will give the same results.

```
Record 1:   1.000000E-01 6219.000000 2.670000 2.330E-03 -315.0000

Record 2:   0.1  6219.0  2.67  0.00233 -315.0

Record 3:   1.E-01
            6219.0
            2.67
            2.33E-3
           -315.0
```

The file assignments are given in the following table:

Unit	File Name	File Type	Description
9	ENG.RLN	Input	Engine data
10	LOX.RLN	Input	LOX tanks & lines data
11	FUEL.RLN	Input	Fuel tanks & lines data
13	SURF.ERR	Output	Convergence error information
14	SURF.OUT	Output	Admittance values
15	(LOX)	Work	Temporary file with LOX data
16	(FUEL)	Work	Temporary file with fuel data
17	(RESULT)	Work	Temporary file for results

2.1 Description of file ENG.RLN

```
Card # 1
   number of engines
 Card # 2
    total flow in engine (lbm/sec),
    chamber pressure (lbf/ft^2),
    pressure drop across orifice (lbf/ft^2)
 Read card # 2 "number of engines" times
```

2.2 Description of files LOX.RLN or FUEL.RLN

```
Card # 1
   title
Card # 2
   number of tanks
 Card # 3
    volume (ft^3),
    mass flow (lbm/sec),
    bulk modulus (lbf/ft^2),
    density (lbm/ft^3)
 Read card # 3 "number of tanks" times
```

```
Card # 4
   number of lines leaving tank
 Card # 5
    tank number,
    engine number (0 if split follows)
 Card # 6
    number of segments,
    number of unique splits
  Card # 7
     section type,
     pipe1,
     pipe2,
     pipe3,
     pipe4,
     pipe5
  Read card # 7 "number of segments" times
if split > 0
  Card # 8
     number of segments,
     number of identical lines,
     engine number
   Card # 9
      section type,
      pipe1,
      pipe2,
      pipe3,
      pipe4,
      pipe5
   Read card # 9 "number of segments" times
  Read card # 8-9 "number of splits" times
 Read card # 5-9 "number of lines" times
```

where

type	name	PIPE1	PIPE2	PIPE3	PIPE4	PIPE5
0	bend	radius	angle	diameter	end len.	
1	straight	length	diameter			
2	inline	length	diameter			
3	tuned	length	diameter			
4	Helmholtz	length	diameter	volume		
5	parallel	length	diameter	volume		
6	pump	length	diameter	$dp/d\mathring{m}$	L	C
7	manifold	volume	bulk mod.			

Dimensions:

radius, length, diameter, end length	- ft
angle	- deg
volume	- ft^3
$\dot{m}/\bar{p}_c$ (non-dimensionalized by $\bar{\dot{m}}/\bar{p}_c$)	- non-dimensional
L	- sec
C	- sec
bulk modulus	- lbf/ft^2

3.0 Output Description

3.1 Output Files

Output from the program is a file (SURF.OUT) which may be printed and various graphs under the control of the user. The print file contains the following:

Title, time, and date

LINE	ENG.	FREQ	|G|	G(R)	G(I)
line no.	engine no.	frequency	amplitude of admittance looking toward tank	real part	complex part

Also, if a split pipe is analyzed, a file (SURF.ERR) is created if any point fails to converge within the specified number of iterations. This file contains:

Title, time, and date

jw = after iterations has error of %

I= J= |G|= |GOLD|=

3.2 Graphs Available

The graphs available are

1. Plot of selected line on upper half of screen and admittance looking toward tank on lower half.
2. Surface plot of pressure transfer function vs frequency and distance along pipe. The surface may be a wire frame or a solid surface. The view angle of the surface may be changed by the user.
3. Contour plot of the pressure transfer function vs frequency and distance along pipe. Nine levels are plotted with levels 1, 5, and 9 emphasized and values printed at the side of the graph.

The color scheme of each of the graph types may be changed by the user.

4.0 Sample Run

The sample run consists of two lox tanks and four engines, two of the engines and lines going to them are identical. The total mass flow from each tank is the same, however the line from the first tank is split with half the mass flow going to engine # 2 and the other half split into two identical engines # 1.

4.1 Input for Sample Run

Sample ENG.RLN file:

```
3
853.5       4.502040E+05      1.610532E+06
1707.0      4.502040E+05      1.610532E+06
3414.0      4.502040E+05      1.610532E+06
```

Sample LOX.RLN file:

```
Sample Run
2
1.956300E+04   2928.0   1.185883E+07   71.4
1.956300E+04   2928.0   1.185883E+07   71.4
2
1 0
13 2
1   15.0      1.416    0.0      0.0    0.0
0   35.0      45.0     1.416    0.0    0.0
1   30.0      1.416    0.0      0.0    0.0
0   3.5       135.0    1.416    0.0    0.0
1   15.0      1.416    0.0      0.0    0.0
1   20.641    1.416    0.0      0.0    0.0
1   20.558    1.416    0.0      0.0    0.0
1   20.558    1.416    0.0      0.0    0.0
1   8.541     1.416    0.0      0.0    0.0
1   6.383     1.416    0.0      0.0    0.0
0   4.25      90.0     1.416    0.0    0.0
1   9.33      1.416    0.0      0.0    0.0
0   3.33      80.0     1.416    0.0    0.0
5   2  1
1   3.53      0.708    0.0      0.0    0.0
1   12.2      0.708    0.0      0.0    0.0
0   1.28      35.0     0.708    0.0    0.0
1   12.2      0.708    0.0      0.0    0.0
7   13.5   1.183346E+07 0.0      0.0    0.0
5   1 2
1   3.53      1.00126  0.0      0.0    0.0
1   12.2      1.00126  0.0      0.0    0.0
0   1.28      35.0     1.00126  0.0    0.0
1   12.2      1.00126  0.0      0.0    0.0
7   13.5   1.183346E+07 0.0      0.0    0.0
```

```
2 3
18 0
1   15.0      1.416      0.0      0.0    0.0
0   35.0      45.0       1.416    0.0    0.0
1   30.0      1.416      0.0      0.0    0.0
0   3.5       135.0      1.416    0.0    0.0
1   15.0      1.416      0.0      0.0    0.0
1   20.641    1.416      0.0      0.0    0.0
1   20.558    1.416      0.0      0.0    0.0
1   20.558    1.416      0.0      0.0    0.0
1   8.541     1.416      0.0      0.0    0.0
1   6.383     1.416      0.0      0.0    0.0
0   4.25      90.0       1.416    0.0    0.0
1   9.33      1.416      0.0      0.0    0.0
0   3.33      80.0       1.416    0.0    0.0
1   3.53      1.416      0.0      0.0    0.0
1   12.2      1.416      0.0      0.0    0.0
0   1.28      35.0       1.416    0.0    0.0
1   12.2      1.416      0.0      0.0    0.0
7   13.5   1.183346E+07  0.0      0.0    0.0
```

4.2 Walkthrough of Sample Run

```
Welcome to ADMIT - a Feedline Analysis Program

To send a plot to the printer

The computer MUST be in GRAPHICS mode

Hit PrScn to send the current plot to the printer
```

```
Maximum no. of frequencies =  90
Maximum points along pipe  =  90

 If you want frequency in rad/sec, hit enter.
If you want it in Hertz, enter "H". h
Is the engine data on file ENG.RLN? (Y/N) y
Is this setup for FUEL or OXIDIZER? Enter F or  O. g
 You did not enter F or O.  Try again
Is this setup for FUEL or OXIDIZER? Enter F or  O. o
Is the lox file name LOX.RLN? (Y/N) y
Max. no. of iterations is set at  20
Do you wish to change it? y
Enter maximum no. of iterations  20
Do you wish to modify lox line data?   y
Do you wish to change tank parameters? n
Do you wish to change the pipe layout? n
```

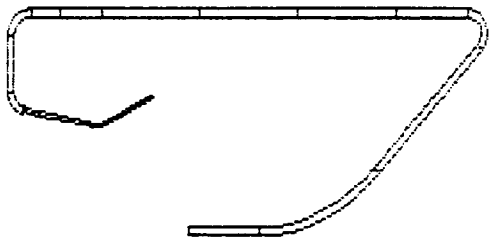
Do you wish to plot piping & admittances? **y**
The following LOX lines may be plotted

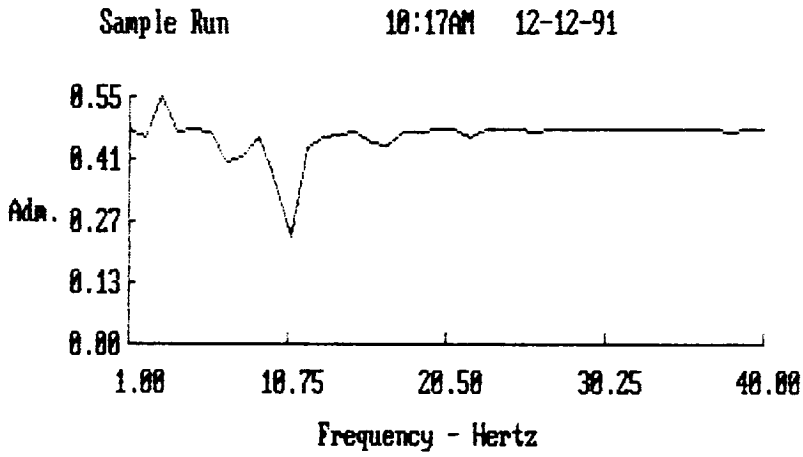
Line #	Tank #	Engine #
1	1	1
2	1	2
3	2	3

Enter line # to be plotted, 0 will end plot **4**
You did not enter a valid line #. Try again

Enter line # to be plotted, 0 will end plot **1**
Do you wish change colors of admittance? **n**

LOX Piping - Tank # 1 Engine # 1



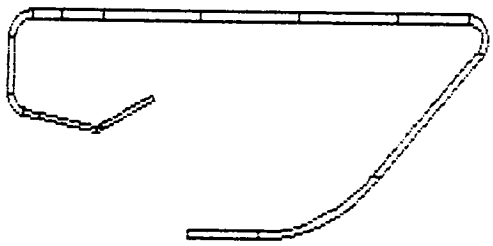


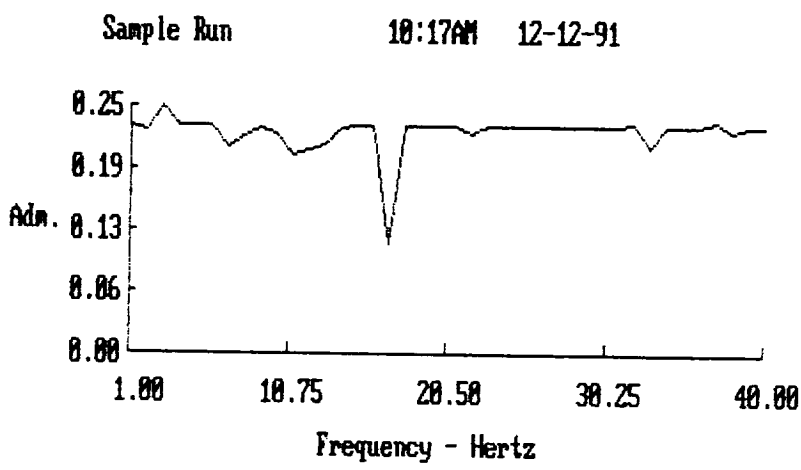
The following LOX lines may be plotted

Line #	Tank #	Engine #
1	1	1
2	1	2
3	2	3

Enter line # to be plotted, 0 will end plot **2**
Do you wish change colors of admittance? **n**

LOX Piping - Tank # 1 Engine # 2



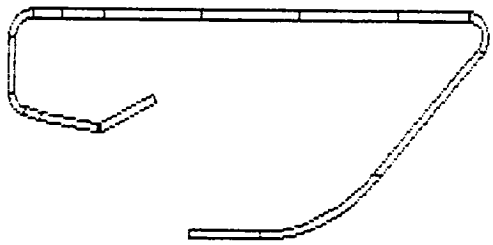


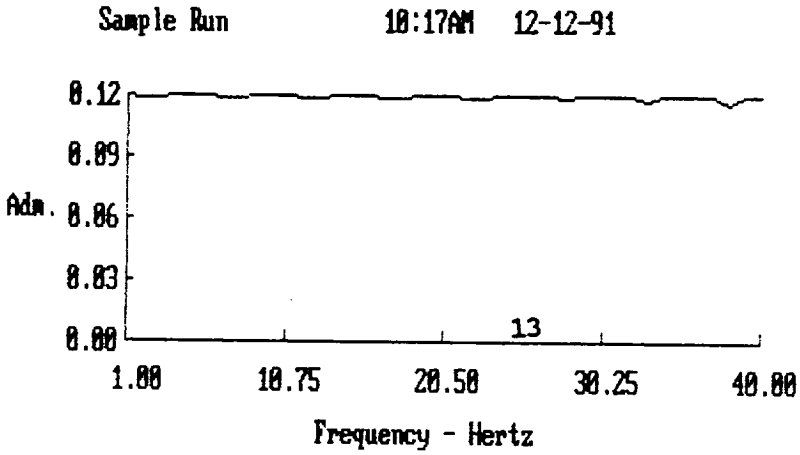
The following LOX lines may be plotted

Line #	Tank #	Engine #
1	1	1
2	1	2
3	2	3

Enter line # to be plotted, 0 will end plot **3**
Do you wish change colors of admittance? **n**

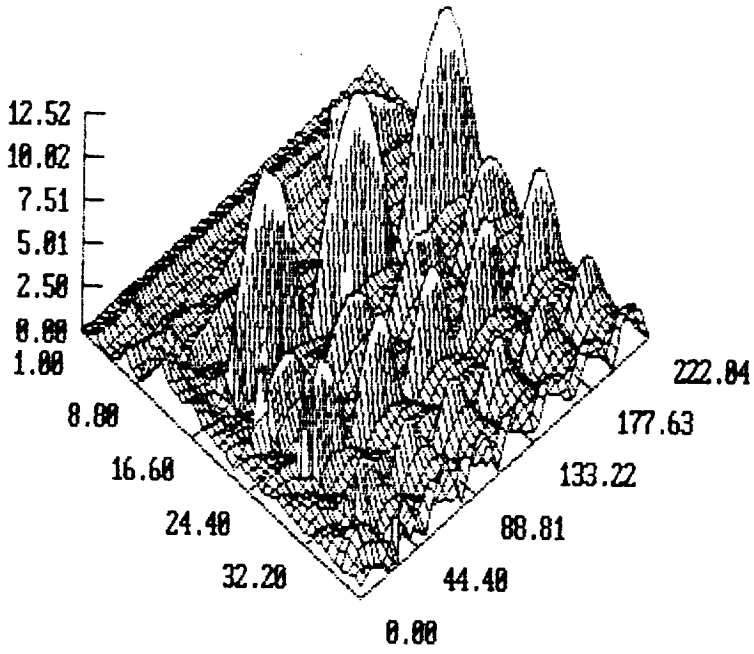
LOX Piping - Tank # 2 Engine # 3





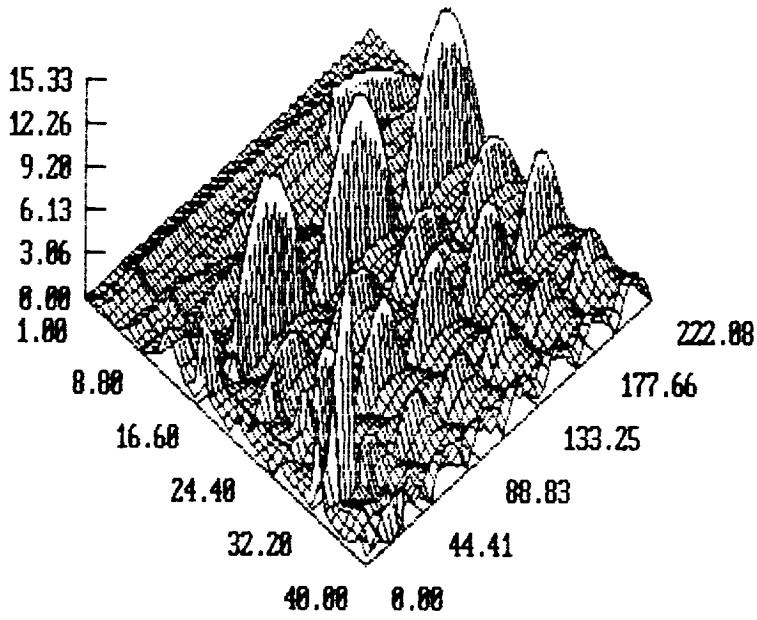
```
 The following LOX lines may be plotted

Line #     Tank #    Engine #

   1          1          1
   2          1          2
   3          2          3

Enter line # to be plotted, 0 will end plot   0
Do you wish to plot surfaces? y
Do you want a wire-frame drawing? y
 The following LOX lines may be plotted

Line #     Tank #    Engine #

   1          1          1
   2          1          2
   3          2          3

Enter line # to be plotted, 0 will end plot   1
```

Sample Run 10:17AM 12-12-91
Pressure Transfer Function = f(freq(Hertz),distance(ft))
LOX Piping - Tank # 1 Engine # 1



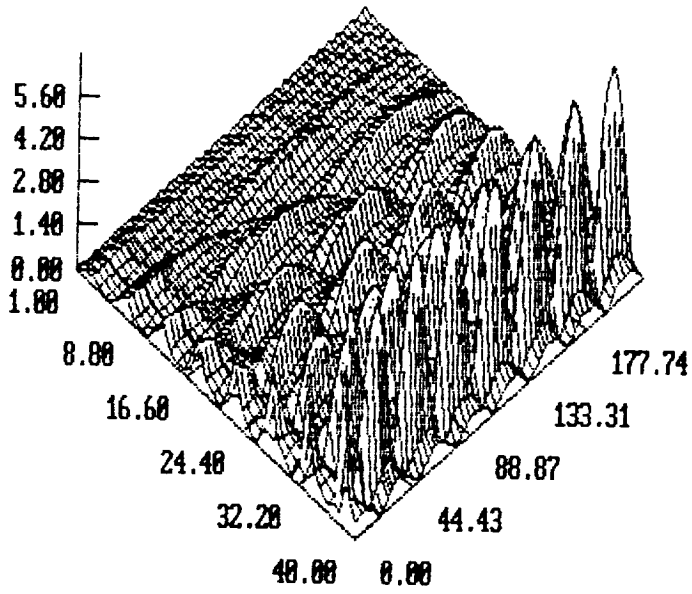
```
Current view is PHI = -45.000   THETA =  30.000
Do you wish another view? n
Current BACKGROUND COLOR =  4 LINE COLOR =  1 FILL COLOR =  3
Do you wish another color? n
Do you want a filled drawing? n
Do you want another surface plot? y
 The following LOX lines may be plotted

Line #    Tank #    Engine #

  1         1         1
  2         1         2
  3         2         3

Enter line # to be plotted, 0 will end plot   2
```

Sample Run 10:17AM 12-12-91
Pressure Transfer Function = f(freq(Hertz),distance(ft))
LOX Piping - Tank # 1 Engine # 2

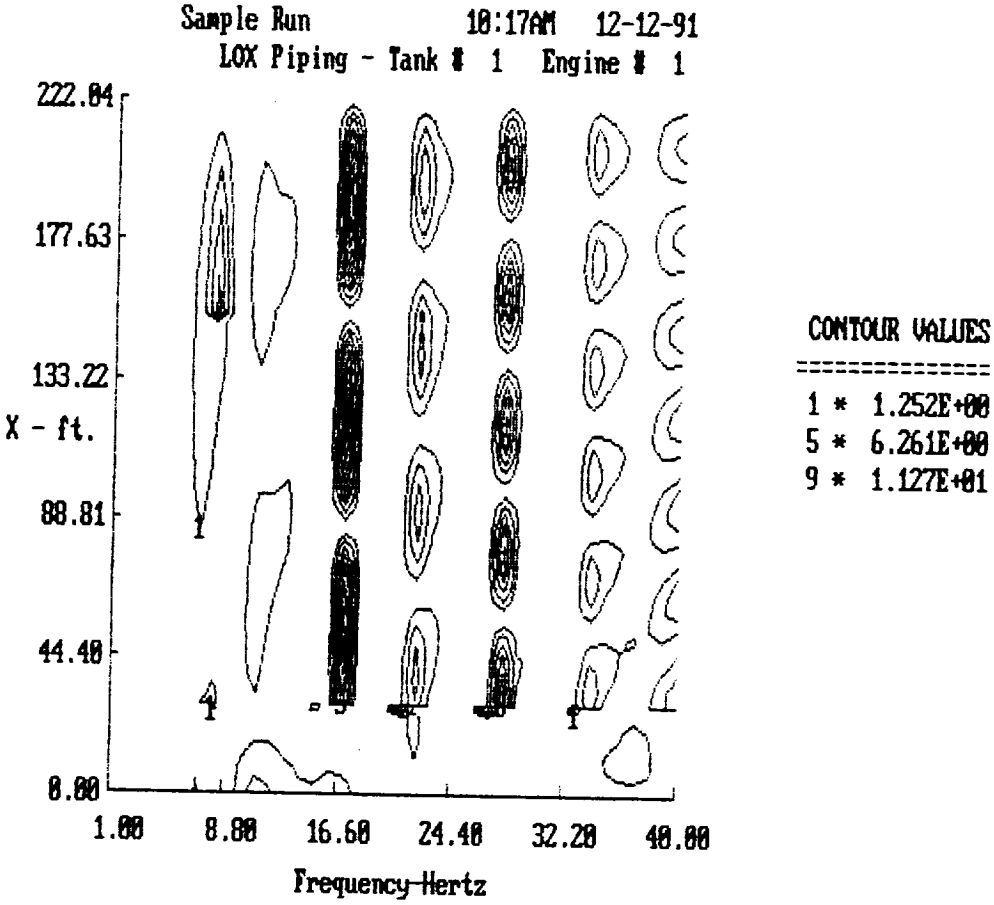


```
Current view is PHI = -45.000   THETA =  30.000
Do you wish another view? n
Current BACKGROUND COLOR =  4 LINE COLOR =  1 FILL COLOR =  3
Do you wish another color? n
Do you want a filled drawing? n
Do you want another surface plot? y
 The following LOX lines may be plotted

Line #     Tank #    Engine #

   1          1          1
   2          1          2
   3          2          3

Enter line # to be plotted, 0 will end plot   3
```

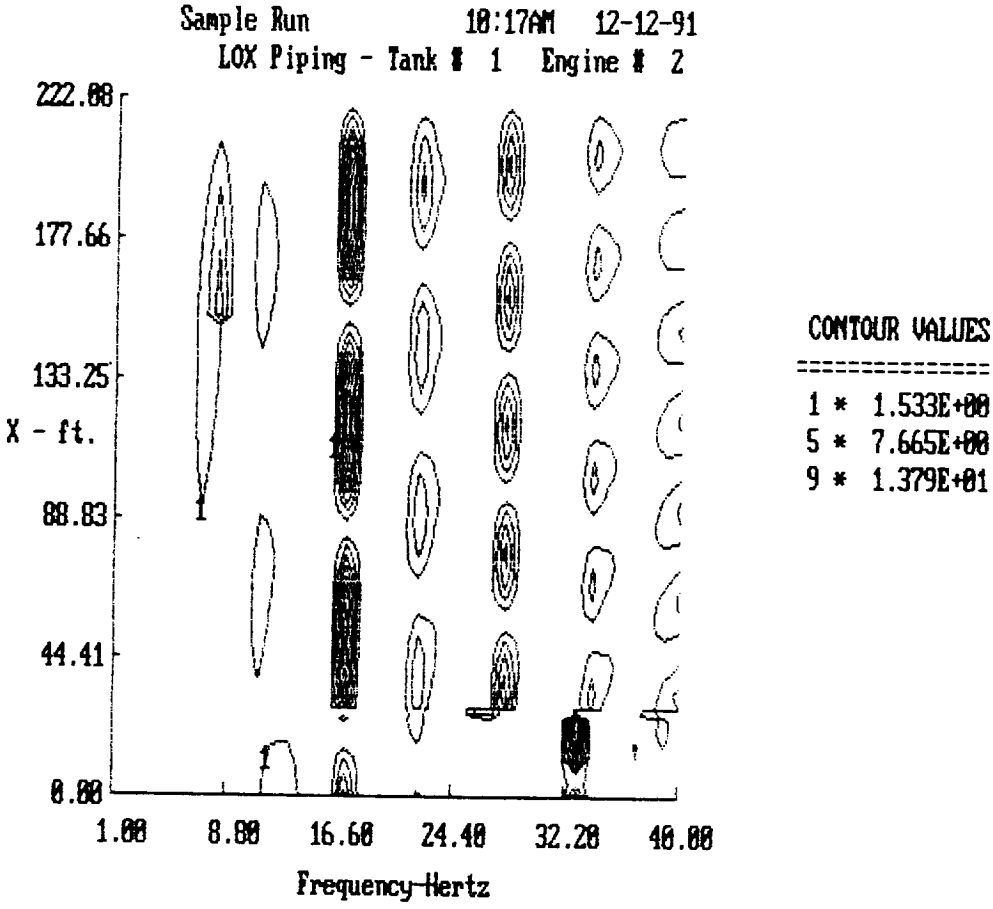
Sample Run 10:17AM 12-12-91
Pressure Transfer Function = f(freq(Hertz),distance(ft))
LOX Piping - Tank # 2 Engine # 3



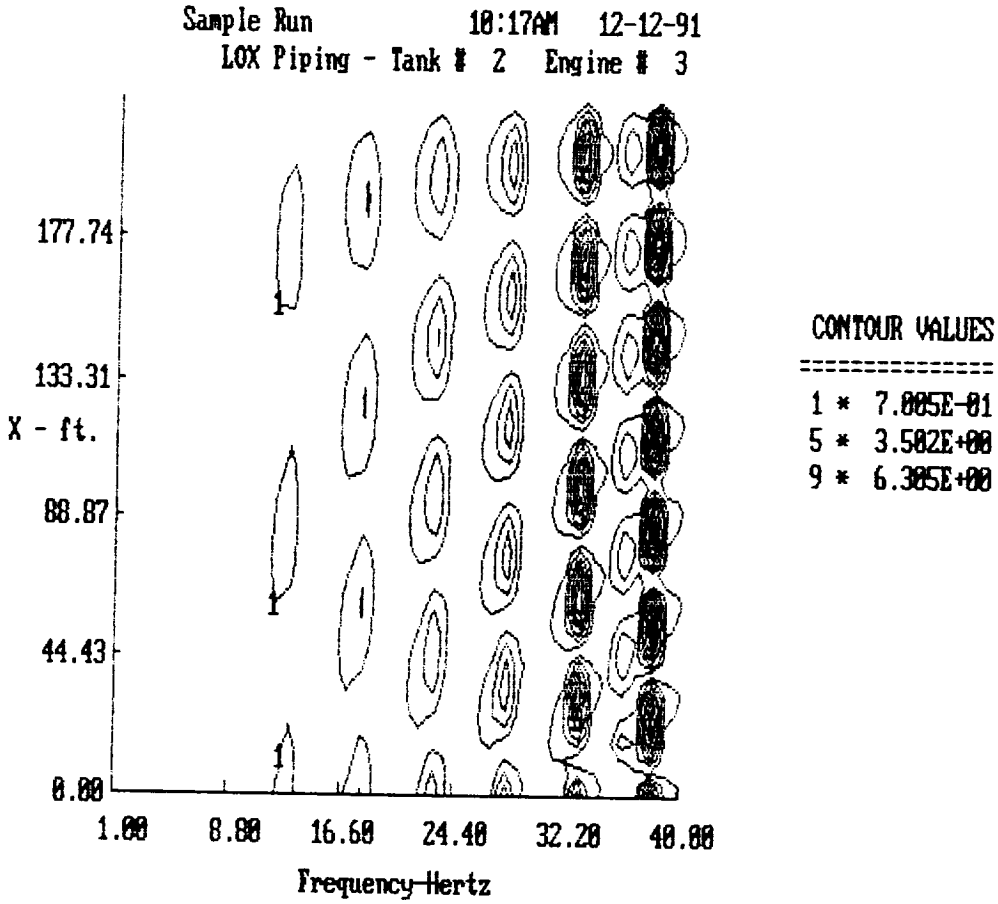
```
Current view is PHI = -45.000   THETA =  30.000
Do you wish another view? n
Current BACKGROUND COLOR =  4 LINE COLOR =  1 FILL COLOR =  3
Do you wish another color? n
Do you want a filled drawing? n
Do you want another surface plot? n
Do you wish to plot contours? y
 The following LOX lines may be plotted

Line #     Tank #    Engine #

   1          1          1
   2          1          2
   3          2          3

Enter line # to be plotted, 0 will end plot   1
```


Sample Run 10:17AM 12-12-91
LOX Piping - Tank # 1 Engine # 1
222.84
177.63
133.22
X - ft.
88.81
44.40
0.00
1.00
8.00
16.60
24.40
32.20
40.00
Frequency-Hertz
CONTOUR VALUES
1 * 1.252E+00
5 * 6.261E+00
9 * 1.127E+01

```
Current BACKGROUND COLOR =  4 LINE COLOR =  1 FILL COLOR =  3
Do you wish another color? n
Do you want another contour plot? y
 The following LOX lines may be plotted

Line #    Tank #   Engine #

   1         1         1
   2         1         2
   3         2         3

Enter line # to be plotted, 0 will end plot   2
```


Sample Run 10:17AM 12-12-91
LOX Piping - Tank # 1 Engine # 2
222.00
177.66
133.25
X - ft.
88.83
44.41
0.00
1.00
8.00
16.60
24.40
32.20
40.00
Frequency-Hertz
CONTOUR VALUES
1 * 1.533E+00
5 * 7.665E+00
9 * 1.379E+01

```
Current BACKGROUND COLOR =  4 LINE COLOR =  1 FILL COLOR =  3
Do you wish another color? n
Do you want another contour plot? y
 The following LOX lines may be plotted

Line #     Tank #    Engine #

   1          1          1
   2          1          2
   3          2          3

Enter line # to be plotted, 0 will end plot   3
```


Sample Run 10:17AM 12-12-91
LOX Piping - Tank # 2 Engine # 3
177.74
133.31
X - ft.
88.87
44.43
0.00
1.00
8.00
16.60
24.40
32.20
40.00
Frequency-Hertz
CONTOUR VALUES
1 * 7.005E-01
5 * 3.502E+00
9 * 6.305E+00

```
Current BACKGROUND COLOR =  4 LINE COLOR =  1 FILL COLOR =  3
Do you wish another color? n
Do you want another contour plot? n
Enter E to exit, F to run new frequency range, or C to run a new case e
```

4.3 Output for Sample Run

SURF.OUT File

Sample Run 10:17AM 12-12-91

LINE	ENG.	FREQ	|G|	G(R)	G(I)
2	1	1.000000E+00	4.788069E-01	4.785361E-01	-1.609942E-02
3	2	1.000000E+00	2.394486E-01	2.393604E-01	-6.496578E-03
4	3	1.000000E+00	1.198275E-01	1.197831E-01	-3.263087E-03
2	1	2.000000E+00	4.657790E-01	4.623671E-01	-5.627374E-02
3	2	2.000000E+00	2.325824E-01	2.311172E-01	-2.606584E-02
4	3	2.000000E+00	1.190449E-01	1.182235E-01	-1.396010E-02
2	1	3.000000E+00	5.503493E-01	5.327964E-01	-1.378852E-01
3	2	3.000000E+00	2.571785E-01	2.552799E-01	-3.119238E-02
4	3	3.000000E+00	1.187674E-01	1.176730E-01	1.608596E-02
2	1	4.000000E+00	4.766055E-01	4.766044E-01	1.030500E-03
3	2	4.000000E+00	2.385123E-01	2.384555E-01	5.208500E-03
4	3	4.000000E+00	1.198004E-01	1.197289E-01	4.137957E-03
2	1	5.000000E+00	4.780528E-01	4.774268E-01	-2.445669E-02
3	2	5.000000E+00	2.393914E-01	2.393727E-01	-2.990401E-03
4	3	5.000000E+00	1.198703E-01	1.198686E-01	6.308582E-04
2	1	6.000000E+00	4.762524E-01	4.730391E-01	-5.523042E-02
3	2	6.000000E+00	2.394595E-01	2.391756E-01	-1.165743E-02
4	3	6.000000E+00	1.198399E-01	1.198079E-01	-2.769017E-03
2	1	7.000000E+00	4.079117E-01	3.857753E-01	-1.325495E-01
3	2	7.000000E+00	2.172124E-01	2.141030E-01	-3.662187E-02
4	3	7.000000E+00	1.191319E-01	1.183965E-01	-1.321685E-02
2	1	8.000000E+00	4.233197E-01	4.232355E-01	-8.441784E-03
3	2	8.000000E+00	2.265927E-01	2.265922E-01	4.792674E-04
4	3	8.000000E+00	1.187229E-01	1.175848E-01	1.639913E-02
2	1	9.000000E+00	4.624276E-01	4.623034E-01	-1.071830E-02
3	2	9.000000E+00	2.366885E-01	2.365803E-01	7.156707E-03
4	3	9.000000E+00	1.197830E-01	1.196941E-01	4.613900E-03
2	1	1.000000E+01	3.672516E-01	3.514259E-01	-1.066469E-01
3	2	1.000000E+01	2.294481E-01	2.286631E-01	-1.896343E-02
4	3	1.000000E+01	1.198659E-01	1.198599E-01	1.200036E-03
2	1	1.100000E+01	2.448379E-01	2.441121E-01	1.883887E-02
3	2	1.100000E+01	2.088115E-01	2.079123E-01	-1.935742E-02
4	3	1.100000E+01	1.198544E-01	1.198369E-01	-2.050537E-03
2	1	1.200000E+01	4.414459E-01	4.064632E-01	1.722271E-01
3	2	1.200000E+01	2.131590E-01	1.895564E-01	-9.749433E-02
4	3	1.200000E+01	1.192847E-01	1.187004E-01	-1.179237E-02
2	1	1.300000E+01	4.622294E-01	4.485251E-01	1.117195E-01
3	2	1.300000E+01	2.189573E-01	2.039270E-01	7.972505E-02
4	3	1.300000E+01	1.187096E-01	1.175585E-01	1.649133E-02

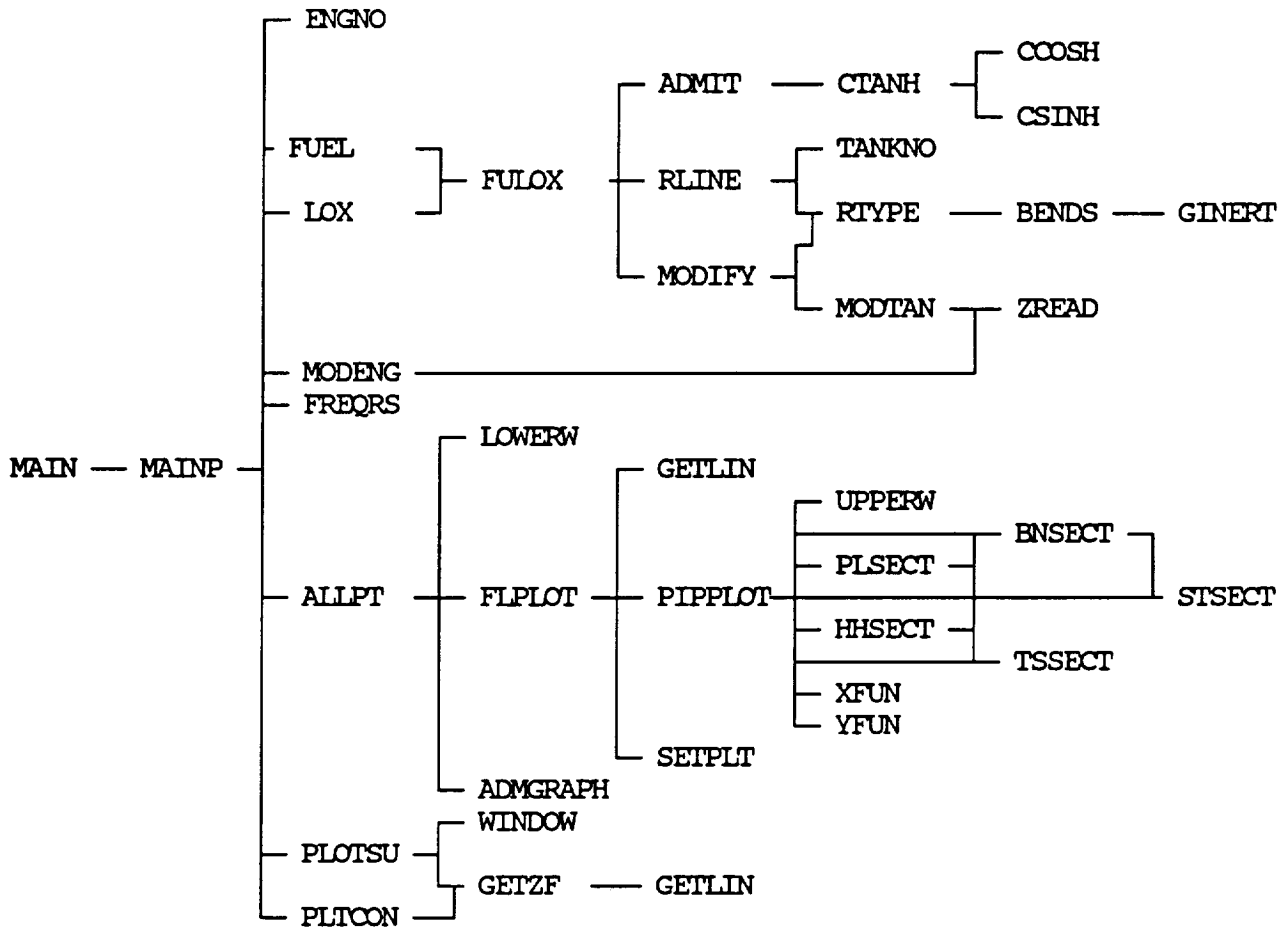
2	1	1.400000E+01	4.713571E-01	4.638380E-01	8.385620E-02
3	2	1.400000E+01	2.354399E-01	2.320770E-01	3.965155E-02
4	3	1.400000E+01	1.197693E-01	1.196667E-01	4.955533E-03
2	1	1.500000E+01	4.731210E-01	4.672862E-01	7.407471E-02
3	2	1.500000E+01	2.383203E-01	2.376733E-01	1.754884E-02
4	3	1.500000E+01	1.198598E-01	1.198476E-01	1.708378E-03
2	1	1.600000E+01	4.523079E-01	4.443953E-01	8.423349E-02
3	2	1.600000E+01	2.380467E-01	2.379751E-01	5.836634E-03
4	3	1.600000E+01	1.198648E-01	1.198577E-01	-1.305985E-03
2	1	1.700000E+01	4.475234E-01	4.409334E-01	7.651746E-02
3	2	1.700000E+01	1.156441E-01	8.863128E-02	7.428367E-02
4	3	1.700000E+01	1.194596E-01	1.190487E-01	-9.899915E-03
2	1	1.800000E+01	4.766612E-01	4.738513E-01	5.167995E-02
3	2	1.800000E+01	2.380264E-01	2.363212E-01	2.843990E-02
4	3	1.800000E+01	1.186682E-01	1.174765E-01	1.677521E-02
2	1	1.900000E+01	4.774443E-01	4.754010E-01	4.412425E-02
3	2	1.900000E+01	2.389633E-01	2.381845E-01	1.927782E-02
4	3	1.900000E+01	1.197584E-01	1.196449E-01	5.212376E-03
2	1	2.000000E+01	4.777899E-01	4.761115E-01	4.001329E-02
3	2	2.000000E+01	2.392843E-01	2.388163E-01	1.495858E-02
4	3	2.000000E+01	1.198525E-01	1.198331E-01	2.156888E-03
2	1	2.100000E+01	4.783327E-01	4.768865E-01	3.716701E-02
3	2	2.100000E+01	2.397596E-01	2.396848E-01	5.987594E-03
4	3	2.100000E+01	1.198706E-01	1.198693E-01	-5.658274E-04
2	1	2.200000E+01	4.652513E-01	4.652245E-01	4.999885E-03
3	2	2.200000E+01	2.290758E-01	2.264920E-01	-3.430861E-02
4	3	2.200000E+01	1.196223E-01	1.193733E-01	-7.715418E-03
2	1	2.300000E+01	4.780885E-01	4.766998E-01	3.641302E-02
3	2	2.300000E+01	2.388598E-01	2.379776E-01	2.051019E-02
4	3	2.300000E+01	1.185458E-01	1.172343E-01	1.758482E-02
2	1	2.400000E+01	4.784867E-01	4.774878E-01	3.090271E-02
3	2	2.400000E+01	2.393547E-01	2.389660E-01	1.363386E-02
4	3	2.400000E+01	1.197503E-01	1.196288E-01	5.392987E-03
2	1	2.500000E+01	4.786937E-01	4.778913E-01	2.770607E-02
3	2	2.500000E+01	2.395140E-01	2.392779E-01	1.063214E-02
4	3	2.500000E+01	1.198455E-01	1.198192E-01	2.515019E-03
2	1	2.600000E+01	4.776323E-01	4.771558E-01	2.133126E-02
3	2	2.600000E+01	2.396199E-01	2.394620E-01	8.695636E-03
4	3	2.600000E+01	1.198719E-01	1.198719E-01	7.350634E-05
2	1	2.700000E+01	4.789912E-01	4.782109E-01	2.732805E-02
3	2	2.700000E+01	2.398119E-01	2.398077E-01	-1.413772E-03
4	3	2.700000E+01	1.197288E-01	1.195859E-01	-5.848744E-03
2	1	2.800000E+01	4.782868E-01	4.771689E-01	3.268143E-02
3	2	2.800000E+01	2.387724E-01	2.378254E-01	2.124559E-02
4	3	2.800000E+01	1.184267E-01	1.169989E-01	1.833409E-02
2	1	2.900000E+01	4.788788E-01	4.782639E-01	2.425972E-02
3	2	2.900000E+01	2.394787E-01	2.392083E-01	1.137745E-02
4	3	2.900000E+01	1.197475E-01	1.196231E-01	5.455597E-03
2	1	3.000000E+01	4.790309E-01	4.785743E-01	2.090940E-02
3	2	3.000000E+01	2.396037E-01	2.394635E-01	8.194673E-03
4	3	3.000000E+01	1.198397E-01	1.198075E-01	2.778069E-03
2	1	3.100000E+01	4.791116E-01	4.787502E-01	1.860632E-02
3	2	3.100000E+01	2.396740E-01	2.395971E-01	6.069809E-03

4	3	3.100000E+01	1.198703E-01	1.198687E-01	6.262391E-04
2	1	3.200000E+01	4.803177E-01	4.799417E-01	1.900072E-02
3	2	3.200000E+01	2.400549E-01	2.400206E-01	4.056983E-03
4	3	3.200000E+01	1.198015E-01	1.197311E-01	-4.106218E-03
2	1	3.300000E+01	4.790590E-01	4.785014E-01	2.310689E-02
3	2	3.300000E+01	2.176105E-01	1.985928E-01	-8.896764E-02
4	3	3.300000E+01	1.180506E-01	1.162569E-01	2.050054E-02
2	1	3.400000E+01	4.789846E-01	4.784731E-01	2.212913E-02
3	2	3.400000E+01	2.394739E-01	2.391977E-01	1.149797E-02
4	3	3.400000E+01	1.197451E-01	1.196184E-01	5.507161E-03
2	1	3.500000E+01	4.792055E-01	4.789122E-01	1.676319E-02
3	2	3.500000E+01	2.396382E-01	2.395265E-01	7.317231E-03
4	3	3.500000E+01	1.198345E-01	1.197972E-01	2.992857E-03
2	1	3.600000E+01	4.793263E-01	4.791647E-01	1.244263E-02
3	2	3.600000E+01	2.397091E-01	2.396743E-01	4.086869E-03
4	3	3.600000E+01	1.198667E-01	1.198615E-01	1.120276E-03
2	1	3.700000E+01	4.805382E-01	4.804518E-01	9.113684E-03
3	2	3.700000E+01	2.429025E-01	2.428962E-01	1.760611E-03
4	3	3.700000E+01	1.198457E-01	1.198194E-01	-2.508721E-03
2	1	3.800000E+01	4.725722E-01	4.723404E-01	1.479767E-02
3	2	3.800000E+01	2.343988E-01	2.343544E-01	4.565530E-03
4	3	3.800000E+01	1.165618E-01	1.133431E-01	2.720287E-02
2	1	3.900000E+01	4.791271E-01	4.787291E-01	1.952631E-02
3	2	3.900000E+01	2.390585E-01	2.383714E-01	1.811246E-02
4	3	3.900000E+01	1.197435E-01	1.196153E-01	5.540901E-03
2	1	4.000000E+01	4.792343E-01	4.789645E-01	1.607835E-02
3	2	4.000000E+01	2.396228E-01	2.394977E-01	7.741811E-03
4	3	4.000000E+01	1.198312E-01	1.197905E-01	3.123407E-03

SURF.ERR File

```
 Sample Run            08:59AM    12-13-91

 jw =     37.0  after 20 iterations has error of   3.218%
            I=  4   J=  3   |G|=  5.4623E-02   |GOLD|=  5.2920E-02
 jw =     38.0  after 20 iterations has error of  26.770%
            I=  1   J=  3   |G|=  1.5091E-02   |GOLD|=  1.1904E-02
```

5.0 Flow Diagram



6.0 Variable Description

Variables in Commons

	/ADMCOL/	
ADMBAC	INTEGER*2	maximum value of admittance for plot
ADMLIN	INTEGER*2	line color of admittance plot
	/ARCCON/	
XC	REAL*4	x coordinate of curve center
YC	REAL*4	y coordinate of curve center
RAD	REAL*4	radius of bend
ANG	REAL*4	angle of bend in radians
ANGLE	REAL*4	angle of bend in degrees
	/EPARAM/	
MENG	INTEGER*2	number of engines
TFLOW(25)	REAL*4	total flow rate of engine (lbm/sec)
PCHMB(25)	REAL*4	chamber pressure (lbf/ft^2)
DPROR(25)	REAL*4	pressure drop across orifices (lbf/ft^2)
PMRAT(25)	REAL*4	chamber pressure/total mass flow
	/FACTOR/	
SFAC	REAL*4	factor for frequency
	/FOPIPE/	
PIPE1F(75,25)	REAL*4	first parameter of pipe description
PIPE2F(75,25)	REAL*4	second parameter of pipe description
PIPE3F(75,25)	REAL*4	third parameter of pipe description
PIPE4F(75,25)	REAL*4	fourth parameter of pipe description
PIPE5F(75,25)	REAL*4	fifth parameter of pipe description
	/INTVAL/	
NSEC(75,25)	INTEGER*2	number of integration points in a section
NPTS(25)	INTEGER*2	number of surface points in a line
	/NOCOL/	
MODE	INTEGER*2	graphics mode of monitor
MODET	INTEGER*2	text mode of monitor
NTROWS	INTEGER*2	number of text rows for graphics
NTCOLS	INTEGER*2	number of text columns for graphics
NPROWS	INTEGER*2	number of pixel rows for graphics
NPCOLS	INTEGER*2	number of pixel columns for graphics
	/PARAM/	
MLINE	INTEGER*2	number of lines from tank
SPLIT(25)	REAL*4	number of unique lines from pipe split
A(25)	REAL*4	speed of sound in the fluid (ft/sec)
CMAN(25)	REAL*4	manifold capacitance
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft^3)
KMAN(25)	REAL*4	bulk modulus of manifold (lbf/ft^2)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft^2)

LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
VOL(25)	REAL*4	volume of tank (ft^3)
VOLMF(25)	REAL*4	volume of manifold (ft^3)
AREA(75,25)	REAL*4	area of pipe section (ft^2)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
L(75,25)	REAL*4	length of pipe section (ft)
PIND(75,25)	REAL*4	inductance of pipe section
PCAP(75,25)	REAL*4	capacitance of pipe section
AVGK(25)	REAL*4	average bulk modulus
SEGMNF(25)	INTEGER*2	number of pipe sections
SECINF(75,25)	INTEGER*2	pipe section type
NOLINE(25)	INTEGER*2	number of identical lines
IENG(25)	INTEGER*2	engine number
ITANK(25)	INTEGER*2	tank number
LOPOLD(25)	INTEGER*2	previous maximum number of iterations
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe

/PIPPXY/

X	REAL*4	x location of current centerline
XH	REAL*4	x location of current upper pipe
XL	REAL*4	x location of current lower pipe
Y	REAL*4	y location of current centerline
YH	REAL*4	y location of current upper pipe
YL	REAL*4	y location of current lower pipe
XMIN	REAL*4	minimum x value of piping layout
XMAX	REAL*4	maximum x value of piping layout
YMIN	REAL*4	minimum y value of piping layout
YMAX	REAL*4	maximum y value of piping layout
SINA	REAL*4	sine of current pipe direction
COSA	REAL*4	cosine of current pipe direction

/SETUP/

PIPE1(150)	REAL*4	current first parameter of pipe description
PIPE2(150)	REAL*4	current second parameter of pipe description
PIPE3(150)	REAL*4	current third parameter of pipe description
PIPE4(150)	REAL*4	current fourth parameter of pipe description
NENG(25)	INTEGER*2	engine for a line
NTANK(25)	INTEGER*2	tank for a line
NLINE(25)	INTEGER*2	section number for a line
NSP(25)	INTEGER*2	starting line number
NEND(25)	INTEGER*2	ending line number
ILINE	INTEGER*2	current line
SEGMN	INTEGER*2	current number of pipe sections
SECIN(150)	INTEGER*2	current pipe section type

/TANK/

MTANK	INTEGER*2	number of tanks

/WCAOUT

NAMLIN(2)	CHAR*24	name of file containing pipe description

/WCAPAS/

IFRST	INTEGER*2	flag for admittance plot

/WCATIT/

TITLE	CHAR*40	title for plots
TITL	CHAR*20	title from pipe file
IHR	INTEGER*2	hour code run
IMIN	INTEGER*2	minute code run
AP	CHAR*2	AM or PM
IYR	INTEGER*2	yesr code run
IMON	INTEGER*2	month code run
IDAY	INTEGER*2	day code run

/WORK1/

G(0:75,25)	COMPLEX*8	admittance looking toward tank
ZT(0:75,25)	COMPLEX*8	impedance looking toward tank
ZG(0:75,25)	COMPLEX*8	impedance looking toward engine

/WORK2/

ZO(75,25)	REAL*4	characteristic impedance

PROGRAM ADMIT

Determines maximum array sizes

Local Variables

I	INTEGER*4	do loop index
IERR	INTEGER*2	error flag for ALLOCATE
IXMAX	INTEGER*4	maximum number of frequencies
IYMAX	INTEGER*4	maximum number of points along piping
X(IXMAX,IYMAX)	REAL*4	frequency array for plotting
XF(IXMAX)	REAL*4	frequency array
Y(IXMAX,IYMAX)	REAL*4	location array for plotting
YF(IYMAX)	REAL*4	location array
Z(IXMAX,IYMAX)	REAL*4	gain array for plotting
ZF(IXMAX,IYMAX)	REAL*4	gain array

SUBROUTINE MAINP(X,Y,Z,XF,YF,ZF,IXMAX,IYMAX)

Logic portion of code

Commons EPARAM FACTOR FOPIPE INTVAL PARAM SETUP WCAOUT WCATIT WORK1 WORK2

Variables in Argument List

IXMAX	INTEGER*4	maximum number of frequencies
IYMAX	INTEGER*4	maximum number of points along piping
X(IXMAX,IYMAX)	REAL*4	frequency array for plotting
XF(IXMAX)	REAL*4	frequency array
Y(IXMAX,IYMAX)	REAL*4	location array for plotting
YF(IYMAX)	REAL*4	location array
Z(IXMAX,IYMAX)	REAL*4	gain array for plotting
ZF(IXMAX,IYMAX)	REAL*4	gain array

Local Variables

AM	CHAR*2	'AM'
ANS	CHAR*1	response to question
GADM(25)	COMPLEX*8	admittance looking toward tank
HFREQ	REAL*4	maximum frequency requested
I	INTEGER*2	do loop index
IFUEL	INTEGER*2	fuel flag
IFULOX	INTEGER*2	type line flag (fuel or lox)
IGONE	INTEGER*2	flag for subroutine fuel or lox
ILOX	INTEGER*2	lox flag
IOPEN	INTEGER*2	flag indicating if SURF.ERR is open
IP	INTEGER*2	current pipe section
ISEC	INTEGER*2	second code run
ISIZ	INTEGER*2	counter for number of integration segments
ISIZZ	INTEGER*2	counter for number of integration segments
I100	INTEGER*2	hundreth of second code run
J	INTEGER*2	do loop index
JJ	INTEGER*2	do loop index
JUNIT	INTEGER*2	unit number of engine data file
K	INTEGER*2	do loop index
LFREQ	REAL*4	minimum frequency requested
NAMENG	CHAR*24	name of engine data file
PM	CHAR*2	'PM'
PTS	INTEGER*2	number of frequencies
S	COMPLEX*8	current frequency
SSIZE	REAL*4	frequency step size
W	REAL*4	oscillatory part of frequency

SUBROUTINE ADMGRAPH

Plots admittance curve

Commons FACTOR NOCOL WCATIT

Variables in Argument List

ADMMAX	REAL*4	maximum value of admittance for plot
HFREQ	REAL*4	maximum frequency requested
LFREQ	REAL*4	minimum frequency requested

Local Variables

XMAJ	REAL*4	distance between tick marks on x axis
XMAX	REAL*4	maximum value of x
XMIN	REAL*4	mimimum value of x
YMAJ	REAL*4	distance between tick marks on y axis
YMAX	REAL*4	maximum value of y
YMIN	REAL*4	mimimum value of y

SUBROUTINE ADMIT

Determines admittance looking toward tank

Commons WCATIT WORK1 WORK2 FACTOR

Variables in Argument List

A	REAL*4	speed of sound in the fluid (ft/sec)
AREA(75,25)	REAL*4	area of pipe section (ft^2)

CMAN(25)	REAL*4	manifold capacitance
CTANK	REAL*4	tank capacitance
DPROR(25)	REAL*4	pressure drop across orifices (lbf/ft^2)
GADM(25)	COMPLEX*8	admittance looking toward tank
IENG(25)	INTEGER*2	engine number
ILINE	INTEGER*2	line number
IP	INTEGER*2	current pipe section
L(75,25)	REAL*4	length of pipe section (ft)
LFLOW	REAL*4	flow rate through pipe (lbm/sec)
LOPEND	INTEGER*2	maximum number of iterations for split pipe
NOLINE(25)	INTEGER*2	number of identical lines
PCAP(75,25)	REAL*4	capacitance of pipe section
PIND(75,25)	REAL*4	inductance of pipe section
PMRAT(25)	REAL*4	chamber pressure/total mass flow
S	COMPLEX*8	current frequency
SECTN(75,25)	INTEGER*2	pipe section type
SEGMN(25)	INTEGER*2	number of pipe sections
SPLIT	REAL*4	number of unique lines from pipe split
TFLOW(25)	REAL*4	total flow rate of engine (lbm/sec)
	Local Variables	
CAPM	COMPLEX*8	intermediate variable
CAPN	COMPLEX*8	intermediate variable
CFAC	COMPLEX*8	intermediate variable
ERRP	REAL*4	convergence error
GDIF	REAL*4	maximum difference in admittance
GOLD(0:75,25)	COMPLEX*8	previous addmittance
GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec^2)
I	INTEGER*2	do loop index
IE	INTEGER*2	current engine number
IOPEN	INTEGER*2	flag indicating if SURF.ERR is open
IWG	INTEGER*2	first index of maximum error
J	INTEGER*2	do loop index
JWG	INTEGER*2	second index of maximum error
K	INTEGER*2	do loop index
KLOOP	INTEGER*2	do loop index
LOPHI	INTEGER*2	intermediate variable
RATPM	REAL*4	intermediate variable
RHS	COMPLEX*8	intermediate variable
TCOUNT	REAL*4	intermediate variable
TL	REAL*4	length/speed of sound
TMASS	REAL*4	intermediate variable
WG	REAL*4	intermediate variable
WGOLD	REAL*4	intermediate variable
ZGEFF	COMPLEX*8	effective impedance for calculations
ZLP	REAL*4	intermediate variable
ZOEFF	REAL*4	effective ZO for calculations
ZOR(25)	REAL*4	intermediate variable
ZTEFF	COMPLEX*8	effective Zt for calculations
ZTOP	REAL*4	intermediate variable

SUBROUTINE ALLPT

Supervises plot of admittance after calculations

Commons NOCOL

Variables in Argument List

IFULOX	INTEGER*2	type line flag (fuel or lox)
IXMAX	INTEGER*2	maximum number of frequencies
IYMAX	INTEGER*2	maximum number of points along piping
PTS	INTEGER*2	number of frequencies
X(PTS)	REAL*4	frequency array for plotting
XP(IXMAX,IYMAX)	REAL*4	frequency array
Y(PTS)	REAL*4	admittance array for plotting
YP(IXMAX,IYMAX)	REAL*4	admittance array

Local Variables

ADMMAX	REAL*4	maximum value of admittance for plot
I	INTEGER*2	do loop index
IEXTEN	INTEGER*2	intermediate variable
IGO	INTEGER*2	first time flag for unsplit pipes
IKEY	INTEGER*2	intermediate variable
IPLOT	INTEGER*2	plot pointer for line

SUBROUTINE BENDS

Computes effective straight pipe for bend

Variables in Argument List

DIME	REAL*4	effective diameter (ft)
PIPE1	REAL*4	radius of bend (ft)
PIPE2	REAL*4	angle of bend (degrees)
PIPE3	REAL*4	diameter of bend (ft)
PIPE4	REAL*4	length of end straight segments (ft)
VALUE	REAL*4	effective length (ft)

Local Variables

GAMMA	REAL*4	intermediate variable
LBEND	REAL*4	intermediate variable
RATIO	REAL*4	intermediate variable
Y	REAL*4	intermediate variable

SUBROUTINE BNSECT

Computes plot coordinates for a bend

Commons ARCCON PIPPXY

Variables in Argument List

ITYPE(200)	INTEGER*2	type plot element
J	INTEGER*2	pointer to element
PIPE1	REAL*4	first parameter of pipe description
PIPE2	REAL*4	second parameter of pipe description
PIPE3	REAL*4	third parameter of pipe description
PIPE4	REAL*4	fourth parameter of pipe description
POINT(8,200)	REAL*4	description of plot element

Local Variables

DIA	REAL*4	intermediate variable
HOLD	REAL*4	intermediate variable
RANG	REAL*4	intermediate variable
SLENTH	REAL*4	intermediate variable
X0	REAL*4	intermediate variable
X1	REAL*4	intermediate variable
X2	REAL*4	intermediate variable
X3	REAL*4	intermediate variable
Y0	REAL*4	intermediate variable
Y1	REAL*4	intermediate variable
Y2	REAL*4	intermediate variable
Y3	REAL*4	intermediate variable

COMPLEX FUNCTION CCOSH

Evaluates the complex hyperbolic cosine

Variables in Argument List

S	COMPLEX*8	current frequency

Local Variables

COSHI	REAL*4	intermediate variable
COSHR	REAL*4	intermediate variable
LAMDA	REAL*4	real part of complex frequency
MU	REAL*4	imaginary part of complex frequency

COMPLEX FUNCTION CSINH

Evaluates the complex hyperbolic sine

Variables in Argument List

S	COMPLEX*8	current frequency

Local Variables

LAMDA	REAL*4	real part of complex frequency
MU	REAL*4	imaginary part of complex frequency
SINHI	REAL*4	intermediate variable
SINHR	REAL*4	intermediate variable

COMPLEX FUNCTION CTANH

Evaluates the complex hyperbolic tangent

Variables in Argument List

S	COMPLEX*8	current frequency

SUBROUTINE ENGNO
Reads engine parameters

Commons EPARAM

Variables in Argument List

IUNIT	INTEGER*2	unit number of engine file

Local Variables

I	INTEGER*2	do loop index

SUBROUTINE FLPLOT
Supervises plot of piping

Commons EPARAM FOPIPE PARAM SETUP

Variables in Argument List

IFULOX	INTEGER*2	type line flag (fuel or lox)
IGO	INTEGER*2	first time flag for unsplit pipes
IPLOT	INTEGER*2	plot pointer for line

Local Variables

I	INTEGER*2	pointer
IUNIT	INTEGER*2	unit number of work file
J	INTEGER*2	pointer
K	INTEGER*2	counter
LL	INTEGER*2	do loop index

SUBROUTINE FREQRS
Computes pressure transfer function

Commons EPARAM FOPIPE INTVAL PARAM SETUP WORK1 WORK2

Variables in Argument List

DXHOLD(IYMAX)	REAL*4	array to hold dx values up to split
IYMAX	INTEGER*2	maximum number of points along piping
K	INTEGER*2	frequency pointer
S	COMPLEX*8	current frequency
YF(IYMAX)	REAL*4	location array
ZF(IYMAX)	REAL*4	gain array
ZFHOLD(IYMAX)	REAL*4	array to hold gain up to split

Local Variables

BOTTOM	COMPLEX*8	intermediate variable
CAPM	COMPLEX*8	intermediate variable
CAPN	COMPLEX*8	intermediate variable
DX	REAL*4	x increment
I	INTEGER*2	do loop index
IP	INTEGER*2	current segment counter
IPP	INTEGER*2	current line counter
ISTART	INTEGER*2	starting loop index
J	INTEGER*2	do loop index
JJ	INTEGER*2	do loop index
KK	INTEGER*2	do loop index
LITTLN	COMPLEX*8	intermediate variable
LSEC	INTEGER*2	number of segments of pipe section
M	INTEGER*2	location pointer

MOLD	INTEGER*2	number of points from tank to split
PRAT	COMPLEX*8	pressure ratio
SUMX	REAL*4	distance from orifice
TOP	COMPLEX*8	intermediate variable
X	REAL*4	distance along pipe section
ZFAC	COMPLEX*8	intermediate variable

SUBROUTINE FUEL

Handles fuel piping logic

Commons EPARAM FOPIPE PARAM WCAOUT

Variables in Argument List

GF(25)	COMPLEX*8	admittance looking toward tank
IGONE	INTEGER*2	flag for subroutine fuel or lox
IUNIT	INTEGER*2	unit number of fuel data file
IUNITP	INTEGER*2	unit number of fuel work file
S	COMPLEX*8	current frequency

Local Variables

ANS	CHAR*1	response to question
FUELIN	CHAR*24	name of fuel data file

SUBROUTINE FULOX

Handles read, modify, and admittance calls for fuel and lox

Commons EPARAM WCATIT

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
AREA(75,25)	REAL*4	area of pipe section (ft^2)
AVGK(25)	REAL*4	average bulk modulus
CMAN(25)	REAL*4	manifold capacitance
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft^3)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
GF(25)	COMPLEX*8	admittance looking toward tank
IENG(25)	INTEGER*2	engine number
IGONE	INTEGER*2	flag for subroutine fuel or lox
ITANK(25)	INTEGER*2	tank number
ITLIN	INTEGER*2	flag indication fuel or lox
IUNIT	INTEGER*2	unit number of piping data file
IUNITP	INTEGER*2	unit number of working file
KMAN(25)	REAL*4	bulk modulus of manifold (lbf/ft^2)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft^2)
L(75,25)	REAL*4	length of pipe section (ft)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe
LOPOLD(25)	INTEGER*2	previous maximum number of iterations
MLINE	INTEGER*2	number of lines from tank
NOLINE(25)	INTEGER*2	number of identical lines
PCAP(75,25)	REAL*4	capacitance of pipe section
PIND(75,25)	REAL*4	inductance of pipe section
PIPE1(75,25)	REAL*4	first parameter of pipe description

PIPE2(75,25)	REAL*4	second parameter of pipe description
PIPE3(75,25)	REAL*4	third parameter of pipe description
PIPE4(75,25)	REAL*4	fourth parameter of pipe description
PIPE5(75,25)	REAL*4	fifth parameter of pipe description
S	COMPLEX*8	current frequency
SECTN(75,25)	INTEGER*2	pipe section type
SEGMN(25)	INTEGER*2	number of pipe sections
SPLIT(25)	REAL*4	number of unique lines from pipe split
VOL(25)	REAL*4	volume of tank (ft^3)
VOLMF(25)	REAL*4	volume of manifold (ft^3)
	Local Variables	
ANS	CHAR*1	response to question
I	INTEGER*2	do loop index
ILINE	INTEGER*2	current line number
IP	INTEGER*2	current segment number
IT	INTEGER*2	current tank number
QUEST1(2)	CHAR*40	question array
QUEST2(2)	CHAR*48	question array
QUEST3(2)	CHAR*40	question array

SUBROUTINE GETLIN

Determines line to be plotted

Commons EPARAM	FOPIPE PARAM SETUP	
	Variables in Argument List	
IFULOX	INTEGER*2	type line flag (fuel or lox)
IGO	INTEGER*2	first time flag for unsplit pipes
IPLOT	INTEGER*2	plot pointer for line
	Local Variables	
ANS	REAL*4	response to question
I	INTEGER*2	do loop index
IP	INTEGER*2	pointer to current segment
J	INTEGER*2	do loop index

SUBROUTINE GETZF

Determines pressure transfer function to be plotted

Commons EPARAM	INTVAL PARAM SETUP	
	Variables in Argument List	
IGO	INTEGER*2	first time flag for unsplit pipes
IFULOX	INTEGER*2	type line flag (fuel or lox)
IPLOT	INTEGER*2	plot pointer for line
IPTS	INTEGER*2	number of frequencies
IXMAX	INTEGER*2	maximum number of frequencies
IYMAX	INTEGER*2	maximum number of points along piping
JPTS	INTEGER*2	number of points on current line
Y(IYMAX)	REAL*4	intermediate array
YF(IYMAX)	REAL*4	plot location array
Z(IYMAX)	REAL*4	intermediate array
ZF(IXMAX,IYMAX)	REAL*4	plot transfer function array

Local Variables

I	INTEGER*2	do loop index
J	INTEGER*2	do loop index
JLIN	INTEGER*2	line pointer
K	INTEGER*2	do loop index

SUBROUTINE GINERT

Evaluates curve fit of inertance of bends

Variables in Argument List

BEND	REAL*4	angle of bend (degrees)
X	REAL*4	ratio of inner to outer radius
Y	REAL*4	inertance

Local Variables

A	REAL*4	intermediate variable
B(3)	REAL*4	coefficient array for inertance fit

SUBROUTINE HHSECT

Computes plot coordinates for Helmholtz resonator

Commons PIPPXY

Variables in Argument List

DIA	REAL*4	diameter of opening (ft)
ITYPE(200)	INTEGER*2	type plot element
J	INTEGER*2	pointer to element
LEN	REAL*4	length of opening (ft)
POINT(8,200)	REAL*4	description of plot element
VOL	REAL*4	volume of reservoir (ft^3)

Local Variables

COSOLD	REAL*4	intermediate variable
DIAM	REAL*4	intermediate variable
SIDE	REAL*4	intermediate variable
SINOLD	REAL*4	intermediate variable
XC	REAL*4	intermediate variable
XHOLD	REAL*4	intermediate variable
XLOLD	REAL*4	intermediate variable
XOLD	REAL*4	intermediate variable
YC	REAL*4	intermediate variable
YHOLD	REAL*4	intermediate variable
YLOLD	REAL*4	intermediate variable
YOLD	REAL*4	intermediate variable

SUBROUTINE LOWERW

Sets up lower plotting window

Commons ADMCOL NOCOL

Variables in Argument List

ADMMAX	REAL*4	maximum value of admittance for plot
HFREQ	REAL*4	maximum frequency requested
LFREQ	REAL*4	minimum frequency requested

Local Variables

ASPECT	REAL*4	aspect ratio of monitor screen
IOPT	INTEGER*2	intermediate variable
JCOL1	INTEGER*2	starting column for admittance window
JCOL2	INTEGER*2	ending column for afdmittance window
JROW1	INTEGER*2	starting row for admittance window
JROW2	INTEGER*2	ending row for admittance window
XLEN	REAL*4	intermediate variable
XMAX	REAL*4	maximum x value for admittance plot
XMIN	REAL*4	minimum x value for admittance plot
XORG	REAL*4	x origin for admittance plot
YLEN	REAL*4	intermediate variable
YMAX	REAL*4	maximum y value for admittance plot
YMIN	REAL*4	minimum y value for admittance plot
YORG	REAL*4	y origin for admittance plot
YOVERX	REAL*4	intermediate variable

SUBROUTINE LOX

Handles lox piping logic

Commons EPARAM FOPIPE PARAM WCAOUT

Variables in Argument List

GOX(25)	COMPLEX*8	admittance looking toward tank
IGONE	INTEGER*2	flag for subroutine fuel or lox
IUNIT	INTEGER*2	unit number of lox data file
IUNITP	INTEGER*2	unit number of lox work file
S	COMPLEX*8	current frequency

Local Variables

ANS	CHAR*1	response to question
LOXIN	CHAR*24	name of lox data file

SUBROUTINE MODENG

Modifies engine parameters

Commons EPARAM

Variables in Argument List

IUNIT	INTEGER*2	unit number of engine data file
NAMENG	CHAR*24	name of engine data file

Local Variables

ANS	CHAR*1	response to question
I	INTEGER*2	pointer
II	INTEGER*2	do loop index
J	INTEGER*2	do loop index
NAME	CHAR*8	name of variable to be modified
VALUE	REAL*4	value of variable to be modified
VARL(3)	CHAR*8	array of names (lower case)
VARU(3)	CHAR*8	array of names (upper case)

SUBROUTINE MODIFY

Allows modifications to input data

Commons EPARAM TANK WCAOUT

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
AREA(75,25)	REAL*4	area of pipe section (ft^2)
AVGK(25)	REAL*4	average bulk modulus
CMAN(25)	REAL*4	manifold capacitance
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft^3)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
IENG(25)	INTEGER*2	engine number
ITANK(25)	INTEGER*2	tank number
IUNIT	INTEGER*2	unit number of fuel or lox file
KMAN(25)	REAL*4	bulk modulus of manifold (lbf/ft^2)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft^2)
L(75,25)	REAL*4	length of pipe section (ft)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe
LOPOLD(25)	INTEGER*2	previous maximum number of iterations
MLINE	INTEGER*2	number of lines from tank
NOLINE(25)	INTEGER*2	number of identical lines
PCAP(75,25)	REAL*4	capacitance of pipe section
PIND(75,25)	REAL*4	inductance of pipe section
PIPE1(75,25)	REAL*4	first parameter of pipe description
PIPE2(75,25)	REAL*4	second parameter of pipe description
PIPE3(75,25)	REAL*4	third parameter of pipe description
PIPE4(75,25)	REAL*4	fourth parameter of pipe description
PIPE5(75,25)	REAL*4	fifth parameter of pipe description
SECIN(75,25)	INTEGER*2	pipe section type
SEGMN(25)	INTEGER*2	number of pipe sections
SPLIT(25)	REAL*4	number of unique lines from pipe split
TITL	CHAR*20	title from input file
VOL(25)	REAL*4	volume of tank (ft^3)
VOLMF(25)	REAL*4	volume of manifold (ft^3)

Local Variables

ANS	CHAR*1	response to question
I	INTEGER*2	pointer
II	INTEGER*2	do loop index
III	INTEGER*2	do loop index
IP	INTEGER*2	pointer to current segment
IPP	INTEGER*2	do loop index
ISEGMN	INTEGER*2	intermediate variable
IT	INTEGER*2	current tank number
J	INTEGER*2	do loop index
K	INTEGER*2	do loop index
M	INTEGER*2	do loop index
NAMNAM	INTEGER*2	pointer, fuel or lox
SECT	REAL*4	type of segment

SUBROUTINE MODTAN

Modifies tank parameters

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft^3)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft^2)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
MTANK	INTEGER*2	number of tanks
VOL(25)	REAL*4	volume of tank (ft^3)

Local Variables

ANS	CHAR*1	response to question
GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec^2)
I	INTEGER*2	pointer
II	INTEGER*2	do loop index
J	INTEGER*2	do loop index
NAME	CHAR*8	name of variable to be modified
VALUE	REAL*4	value of variable to be modified
VARL(4)	CHAR*8	array of names (lower case)
VARU(4)	CHAR*8	array of names (upper case)

SUBROUTINE PIPPLOT

Supervises plot of piping layout

Commons ARCCON PIPPXY

Variables in Argument List

IENG	INTEGER*2	engine number
ILOX	INTEGER*2	flag for fuel or lox
ITANK	INTEGER*2	tank number
PIPE1(75)	REAL*4	first parameter of pipe description
PIPE2(75)	REAL*4	second parameter of pipe description
PIPE3(75)	REAL*4	third parameter of pipe description
PIPE4(75)	REAL*4	fourth parameter of pipe description
SECTN(75)	INTEGER*2	pipe section type
SEGMN	INTEGER*2	number of pipe sections

Local Variables

I	INTEGER*2	do loop index
ITYPE(200)	INTEGER*2	type plot element
J	INTEGER*2	pointer to element
POINT(8,200)	REAL*4	description of plot element
XP(2)	REAL*4	x plot array
XRANGE	REAL*4	range of x values
X0	REAL*4	intermediate variable
X1	REAL*4	intermediate variable
X2	REAL*4	intermediate variable
X3	REAL*4	intermediate variable
YP(2)	REAL*4	y plot array
YRANGE	REAL*4	range of y values
Y0	REAL*4	intermediate variable
Y1	REAL*4	intermediate variable
Y2	REAL*4	intermediate variable

Y3	REAL*4	intermediate variable

SUBROUTINE PLOTSU
Supervises the surface plot

Commons FACTOR SETUP WCATIT

Variables in Argument List

IFULOX	INTEGER*2	type line flag (fuel or lox)
IPTS	INTEGER*2	actual number of frequencies
IXMAX	INTEGER*4	maximum number of frequencies
IYMAX	INTEGER*4	maximum number of points along piping
MLINE	INTEGER*2	number of separate lines
X(IPTS,JPTS)	REAL*4	frequency array for plotting
XF(IXMAX)	REAL*4	frequency array
Y(IPTS,JPTS)	REAL*4	location array for plotting
YF(IYMAX)	REAL*4	location array
Z(IPTS,JPTS)	REAL*4	gain array for plotting
ZF(IXMAX,IYMAX)	REAL*4	gain array

Local Variables

ANS	CHAR*1	response to question
ASPECT	REAL*4	aspect ratio of monitor
FOPIPE(2)	CHAR*4	name (fuel or lox) array for plot
I	INTEGER*2	do loop index
IBOARD	INTEGER*2	type graphics board installed
ICOLR	INTEGER*2	background color
IEXTEN	INTEGER*2	extension of key hit
IFIL	INTEGER*2	fill color
IGO	INTEGER*2	flag for changes
IGOO	INTEGER*2	first time flag for unsplit pipes
IKEY	INTEGER*2	code of key hit
ILIN	INTEGER*2	line color
IPLOT	INTEGER*2	plot pointer for line
IWIRE	INTEGER*2	flag for wire-frame or filled
IWR	INTEGER*2	temporary flag for wire-frame or filled
IWRK1(640)	INTEGER*2	work array for plot routine
IWRK2(640)	INTEGER*2	work array for plot routine
J	INTEGER*2	do loop index
JPTS	INTEGER*2	number of points on current line
LEGEND	CHAR*45	legend for CGA monitor
LEGENDH	CHAR*58	legend for EGA or VGA monitor (Hertz)
LEGENDR	CHAR*58	legend for EGA or VGA monitor (rad/sec)
MODE	INTEGER*2	graphics mode
MODET	INTEGER*2	text mode
NCOLT	INTEGER*2	number of columns in text mode
P	REAL*4	phi rotation angle (degrees)
PIPING	CHAR*38	array for line identification
T	REAL*4	theta rotation angle (degrees)
XFAC	REAL*4	intermediate variable
XINV	REAL*4	intermediate variable
XLEN	REAL*4	length of x axis
XMAJ	REAL*4	distance between tick marks on x axis
XMAX	REAL*4	maximum value for x axis

XMIN	REAL*4	minimum value for x axis
XYZLEN	REAL*4	intermediate variable
YFAC	REAL*4	intermediate variable
YINV	REAL*4	intermediate variable
YLEN	REAL*4	length of y axis
YMAJ	REAL*4	distance between tick marks on y axis
YMAX	REAL*4	maximum value for y axis
YMIN	REAL*4	minimum value for y axis
ZFAC	REAL*4	intermediate variable
ZINV	REAL*4	intermediate variable
ZLEN	REAL*4	length of z axis
ZMAJ	REAL*4	distance between tick marks on z axis
ZMAX	REAL*4	maximum value for z axis
ZMIN	REAL*4	minimum value for z axis

SUBROUTINE PLSECT

Computes plot coordinates for parallel resonator

Commons ARCCON PIPPXY

Variables in Argument List

DIA	REAL*4	diameter of parallel segment (ft)
ITYPE(200)	INTEGER*2	type plot element
J	INTEGER*2	pointer to element
LEN	REAL*4	length of parallel segment (ft)
POINT(8,200)	REAL*4	description of plot element
VOL	REAL*4	volume of bypassed segment (ft^3)

Local Variables

ANGOLD	REAL*4	intermediate variable
ANGSAV	REAL*4	intermediate variable
COSOLD	REAL*4	intermediate variable
DIAM	REAL*4	intermediate variable
PDIA	REAL*4	intermediate variable
PLEN	REAL*4	intermediate variable
RADIUS	REAL*4	intermediate variable
SIDE	REAL*4	intermediate variable
SINOLD	REAL*4	intermediate variable
TURN	REAL*4	intermediate variable
XHC	REAL*4	intermediate variable
XHOLD	REAL*4	intermediate variable
XHSAV	REAL*4	intermediate variable
XLC	REAL*4	intermediate variable
XLOLD	REAL*4	intermediate variable
XLSAV	REAL*4	intermediate variable
XOLD	REAL*4	intermediate variable
XSAV	REAL*4	intermediate variable
YHC	REAL*4	intermediate variable
YHOLD	REAL*4	intermediate variable
YHSAV	REAL*4	intermediate variable
YLC	REAL*4	intermediate variable
YLOLD	REAL*4	intermediate variable
YLSAV	REAL*4	intermediate variable
YOLD	REAL*4	intermediate variable

YSAV	REAL*4	intermediate variable

SUBROUTINE PLTCON

Supervises plot of contour plot

Commons FACTOR SETUP WCATIT

Variables in Argument List

IFULOX	INTEGER*2	type line flag (fuel or lox)
IPTS	INTEGER*2	actual number of frequencies
IXMAX	INTEGER*4	maximum number of frequencies
IYMAX	INTEGER*4	maximum number of points along piping
MLINE	INTEGER*2	number of separate lines
X(IPTS)	REAL*4	frequency array for plotting
XF(IXMAX)	REAL*4	frequency array
Y(IYMAX)	REAL*4	location array for plotting
YF(IYMAX)	REAL*4	location array
Z(IPTS,IYMAX)	REAL*4	gain array for plotting
ZF(IXMAX,IYMAX)	REAL*4	gain array

Local Variables

ANS	REAL*4	response to question
ASPECT	REAL*4	aspect ratio of monitor
CONS(10)	REAL*4	array for values of contour lines
FOPIPE(2)	CHAR*4	name (fuel or lox) array for plot
I	INTEGER*2	do loop index
IBOARD	INTEGER*2	type graphics board installed
ICOLR	INTEGER*2	background color
IDEF	INTEGER*2	flag for plot routine
IEXTEN	INTEGER*2	extension of key hit
IFIL	INTEGER*2	fill color
IGO	INTEGER*2	first time flag for unsplit pipes
IKEY	INTEGER*2	code of key hit
ILIN	INTEGER*2	line color
IOPT	INTEGER*2	flag for plot routine
IPLOT	INTEGER*2	plot pointer for line
J	INTEGER*2	do loop index
JCOL1	INTEGER*2	starting column for contour plot window
JCOL2	INTEGER*2	ending column for contour plot window
JPTS	INTEGER*2	number of points on current line
JROW1	INTEGER*2	starting row for contour plot window
JROW2	INTEGER*2	ending row for contour plot window
LABL(10)	INTEGER*2	flags for labeling contours
MODE	INTEGER*2	graphics mode
MODET	INTEGER*2	text mode
NCOLT	INTEGER*2	number of columns in text mode
PIPING	CHAR*38	array for line identification
XMAJ	REAL*4	distance between tick marks on x axis
XMAX	REAL*4	maximum value for x axis
XMIN	REAL*4	minimum value for x axis
XORG	REAL*4	origin of x axis
YMAJ	REAL*4	distance between tick marks on y axis
YMAX	REAL*4	maximum value for y axis
YMIN	REAL*4	minimum value for y axis

YORG	REAL*4	origin of y axis
YOVERX	REAL*4	intermediate variable
ZLEN	REAL*4	intermediate variable
ZMAX	REAL*4	maximum value for z
ZMIN	REAL*4	minimum value for z

SUBROUTINE RLINE

Reads fuel or lox file.

Commons EPARAM TANK

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
AREA(75,25)	REAL*4	area of pipe section (ft^2)
AVGK(25)	REAL*4	average bulk modulus
CMAN(25)	REAL*4	manifold capacitance
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft^3)
DIA(75,25)	REAL*4	diameter of pipe section (ft)
IENG(25)	INTEGER*2	engine number
ITANK(25)	INTEGER*2	tank number
IUNIT	INTEGER*2	unit number of fuel or lox file
KMAN(25)	REAL*4	bulk modulus of manifold (lbf/ft^2)
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft^2)
L(75,25)	REAL*4	length of pipe section (ft)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
LOPEND(25)	INTEGER*2	maximum number of iterations for split pipe
LOPOLD(25)	INTEGER*2	previous maximum number of iterations
MLINE	INTEGER*2	number of lines from tank
NOLINE(25)	INTEGER*2	number of identical lines
PCAP(75,25)	REAL*4	capacitance of pipe section
PIND(75,25)	REAL*4	inductance of pipe section
PIPE1(75,25)	REAL*4	first parameter of pipe description
PIPE2(75,25)	REAL*4	second parameter of pipe description
PIPE3(75,25)	REAL*4	third parameter of pipe description
PIPE4(75,25)	REAL*4	fourth parameter of pipe description
PIPE5(75,25)	REAL*4	fifth parameter of pipe description
SECIN(75,25)	INTEGER*2	pipe section type
SEGMN(25)	INTEGER*2	number of pipe sections
SPLIT(25)	REAL*4	number of unique lines from pipe split
TITL	CHAR*20	title from input file
VOL(25)	REAL*4	volume of tank (ft^3)
VOLMF(25)	REAL*4	volume of manifold (ft^3)

Local Variables

ANS	CHAR*1	response to question
DIVAVG	REAL*4	intermediate variable
I	INTEGER*2	do loop index
IE	INTEGER*2	current engine number
IT	INTEGER*2	current tank number
J	INTEGER*2	do loop index
M	INTEGER*2	pointer
MM	INTEGER*2	do loop index

SUBROUTINE RTYPE
Stores values for different types of piping

Variables in Argument List

AREA	REAL*4	area of pipe section (ft^2)
AVGK	REAL*4	average bulk modulus
CMAN	REAL*4	manifold capacitance
DENS	REAL*4	density of fluid (lbm/ft^3)
DIA	REAL*4	diameter of pipe section (ft)
KMAN	REAL*4	bulk modulus of manifold (lbf/ft^2)
L	REAL*4	length of pipe section (ft)
PCAP	REAL*4	capacitance of pipe section
PIND	REAL*4	inductance of pipe section
PIPE1	REAL*4	first parameter of pipe description
PIPE2	REAL*4	second parameter of pipe description
PIPE3	REAL*4	third parameter of pipe description
PIPE4	REAL*4	fourth parameter of pipe description
PIPE5	REAL*4	fifth parameter of pipe description
SECTN	INTEGER*2	pipe section type
VOLMF	REAL*4	volume of manifold (ft^3)

Local Variables

AREAB	REAL*4	area of pipe
DIME	REAL*4	diameter of pipe
GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec^2)
PI	REAL*4	mathematical constant
VALUE	REAL*4	length of pipe

SUBROUTINE SETPLT
Sets up the plot environment

Commons ADMCOL NOCOL WCAPAS

Local Variables

ANS	CHAR*1	response to question
IBOARD	INTEGER*2	type graphics board installed
ITIM	INTEGER*2	flag for initialization
NCOLT	INTEGER*2	number of columns in text mode

SUBROUTINE STSECT
Computes plot coordinates for a straight section

Commons PIPPXY

Variables in Argument List

DIA	REAL*4	diameter of segment (ft)
ITYPE(200)	INTEGER*2	type plot element
J	INTEGER*2	pointer to element
LEN	REAL*4	length of segment (ft)
POINT(8,200)	REAL*4	description of plot element

SUBROUTINE TANKNO
Reads tank parameters

Variables in Argument List

A(25)	REAL*4	speed of sound in the fluid (ft/sec)
CTANK(25)	REAL*4	tank capacitance
DENS(25)	REAL*4	density of fluid (lbm/ft^3)
IUNIT	INTEGER*2	unit number of fuel or lox file
KTANK(25)	REAL*4	bulk modulus of tank (lbf/ft^2)
LFLOW(25)	REAL*4	flow rate through pipe (lbm/sec)
MTANK	INTEGER*2	number of tanks
VOL(25)	REAL*4	volume of tank (ft^3)

Local Variables

GRAV	REAL*4	gravitational constant (lbm-ft/lbf-sec^2)
I	INTEGER*2	do loop index

SUBROUTINE TSSECT

Computes plot coordinates for a tuned stub

Commons PIPPXY

Variables in Argument List

DIA	REAL*4	diameter of tuned stub (ft)
ITYPE(200)	INTEGER*2	type plot element
J	INTEGER*2	pointer to element
LEN	REAL*4	length of tuned stub
POINT(8,200)	REAL*4	description of plot element

Local Variables

DIAM	REAL*4	intermediate variable

SUBROUTINE UPPERW

Sets up upper plotting window

Commons ADMCOL NOCOL

Variables in Argument List

IENG	INTEGER*2	engine number
IFULOX	INTEGER*2	type line flag (fuel or lox)
ITANK	INTEGER*2	tank number
X0	REAL*4	minimum value of x for piping layout window
X1	REAL*4	maximum value of x for piping layout window
Y0	REAL*4	minimum value of y for piping layout window
Y1	REAL*4	maximum value of y for piping layout window

Local Variables

ASPECT	REAL*4	aspect ratio of monitor
CHANGE	REAL*4	intermediate variable
FULOX	CHAR*36	array for line identification
IOPT	INTEGER*2	flag for plot routine
JCOL1	INTEGER*2	starting column for pipe layout plot window
JCOL2	INTEGER*2	ending column for pipe layout plot window
JROW1	INTEGER*2	starting row for pipe layout plot window
JROW2	INTEGER*2	ending row for pipe layout plot window
XMAX	REAL*4	maximum value for x axis
XMIN	REAL*4	minimum value for x axis
XORG	REAL*4	origin of x axis
YMAX	REAL*4	maximum value for x axis

YMAX0	REAL*4	intermediate variable
YMIN	REAL*4	minimum value for x axis
YORG	REAL*4	origin of x axis
YOVERX	REAL*4	intermediate variable

SUBROUTINE WINDOW

Sets up window for surface plot

Variables in Argument List

MODE	INTEGER*2	graphics mode
XFIN	REAL*4	final x value
XSCALE	REAL*4	aspect ratio of monitor
XST	REAL*4	starting x value
YFIN	REAL*4	final y value
YST	REAL*4	starting y value
ZFIN	REAL*4	final z value
ZST	REAL*4	starting z value

Local Variables

ASPECT	REAL*4	aspect ratio of monitor
IOPT	INTEGER*2	flag for plot routine
JCOL1	INTEGER*2	starting column for surface plot window
JCOL2	INTEGER*2	ending column for surface plot window
JROW1	INTEGER*2	starting row for surface plot window
JROW2	INTEGER*2	ending row for surface plot window
XMAX	REAL*4	maximum value for x axis
XMIN	REAL*4	minimum value for x axis
XORG	REAL*4	origin of x axis
YMAX	REAL*4	maximum value for y axis
YMIN	REAL*4	minimum value for y axis
YORG	REAL*4	origin of y axis
YOVERX	REAL*4	intermediate variable

FUNCTION XFUN

Parametric function for plotting of bends

Commons ARCCON

Variables in Argument List

T	REAL*4	angle in radians

FUNCTION YFUN

Parametric function for plotting of bends

Commons ARCCON

Variables in Argument List

T	REAL*4	angle in radians

SUBROUTINE ZREAD

Reads input for input modification

	Variables in Argument List	
NAME(8)	CHAR*1	name of input variable
VALUE	REAL*4	value of input variable
	Local Variables	
BLK	CHAR*1	' '
CARD(80)	CHAR*1	card image
CEND(3)	CHAR*1	'E','N','D'
COMMA	CHAR*1	','
CTIT(5)	CHAR*1	'T','I','T','L','E'
DCARD	CHAR*80	card image
E	CHAR*1	'E'
FRACT	REAL*4	fractional part of number
I	INTEGER*2	do loop index
ICOUNT	INTEGER*2	position counter
ID	INTEGER*2	position counter
II	INTEGER*2	position counter
J	INTEGER*2	do loop index
JJ	INTEGER*2	position counter
LE	CHAR*1	'e'
LEND(3)	CHAR*1	'e','n','d'
LTIT(5)	CHAR*1	't','i','t','l','e'
MINUS	CHAR*1	'-'
NUMBER(10)	CHAR*1	'0','1','2','3','4','5','6','7','8','9'
PERIOD	CHAR*1	'.'
PLUS	CHAR*1	'+'
POUND	CHAR*1	'#'
QUEST	CHAR*1	'?'
SIGN	REAL*4	sign of number or exponent
WHOLE	REAL*4	whole part of number

7.0 Program Listing

```
C
C      PROGRAM ADMIT  03-24-92
C
C         Program to compute and plot admittance coefficients, pipe layout,
C                   and pressure transfer function
C
C                      Variable Dimension Version
C
C         This program will handle the following type elements
C
C               Straight pipes
C               Bends
C               Split pipes (into identical lines)
C               Inline accumulators
C               Tuned stub accumulators
C               Helmholtz resonators
C               Parallel resonators
C               Pumps
C
C
      INTEGER*4 IXMAX,IYMAX,I
      REAL X[ALLOCATABLE](:,:),Y[ALLOCATABLE](:,:),Z[ALLOCATABLE](:,:),
     *     XF[ALLOCATABLE](:),YF[ALLOCATABLE](:),ZF[ALLOCATABLE](:,:)
      DO 21 I=150,1,-1
       IXMAX=I
       IYMAX=I
       IERR=0
       ALLOCATE(X(IXMAX,IYMAX),Y(IXMAX,IYMAX),Z(IXMAX,IYMAX),STAT=IERR)
       ALLOCATE(XF(IXMAX),YF(IYMAX),ZF(IXMAX,IYMAX),STAT=IERR)
       IF(IERR.EQ.0)  GO TO 22
       DEALLOCATE(X,Y,Z,XF,YF,ZF,STAT=IERR)
   21 CONTINUE
      STOP
   22 CONTINUE
      CALL QCLEAR(0,7)
      WRITE(*,'(10X,A)')
     *'╔═══════════════════════════════════════════════════════════════════╗'
      WRITE(*,'(10X,A)')
     *'║                                                                   ║'
      WRITE(*,'(10X,A)')
     *'║         Welcome to ADMIT - a Feedline Analysis Program            ║'
      WRITE(*,'(10X,A)')
     *'║                                                                   ║'
      WRITE(*,'(10X,A)')
     *'║                    To send a plot to the printer                  ║'
      WRITE(*,'(10X,A)')
     *'║                                                                   ║'
      WRITE(*,'(10X,A)')
     *'║               The computer MUST be in GRAPHICS mode               ║'
      WRITE(*,'(10X,A)')
```

```
     *'╟
      WRITE(*,'(10X,A)')                                                    ╟'
     *'╟      Hit PrScn to send the current plot to the printer             ╟'
      WRITE(*,'(10X,A)')
     *'╟
      WRITE(*,'(10X,A)')                                                    ╟'
     *'╚═══════════════════════════════════════════════════════════════════╝'
      WRITE(*,*)' '
      WRITE(*,'(20X,A,I3)')'Maximum no. of frequencies = ',IXMAX
      WRITE(*,'(20X,A,I3)')'Maximum points along pipe  = ',IYMAX
      WRITE(*,*)' '
      CALL MAINP(X,Y,Z,XF,YF,ZF,IXMAX,IYMAX)
      STOP
      END
      SUBROUTINE MAINP(X,Y,Z,XF,YF,ZF,IXMAX,IYMAX)
C       Logic portion of code
      INTEGER*4 IXMAX,IYMAX
      REAL X(IXMAX,IYMAX),Y(IXMAX,IYMAX),Z(IXMAX,IYMAX)
      REAL XF(IXMAX),YF(IYMAX),ZF(IXMAX,IYMAX)
      INTEGER*2 PTS
      CHARACTER ANS*1
      CHARACTER*24 NAMLIN(2),NAMENG
      COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
      INTEGER SEGMNF(25),SECINF(75,25),NOLINE(25),IENG(25),ITANK(25),
     *        LOPOLD(25),LOPEND(25)
      REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25),LFREQ
      COMMON /PARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
     *              DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
     *              AREA(75,25),DIA(75,25),L,PIND(75,25),
     *              PCAP(75,25),AVGK(25),
     *              SEGMNF,SECINF,NOLINE,IENG,ITANK,LOPOLD,LOPEND
      COMMON /FOPIPE/PIPE1F(75,25),PIPE2F(75,25),PIPE3F(75,25),
     *               PIPE4F(75,25),PIPE5F(75,25)
      INTEGER SEGMN,SECIN(150)
      COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
     *              NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
     *              SEGMN,SECIN
      COMMON /WCAOUT/NAMLIN
      COMMON /INTVAL/NSEC(75,25),NPTS(25)
      COMPLEX G(0:75,25),ZT(0:75,25),ZG(0:75,25),S,GADM(25)
      COMMON /WORK1/G,ZT,ZG
      COMMON /WORK2/ZO(75,25)
      INTEGER*2 IHR,IMIN,ISEC,I100,IYR,IMON,IDAY
      CHARACTER*2 AM,PM,AP
      CHARACTER*40 TITLE
      CHARACTER*20 TITL
      COMMON /WCATIT/TITLE,TITL,IHR,IMIN,AP,IYR,IMON,IDAY
      COMMON /FACTOR/SFAC
      DATA AM/'AM'/,PM/'PM'/
      DATA IOPEN/0/
    1 FORMAT(2I5,1P4E15.6)
    2 FORMAT(/10X,A//'  LINE','  ENG.',5X,'FREQ',11X,'|G|',12X,'G(R)',
     *       11X,'G(I)'/)
```

```
    3 FORMAT(A20,2X,I2.2,':',I2.2,A2,3X,I2.2,'-',I2.2,'-',I2.2)
      OPEN(UNIT=17,FORM='UNFORMATTED')
      OPEN(UNIT=14,FILE='SURF.OUT')
      SFAC=1.0
      WRITE(*,'(A)')' If you want frequency in rad/sec, hit enter.'
      WRITE(*,'(A\)')' If you want it in Hertz, enter "H". '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'H'.OR.ANS.EQ.'h')  SFAC=6.283185
      LOPOLD=20
      CALL GETTIM(IHR,IMIN,ISEC,I100)
      CALL GETDAT(IYR,IMON,IDAY)
      IYR=IYR-1900
      IF(IHR.LT.12)  THEN
       AP=AM
      ELSE
       AP=PM
       IF(IHR.GT.12)  IHR=IHR-12
      ENDIF
      WRITE(*,'(A\)')' Is the engine data on file ENG.RLN? (Y/N) '
      READ(*,'(A)')ANS
      IF(ANS.NE.'N'.AND.ANS.NE.'n')  THEN
       OPEN(UNIT=9,FILE='ENG.RLN')
      ELSE
       WRITE(*,'(A\)')'  Enter name of file with the engine data '
       READ(*,'(A)')NAMENG
       OPEN(UNIT=9,FILE=NAMENG)
      ENDIF
      JUNIT=9
      CALL ENGNO(JUNIT)
   22 CONTINUE
      ISIZ=0
      IGONE=2
      WRITE(*,'(A\)')' Is this setup for FUEL or OXIDIZER? Enter F or O
     *. '
      READ(*,'(A)')ANS
       IFUEL=1
       ILOX=1
      IF(ANS.EQ.'F'.OR.ANS.EQ.'f')  THEN
       IFUEL=0
       IFULOX=1
       CALL FUEL(S,GADM,11,16,IGONE)
      ELSEIF(ANS.EQ.'O'.OR.ANS.EQ.'o')  THEN
       ILOX=0
       IFULOX=2
       CALL LOX(S,GADM,10,15,IGONE)
      ELSE
       WRITE(*,*)' You did not enter F or O.  Try again'
       GO TO 22
      ENDIF
      WRITE(TITLE,3)TITL,IHR,IMIN,AP,IMON,IDAY,IYR
      IGONE=0
   23 CONTINUE
   24 CONTINUE
```

```
 25 CONTINUE
    WRITE(*,*)' '
    WRITE(*,*)TITLE
    WRITE(*,*)' '
    REWIND 17
    IF(SFAC.EQ.1.0)  THEN
     WRITE(*,*)' Enter range of frequencies in rad/sec '
    ELSE
     WRITE(*,*)' Enter range of frequencies in Hertz '
    ENDIF
    WRITE(*,*)' Low freq, high freq, #pts'
    READ(*,*)LFREQ,HFREQ,PTS
    IF(PTS.LT.1)  GO TO 34
    IF(PTS.GT.IXMAX)  THEN
     WRITE(*,*)' Maximum number of points for this option is IXMAX =',
   *          IXMAX
     WRITE(*,'(A\)')'    Do you want PTS reduced to IXMAX? Y or N '
     READ(*,'(A)')ANS
     IF(ANS.EQ.'N'.OR.ANS.EQ.'n')  GO TO 34
     PTS=IXMAX
    ENDIF
    IF(LFREQ.EQ.0.0)  LFREQ=1.0E-5
    SSIZE=0.0
    IF(PTS.NE.1)  SSIZE=(HFREQ-LFREQ)/(PTS-1)
    IF(ISIZ.NE.0)  THEN
     WRITE(*,'(A\)')' Do you wish to change segments of sections? '
     READ(*,'(A)')ANS
     IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  GO TO 26
     GO TO 31
    ENDIF
 26 CONTINUE
    IP=0
    DO 30 J=1,MLINE
     IP=IP+1
     ISIZ=0
     DO 27 I=1,SEGMNF(IP)
      IF(SECINF(I,IP).LE.1)  THEN
       WRITE(*,'(A,F10.5,A\)')' How many segments should this',
   *       L(I,IP),' ft. long section be broken into? '
       READ(*,*)NSEC(I,IP)
       IF(NSEC(I,IP).LE.1)  NSEC(I,IP)=2
      ELSEIF(SECINF(I,IP).EQ.2)  THEN
       NSEC(I,IP)=2
      ELSE
       NSEC(I,IP)=2
      ENDIF
      ISIZ=ISIZ+NSEC(I,IP)
      IF(ISIZ.GT.IYMAX)  THEN
       WRITE(*,*)' Too many segments ',ISIZ
       WRITE(*,*)' Maximum is IYMAX =',IYMAX,' Try again.'
       GO TO 26
      ENDIF
 27   CONTINUE
```

```
      IF(SPLIT(J).LE.0)  GO TO 30
      ISIZZ=ISIZ
      DO 29 JJ=1,SPLIT(J)
       IP=IP+1
       ISIZ=ISIZZ
       DO 28 I=1,SEGMNF(IP)
        IF(SECINF(I,IP).LE.1)  THEN
        WRITE(*,'(A,F10.5,A\)')' How many segments should this',
     *       L(I,IP),' ft. long section be broken into? '
        READ(*,*)NSEC(I,IP)
        IF(NSEC(I,IP).LE.1)  NSEC(I,IP)=2
       ELSEIF(SECINF(I,IP).EQ.2)  THEN
        NSEC(I,IP)=2
       ELSE
        NSEC(I,IP)=2
       ENDIF
       ISIZ=ISIZ+NSEC(I,IP)
       IF(ISIZ.GT.IYMAX)  THEN
        WRITE(*,*)' Too many segments ',ISIZ
        WRITE(*,*)' Maximum is IYMAX =',IYMAX,' Try again.'
        GO TO 26
       ENDIF
28    CONTINUE
29   CONTINUE
30  CONTINUE
31  CONTINUE
    WRITE(14,2)TITLE
    IF(IOPEN.NE.0.AND.LOPEND(1).NE.1)  THEN
     WRITE(13,*)' '
     WRITE(13,*)' '
     WRITE(13,*)TITLE
     WRITE(13,*)' '
    ENDIF
    WRITE(*,*)' Calculations are proceeding'
    DO 33 K=1,PTS
     W=LFREQ+SSIZE*(K-1)
     XF(K)=W
     S=CMPLX(0.0,W*SFAC)
     IF(IFUEL.EQ.0)  CALL FUEL(S,GADM,11,16,IGONE)
     IF(ILOX.EQ.0)  CALL LOX(S,GADM,10,15,IGONE)
     CALL FREQRS(S,YF,ZF(1,1),ZF(1,2),ZF(1,3),K,IYMAX)
     IP=0
     DO 32 I=1,25
      X(K,I)=W
      Y(K,I)=CABS(GADM(I))
      IF(IENG(I).NE.0)  THEN
       IP=IP+1
       WRITE(14,1)I,IENG(I),W,Y(K,IP),GADM(IP)
      ENDIF
32    CONTINUE
33   CONTINUE
    WRITE(*,'(A\)')' Do you wish to plot piping & admittances? '
    READ(*,'(A)')ANS
```

```
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       CALL ALLPT(X,Y,Z(1,1),Z(1,3),PTS,IXMAX,IYMAX,IFULOX)
      ENDIF
      WRITE(*,'(A\)')' Do you wish to plot surfaces? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       CALL PLOTSU(X,Y,Z,XF,YF,ZF,PTS,IXMAX,IYMAX,IFULOX,MLINE*MENG)
      ENDIF
      WRITE(*,'(A\)')' Do you wish to plot contours? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       CALL PLTCON(X,Y,Z,XF,YF,ZF,PTS,IXMAX,IYMAX,IFULOX,MLINE*MENG)
      ENDIF
   34 CONTINUE
      WRITE(*,'(A\)')' Enter E to exit, F to run new frequency range, or
     * C to run a new case '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'F'.OR.ANS.EQ.'f')  GO TO 25
      IF(ANS.EQ.'E'.OR.ANS.EQ.'e')  RETURN
      IF(ANS.EQ.'C'.OR.ANS.EQ.'c')  THEN
       WRITE(*,'(A\)')'    Do you wish to modify engine file.? '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
        CALL MODENG(JUNIT,NAMENG)
       ELSE
        WRITE(*,'(A\)')'    Do you wish to rewind engine file.? '
        READ(*,'(A)')ANS
        IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  REWIND JUNIT
        CALL ENGNO(JUNIT)
       ENDIF
       IGONE=1
       IF(IFULOX.EQ.1)  THEN
        CALL FUEL(S,GADM,11,16,IGONE)
       ELSE
        CALL LOX(S,GADM,10,15,IGONE)
       ENDIF
       WRITE(TITLE,3)TITL,IHR,IMIN,AP,IMON,IDAY,IYR
       ISIZ=0
       GO TO 25
      ENDIF
      WRITE(*,*)' You did not enter E, F, or C. Try again.'
      GO TO 34
      END
      SUBROUTINE ADMGRAPH(LFREQ,HFREQ,ADMMAX)
C       Plots admittance curve
      CHARACTER*40 TITLE
      CHARACTER*20 TITL
      INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
      CHARACTER*2 AP
      COMMON /WCATIT/TITLE,TITL,IHR,IMIN,AP,IYR,IMON,IDAY
      COMMON /NOCOL/MODE,MODET,NTROWS,NTCOLS,NPROWS,NPCOLS
      COMMON /FACTOR/SFAC
      REAL LFREQ
```

```
      XMIN=LFREQ
      XMAX=HFREQ
      YMIN=0.0
      YMAX=ADMMAX
      XMAJ=0.25*(XMAX-XMIN)
      YMAJ=0.25*(YMAX-YMIN)
      IF(MODE.NE.18)  THEN
       CALL QPTXT(40,TITLE,7,17,11)
      ELSE
       CALL QPTXT(40,TITLE,7,17,14)
      ENDIF
      CALL QXAXIS(XMIN,XMAX,XMAJ,0,-1,2)
      IF(SFAC.EQ.1)  THEN
       CALL QPTXTA(20,'Frequency - rad/sec ',7)
      ELSE
       CALL QPTXTA(20,' Frequency - Hertz  ',7)
      ENDIF
      CALL QYAXIS(YMIN,YMAX,YMAJ,0,0,0)
      CALL QPTXTD(8,'Adm.    ',7)
      CALL QYAXIS(YMIN,YMAX,YMAJ,0,-1,2)
      RETURN
      END
      SUBROUTINE ADMIT(S,GADM,A,AREA,CMAN,CTANK,DPROR,L,LFLOW,PMRAT,
     *                 SEGMN,SECIN,SPLIT,LOPEND,PCAP,PIND,IENG,TFLOW,
     *                 NOLINE,IP,ILINE)
C       Determines admittance looking toward tank
      CHARACTER*40 TITLE
      CHARACTER*20 TITL
      INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
      CHARACTER*2 AP
      COMMON /WCATIT/TITLE,TITL,IHR,IMIN,AP,IYR,IMON,IDAY
      INTEGER SEGMN(25),SECIN(75,25)
      INTEGER IENG(25),NOLINE(25)
      REAL AREA(75,25),PCAP(75,25),PIND(75,25),L(75,25),LFLOW,ZO(75,25),
     *     CMAN(25),DPROR(25),PMRAT(25),ZOR(25),TFLOW(25)
      COMPLEX G(0:75,25),ZT(0:75,25),ZG(0:75,25),GOLD(0:75,25),GADM(25),
     *        S,ZGEFF,ZTEFF
      COMMON /WORK1/G,ZT,ZG
      COMMON /WORK2/ZO
      COMMON /FACTOR/SFAC
      COMPLEX CTANH,RHS,CFAC,CAPN,CAPM
      DATA GRAV/32.2/
      DATA IOPEN/0/
      ZTOP=A/GRAV
        TMASS=0.0
        TCOUNT=0.0
      DO 22 J=IP,IP+SPLIT
       GOLD(0,J)=0.0
       SECIN(SEGMN(J)+1,J)=0
       DO 21 I=1,SEGMN(J)
        GOLD(I,J)=0.0
        ZO(I,J)=0.0
        IF(SECIN(I,J).LE.2)  THEN
```

```
            ZO(I,J)=ZTOP/AREA(I,J)
           ELSEIF(SECTN(I,J).EQ.7)  THEN
            ZO(I,J)=0.0
           ELSE
            ZO(I,J)=SQRT(PIND(I,J)/PCAP(I,J))
           ENDIF
   21     CONTINUE
          IF(IENG(J).NE.0)  THEN
           IE=IENG(J)
           ZOR(J)=2.0*DPROR(IE)/LFLOW
           IF(J.EQ.IP.AND.SPLIT.EQ.0.0)  THEN
            TMASS=TFLOW(IE)
           ELSEIF(J.NE.IP)  THEN
            TMASS=TMASS+NOLINE(J)*TFLOW(IE)
            TCOUNT=TCOUNT+NOLINE(J)
           ENDIF
          ENDIF
   22    CONTINUE
         IF(TCOUNT.EQ.0.0)  TCOUNT=1.0
         G(0,IP)=CTANK*S
         G(0,IP)=G(0,IP)/TCOUNT
         ZT(0,IP)=1.0/G(0,IP)
         DO 31 KLOOP=1,LOPEND
          DO 25 J=IP,IP+SPLIT
           IF(J.NE.IP)  THEN
            G(0,J)=G(SEGMN(IP),IP)
            ZT(0,J)=1.0/G(0,J)
           ENDIF
           DO 24 I=1,SEGMN(J)
            ZGEFF=G(I-1,J)
            IF(SECTN(I,J).LE.1)  THEN
C                bend in pipe or straight section
             TL=L(I,J)/A
             IF(KLOOP.NE.1.AND.SPLIT.NE.0.AND.J.NE.IP.AND.I.EQ.1)  THEN
              ZGEFF=0.0
              DO 23 K=IP+1,IP+SPLIT
               IE=IENG(K)
               IF(K.EQ.J)  THEN
                ZGEFF=ZGEFF+(NOLINE(K)-1.0)/ZG(0,K)
               ELSE
                ZGEFF=ZGEFF+NOLINE(K)/ZG(0,K)
               ENDIF
   23         CONTINUE
              ZGEFF=G(SEGMN(IP),IP)+ZGEFF
             ENDIF
             G(I,J)=(1.0+CTANH(S*TL)/(ZGEFF*ZO(I,J)))/(1.0+ZGEFF*
     *              ZO(I,J)*CTANH(S*TL))
            ELSEIF(SECTN(I,J).EQ.2)  THEN
C                inline resonator
             G(I,J)=1.0+PCAP(I,J)*S/ZGEFF
            ELSEIF(SECTN(I,J).EQ.3)  THEN
C                tuned stub
             G(I,J)=1.0+CTANH(S*SQRT(PIND(I,J)*PCAP(I,J)))/(ZO(I,J)*
```

```
     *           ZGEFF)
          ELSEIF(SECTN(I,J).EQ.4)  THEN
C              helmholtz resonator
           G(I,J)=1.0+S*PCAP(I,J)/(1.0+PIND(I,J)*PCAP(I,J)*S**2)/ZGEFF
          ELSEIF(SECTN(I,J).EQ.5)  THEN
C              parallel resonator
           G(I,J)=PIND(I,J)*PCAP(I,J)*S**2+1.0
           G(I,J)=G(I,J)/(G(I,J)+PIND(I,J)*S*ZGEFF)
          ELSEIF(SECTN(I,J).EQ.6)  THEN
C              pump
           G(I,J)=(1.0+PCAP(I,J)*S/ZGEFF)/(1.0+(PIND(I,J)*S+
     *           AREA(I,J))*(PCAP(I,J)*S+ZGEFF))
          ELSEIF(SECTN(I,J).EQ.7)  THEN
           G(SEGMN(J),J)=1.0+CMAN(J)*S/ZGEFF
          ENDIF
          G(I,J)=G(I,J)*ZGEFF
          ZT(I,J)=1.0/G(I,J)
   24    CONTINUE
         IF(SPLIT.NE.0.0.AND.J.EQ.IP)  GO TO 25
         G(SEGMN(J)+1,J)=1.0/(1.0+ZOR(J)*G(SEGMN(J),J))
         G(SEGMN(J)+1,J)=G(SEGMN(J)+1,J)*G(SEGMN(J),J)
   25   CONTINUE
        DO 28 J=IP+SPLIT,IP,-1
         IF(J.EQ.IP.AND.SPLIT.NE.0.0)  THEN
          LOPHI=SEGMN(J)
         ELSE
          ZG(SEGMN(J)-1,J)=ZOR(J)/(ZOR(J)*CMAN(J)*S+1.0)
          LOPHI=SEGMN(J)-2
         ENDIF
         IF(LOPHI.NE.0)  THEN
          DO 27 I=LOPHI,1,-1
           IF(I.EQ.LOPHI.AND.J.EQ.IP.AND.SPLIT.NE.0.0)  THEN
            ZG(I,J)=0.0
            ZTEFF=ZT(I-1,J)
            DO 26 K=IP+1,IP+SPLIT
             ZGEFF=ZG(1,K)
             ZOEFF=ZO(1,K)
             ZLP=L(1,K)
             TL=(L(I,J)+ZLP)/A
             CAPN=(ZOEFF-ZTEFF)/(ZOEFF+ZTEFF)
             CAPM=(ZOEFF-ZGEFF)/(ZOEFF+ZGEFF)
             CFAC=CEXP(-2.0*S*TL)
             RHS=(ZOEFF+ZGEFF)*(1.0-CAPN*CAPM*CFAC)*CEXP(S*ZLP/A)
             CFAC=CAPN*CFAC*CEXP(2.0*S*ZLP/A)
             ZG(0,K)=(RHS-ZOEFF*(1.0-CFAC))/(1.0+CFAC)
             ZG(I,J)=ZG(I,J)+NOLINE(K)/ZG(0,K)
   26       CONTINUE
            ZG(I,J)=1.0/ZG(I,J)
           ELSE
            ZGEFF=ZG(I+1,J)
            ZOEFF=ZO(I+1,J)
            ZLP=L(I+1,J)
            ZTEFF=ZT(I-1,J)
```

```
            IF(SECTN(I+1,J).LE.1)  THEN
C              bend in pipe or straight section
               TL=(L(I,J)+ZLP)/A
               CAPN=(ZOEFF-ZTEFF)/(ZOEFF+ZTEFF)
               CAPM=(ZOEFF-ZGEFF)/(ZOEFF+ZGEFF)
               CFAC=CEXP(-2.0*S*TL)
               RHS=(ZOEFF+ZGEFF)*(1.0-CAPN*CAPM*CFAC)*CEXP(S*ZLP/A)
               CFAC=CAPN*CFAC*CEXP(2.0*S*ZLP/A)
               ZG(I,J)=(RHS-ZOEFF*(1.0-CFAC))/(1.0+CFAC)
            ELSEIF(SECTN(I+1,J).EQ.2)  THEN
C              inline resonator
               ZG(I,J)=ZGEFF/(ZGEFF*PCAP(I+1,J)*S+1.0)
            ELSEIF(SECTN(I+1,J).EQ.3)  THEN
C              tuned stub
               ZG(I,J)=ZOEFF/CTANH(S*SQRT(PIND(I+1,J)*PCAP(I+1,J)))
               ZG(I,J)=(ZG(I,J)*ZGEFF)/(ZG(I,J)+ZGEFF)
            ELSEIF(SECTN(I+1,J).EQ.4)  THEN
C              helmholtz resonator
               ZG(I,J)=(1.0+PIND(I+1,J)*PCAP(I+1,J)*S**2)/(PCAP(I+1,J)*S)
               ZG(I,J)=(ZG(I,J)*ZGEFF)/(ZG(I,J)+ZGEFF)
            ELSEIF(SECTN(I+1,J).EQ.5)  THEN
C              parallel resonator
               ZG(I,J)=ZGEFF+PIND(I+1,J)*S/(PIND(I+1,J)*PCAP(I+1,J)*S**2+
     *                 1.0)
            ELSEIF(SECTN(I+1,J).EQ.6)  THEN
C              pump
               ZG(I,J)=ZGEFF+PIND(I+1,J)*S-AREA(I+1,J)
               ZG(I,J)=ZG(I,J)/(1.0+ZG(I,J)*PCAP(I+1,J)*S)
            ENDIF
          ENDIF
   27    CONTINUE
        ENDIF
   28  CONTINUE
       ERRP=0.0
       DO 30 J=IP,IP+SPLIT
        DO 29 I=1,SEGMN(J)
         GDIF=CABS(GOLD(I,J))
         IF(GDIF.NE.0.0)  GDIF=ABS(GDIF-CABS(G(I,J)))/GDIF
         IF(GDIF.GT.ERRP)  THEN
          ERRP=GDIF
          WG=CABS(G(I,J))
          WGOLD=CABS(GOLD(I,J))
          IWG=I
          JWG=J
         ENDIF
         GOLD(I,J)=G(I,J)
   29   CONTINUE
   30  CONTINUE
       IF(KLOOP.GT.1.AND.ERRP.LT.0.001)  GO TO 32
   31 CONTINUE
       IF(LOPEND.EQ.1)  GO TO 32
       IF(IOPEN.EQ.0)  THEN
        OPEN(UNIT=13,FILE='SURF.ERR')
```

```
        WRITE(13,*)' '
        WRITE(13,*)' '
        WRITE(13,*)TITLE
        WRITE(13,*)' '
        IOPEN=1
       ENDIF
       WRITE(13,'('' jw ='',F8.1,''  after'',I3,'' iterations'',
     *            '' has error of'',F8.3,''% '')')
     *           AIMAG(S)/SFAC,LOPEND,100.0*ERRP
       WRITE(13,'(10X,'' I='',I3,3X,''J='',I3,3X,''|G|='',1PE12.4,3X,
     *           ''|GOLD|='',E12.4)')IWG,JWG,WG,WGOLD
   32 CONTINUE
      DO 35 J=IP,IP+SPLIT
       IF(IENG(J).EQ.0.0)  THEN
        RATPM=0.0
        DO 33 I=IP+1,IP+SPLIT
         RATPM=RATPM+PMRAT(IENG(I))
   33   CONTINUE
        LOPHI=SEGMN(J)
       ELSE
        RATPM=PMRAT(IENG(J))
        LOPHI=SEGMN(J)+1
       ENDIF
       DO 34 I=0,LOPHI
        G(I,J)=RATPM*G(I,J)
   34  CONTINUE
       IF(J.EQ.IP.AND.SPLIT.NE.0.0)  GO TO 35
       GADM(ILINE)=G(LOPHI,J)
       ILINE=ILINE+1
   35 CONTINUE
      RETURN
      END
      SUBROUTINE ALLPT(XP,YP,X,Y,PTS,IXMAX,IYMAX,IFULOX)
C       Supervises plot of admittance after calculations
      INTEGER*2 PTS
      INTEGER*4 IXMAX,IYMAX
      COMMON /NOCOL/MODE,MODET,NTROWS,NTCOLS,NPROWS,NPCOLS
      REAL X(PTS),Y(PTS),XP(IXMAX,IYMAX),YP(IXMAX,IYMAX)
      IGO=0
   21 CONTINUE
      CALL FLPLOT(IFULOX,IPLOT,IGO)
      IF(IPLOT.EQ.0)  RETURN
      DO 22 I=1,PTS
       X(I)=XP(I,IPLOT)
       Y(I)=YP(I,IPLOT)
   22 CONTINUE
      ADMMAX=Y(1)
      DO 23 I=2,PTS
       IF(Y(I).GT.ADMMAX)  ADMMAX=Y(I)
   23 CONTINUE
      CALL LOWERW(X(1),X(PTS),ADMMAX)
      CALL ADMGRAPH(X(1),X(PTS),ADMMAX)
      CALL QTABL(1,PTS,X,Y)
```

```
   24 CONTINUE
      CALL QONKEY(IKEY)
      IF(IKEY.EQ.0)  GO TO 24
      CALL QINKEY(IEXTEN,IKEY)
      CALL QSMODE(MODET)
      GO TO 21
      RETURN
      END
      SUBROUTINE BENDS(PIPE1,PIPE2,PIPE3,PIPE4,VALUE,DIME)
C       Computes effective straight pipe for bend
      REAL LBEND
      LBEND=0.0174533*PIPE1*ABS(PIPE2)
      RATIO=(PIPE1-0.5*PIPE3)/(PIPE1+0.5*PIPE3)
      CALL GINERT(ABS(PIPE2),RATIO,Y)
      GAMMA=(LBEND+Y*PIPE3)/LBEND
      VALUE=GAMMA*(LBEND+2.0*PIPE4)
      DIME=PIPE3/(GAMMA)**0.25
      RETURN
      END
      SUBROUTINE BNSECT(J,ITYPE,POINT,PIPE1,PIPE2,PIPE3,PIPE4)
C       Computes plot coordinates for a bend
      COMMON /PIPPXY/X,XH,XL,Y,YH,YL,XMIN,XMAX,YMIN,YMAX,SINA,COSA
      COMMON /ARCCON/XC,YC,RAD,ANG,ANGLE
      REAL POINT(8,200)
      INTEGER*2 ITYPE(200)
C          first straight section of bend
      IF(PIPE4.NE.0.0)  CALL STSECT(J,ITYPE,POINT,PIPE4,PIPE3)
C          curved section of bend
      IF(PIPE2.GE.0.0)  THEN
       XC=X-SINA*PIPE1
       YC=Y+COSA*PIPE1
       DIA= 0.5
      ELSE
       XC=X+SINA*PIPE1
       YC=Y-COSA*PIPE1
       DIA=-0.5
      ENDIF
      J=J+1
      ITYPE(J)=0
      POINT(1,J)=XC
      POINT(2,J)=YC
      POINT(3,J)=ANG
      ANG=ANG+0.01745329*PIPE2
      ANGLE=ANGLE+0.5*PIPE2
      RANG=0.01745329*ANGLE
      COSA=COS(RANG)
      SINA=SIN(RANG)
      RAD=PIPE1-DIA*PIPE3
      POINT(4,J)=ANG
      POINT(5,J)=RAD
      X0=XC-RAD
      Y0=YC+RAD
      X1=XC+RAD
```

```
      Y1=YC-RAD
      X2=XH
      Y2=YH
      SLENTH=2.0*RAD*SIN(0.00872665*ABS(PIPE2))
      XH=X2+COSA*SLENTH
      YH=Y2+SINA*SLENTH
      X3=XH
      Y3=YH
      IF(DIA.LT.0.0)  THEN
       HOLD=X2
       X2=X3
       X3=HOLD
       HOLD=Y2
       Y2=Y3
       Y3=HOLD
      ENDIF
      RAD=PIPE1+DIA*PIPE3
      X0=XC-RAD
      Y0=YC+RAD
      X1=XC+RAD
      Y1=YC-RAD
      X2=XL
      Y2=YL
      SLENTH=2.0*RAD*SIN(0.00872665*ABS(PIPE2))
      XL=X2+COSA*SLENTH
      YL=Y2+SINA*SLENTH
      X3=XL
      Y3=YL
      IF(DIA.LT.0.0)  THEN
       HOLD=X2
       X2=X3
       X3=HOLD
       HOLD=Y2
       Y2=Y3
       Y3=HOLD
      ENDIF
      J=J+1
      ITYPE(J)=0
      POINT(1,J)=POINT(1,J-1)
      POINT(2,J)=POINT(2,J-1)
      POINT(3,J)=POINT(3,J-1)
      POINT(4,J)=POINT(4,J-1)
      POINT(5,J)=RAD
      SLENTH=2.0*PIPE1*SIN(0.00872665*ABS(PIPE2))
      X=X+COSA*SLENTH
      Y=Y+SINA*SLENTH
      XMIN=AMIN1(X,XL,XH,XMIN)
      XMAX=AMAX1(X,XL,XH,XMAX)
      YMIN=AMIN1(Y,YL,YH,YMIN)
      YMAX=AMAX1(Y,YL,YH,YMAX)
C          last straight section of bend
      ANGLE=ANGLE+0.5*PIPE2
      RANG=0.01745329*ANGLE
```

```
      COSA=COS(RANG)
      SINA=SIN(RANG)
      J=J+1
      ITYPE(J)=1
      POINT(1,J)=XH
      POINT(2,J)=YH
      POINT(3,J)=XL
      POINT(4,J)=YL
      X=X+COSA*PIPE4
      XH=X-0.5*SINA*PIPE3
      XL=X+0.5*SINA*PIPE3
      Y=Y+SINA*PIPE4
      YH=Y+0.5*COSA*PIPE3
      YL=Y-0.5*COSA*PIPE3
      POINT(5,J)=XH
      POINT(6,J)=YH
      POINT(7,J)=XL
      POINT(8,J)=YL
      XMIN=AMIN1(X,XL,XH,XMIN)
      XMAX=AMAX1(X,XL,XH,XMAX)
      YMIN=AMIN1(Y,YL,YH,YMIN)
      YMAX=AMAX1(Y,YL,YH,YMAX)
      RETURN
      END
      COMPLEX FUNCTION CCOSH(S)
C       Evaluates the complex hyperbolic cosine
      COMPLEX S
      REAL LAMDA, MU
      LAMDA=REAL(S)
      MU=AIMAG(S)
      COSHR=COSH(LAMDA)*COS(MU)
      COSHI=SINH(LAMDA)*SIN(MU)
      CCOSH=CMPLX(COSHR,COSHI)
      RETURN
      END
      COMPLEX FUNCTION CSINH(S)
C       Evaluates the complex hyperbolic sine
      COMPLEX S
      REAL LAMDA, MU
      LAMDA=REAL(S)
      MU=AIMAG(S)
      SINHR=SINH(LAMDA)*COS(MU)
      SINHI=COSH(LAMDA)*SIN(MU)
      CSINH=CMPLX(SINHR,SINHI)
      RETURN
      END
      COMPLEX FUNCTION CTANH(S)
C       Evaluates the complex hyperbolic tangent
      COMPLEX CCOSH,CSINH,S
      CTANH=CSINH(S)/CCOSH(S)
      RETURN
      END
      SUBROUTINE ENGNO(IUNIT)
```

```
C       Reads engine parameters
        COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
        READ(IUNIT,*)MENG
        IF(MENG.GT.25)  THEN
         WRITE(*,*)'  Number of engines must be less than 25'
         STOP
        ENDIF
        IF(MENG.LE.0)  MENG=1
        DO 21 I=1,MENG
         READ(IUNIT,*)TFLOW(I),PCHMB(I),DPROR(I)
         PMRAT(I)=PCHMB(I)/TFLOW(I)
   21 CONTINUE
        RETURN
        END
        SUBROUTINE FLPLOT(IFULOX,IPLOT,IGO)
C       Supervises plot of piping
        COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
        INTEGER SEGMNF(25),SECINF(75,25),NOLINE(25),IENG(25),ITANK(25),
       *        LOPOLD(25),LOPEND(25)
        REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
        COMMON /PARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
       *              DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
       *              AREA(75,25),DIA(75,25),L,PIND(75,25),
       *              PCAP(75,25),AVGK(25),
       *              SEGMNF,SECINF,NOLINE,IENG,ITANK,LOPOLD,LOPEND
        COMMON /FOPIPE/PIPE1F(75,25),PIPE2F(75,25),PIPE3F(75,25),
       *              PIPE4F(75,25),PIPE5F(75,25)
        INTEGER SEGMN,SECTN(150)
        COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
       *              NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
       *              SEGMN,SECTN
        IF(IFULOX.EQ.1)  THEN
         IUNIT=16
        ELSE
         IUNIT=15
        ENDIF
        J=0
        CALL GETLIN(IFULOX,IPLOT,IGO)
        IF(IPLOT.EQ.0)  RETURN
        J=NEND(IPLOT)
        I=NLINE(IPLOT)
        K=0
        SEGMN=0
        SEGMN=SEGMN+SEGMNF(I)
        REWIND IUNIT
        READ(IUNIT)PIPE1F,PIPE2F,PIPE3F,PIPE4F,PIPE5F
        DO 26 LL=1,SEGMNF(I)
         K=K+1
         SECTN(K)=SECINF(LL,I)
         PIPE1(K)=PIPE1F(LL,I)
         PIPE2(K)=PIPE2F(LL,I)
         PIPE3(K)=PIPE3F(LL,I)
         PIPE4(K)=PIPE4F(LL,I)
```

```
   26 CONTINUE
      IF(I.NE.J)  THEN
       SEGMN=SEGMN+SEGMNF(J)
       DO 27 LL=1,SEGMNF(J)
        K=K+1
        SECTN(K)=SECTNF(LL,J)
        PIPE1(K)=PIPE1F(LL,J)
        PIPE2(K)=PIPE2F(LL,J)
        PIPE3(K)=PIPE3F(LL,J)
        PIPE4(K)=PIPE4F(LL,J)
   27  CONTINUE
      ENDIF
      CALL SETPLT
      CALL PIPPLOT(SEGMN,SECTN,PIPE1,PIPE2,PIPE3,PIPE4,IFULOX,
     *             NTANK(IPLOT),NENG(IPLOT))
      RETURN
      END
      SUBROUTINE FREQRS(S,YF,ZF,ZFHOLD,DXHOLD,K,IYMAX)
C       Computes pressure transfer function
      COMPLEX G(0:75,25),ZT(0:75,25),ZG(0:75,25),S
      COMMON /WORK1/G,ZT,ZG
      COMMON /WORK2/ZO(75,25)
      COMPLEX LITTLN,CAPM,CAPN,ZFAC,TOP,BOTTOM,PRAT
      COMMON /INTVAL/NSEC(75,25),NPTS(25)
      COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
      INTEGER SEGMNF(25),SECTNF(75,25),NOLINE(25),IENG(25),ITANK(25),
     *        LOPOLD(25),LOPEND(25)
      REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
      COMMON /PARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
     *              DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
     *              AREA(75,25),DIA(75,25),L,PIND(75,25),
     *              PCAP(75,25),AVGK(25),
     *              SEGMNF,SECTNF,NOLINE,IENG,ITANK,LOPOLD,LOPEND
      COMMON /FOPIPE/PIPE1F(75,25),PIPE2F(75,25),PIPE3F(75,25),
     *              PIPE4F(75,25),PIPE5F(75,25)
      INTEGER SEGMN,SECTN(150)
      COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
     *              NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
     *              SEGMN,SECTN
      INTEGER*4 IYMAX
      REAL YF(IYMAX),ZF(IYMAX),ZFHOLD(IYMAX),DXHOLD(IYMAX)
      IP=0
      IPP=0
      DO 28 KK=1,MLINE
       IP=IP+1
       LITTLN=S/A(KK)
       SUMX=0.0
       M=1
       ISTART=SEGMNF(IP)
       IF(SPLIT(KK).EQ.0)  ISTART=ISTART-1
       DO 22 I=ISTART,1,-1
        CAPN=(ZO(I,IP)-ZT(I-1,IP))/(ZO(I,IP)+ZT(I-1,IP))
        CAPM=(ZO(I,IP)-ZG(I,IP))/(ZO(I,IP)+ZG(I,IP))
```

```
      ZFAC=ZO(I,IP)/(ZO(I,IP)+ZG(I,IP))
      LSEC=NSEC(I,IP)
      DX=0.0
      IF(SECTNF(I,IP).EQ.3.OR.SECTNF(I,IP).EQ.4)  THEN
       DX=DIA(I,IP)/(LSEC-1)
      ELSE
       DX=L(I,IP)/(LSEC-1)
      ENDIF
      BOTTOM=1.0-CAPM*CAPN*CEXP(-2.0*LITTLN*L(I,IP))
      DO 21 J=1,LSEC
       X=DX*(J-1)
       IF(SECTNF(I,IP).GT.1.AND.SECTNF(I,IP).LT.6)  THEN
        IF(J.EQ.LSEC)  PRAT=ZT(I-1,IP)/(ZT(I-1,IP)+ZG(I,IP))
       ELSE
        TOP=CEXP(-LITTLN*X)-CAPN*CEXP(-LITTLN*(2.0*L(I,IP)-X))
        PRAT=ZFAC*TOP/BOTTOM
       ENDIF
       IF(J.NE.1)  THEN
        SUMX=SUMX+DX
        M=M+1
        ZF(M)=CABS(PRAT)
        YF(M)=SUMX
        ZFHOLD(M)=ZF(M)
        DXHOLD(M)=DX
       ELSE
        IF(I.EQ.ISTART)  THEN
         ZF(M)=CABS(PRAT)
         YF(M)=SUMX
         ZFHOLD(M)=ZF(M)
         DXHOLD(M)=0.0
        ENDIF
       ENDIF
21    CONTINUE
22    CONTINUE
      IF(SPLIT(KK).LE.0.0)  GO TO 27
      MOLD=M
      DO 26 JJ=1,SPLIT(KK)
       IP=IP+1
       IPP=IPP+1
       SUMX=0.0
       M=1
       DO 24 I=SEGMNF(IP)-1,1,-1
        CAPN=(ZO(I,IP)-ZT(I-1,IP))/(ZO(I,IP)+ZT(I-1,IP))
        CAPM=(ZO(I,IP)-ZG(I,IP))/(ZO(I,IP)+ZG(I,IP))
        ZFAC=ZO(I,IP)/(ZO(I,IP)+ZG(I,IP))
        LSEC=NSEC(I,IP)
        DX=0.0
        IF(SECTNF(I,IP).EQ.3.OR.SECTNF(I,IP).EQ.4)  THEN
         DX=DIA(I,IP)/(LSEC-1)
        ELSE
         DX=L(I,IP)/(LSEC-1)
        ENDIF
        BOTTOM=1.0-CAPM*CAPN*CEXP(-2.0*LITTLN*L(I,IP))
```

```
            DO 23 J=1,LSEC
             X=DX*(J-1)
             IF(SECTNF(I,IP).GT.1.AND.SECTNF(I,IP).LT.6)  THEN
              IF(J.EQ.LSEC)  PRAT=ZT(I-1,IP)/(ZT(I-1,IP)+ZG(I,IP))
             ELSE
              TOP=CEXP(-LITTLN*X)-CAPN*CEXP(-LITTLN*(2.0*L(I,IP)-X))
              PRAT=ZFAC*TOP/BOTTOM
             ENDIF
             IF(J.NE.1)  THEN
              SUMX=SUMX+DX
              M=M+1
              ZF(M)=CABS(PRAT)
              YF(M)=SUMX
             ELSE
              IF(I.EQ.SEGMNF(IP)-1)  THEN
               ZF(M)=CABS(PRAT)
               YF(M)=SUMX
              ENDIF
             ENDIF
   23     CONTINUE
   24    CONTINUE
         DO 25 I=1,MOLD
          M=M+1
          ZF(M)=ZFHOLD(I)
          YF(M)=YF(M-1)+DXHOLD(I)
   25    CONTINUE
         IF(K.EQ.1)  NPTS(IPP)=M
          WRITE(17)YF,ZF
   26   CONTINUE
        GO TO 28
   27   CONTINUE
        IPP=IPP+1
        IF(K.EQ.1)  NPTS(IPP)=M
        WRITE(17)YF,ZF
   28 CONTINUE
      RETURN
      END
      SUBROUTINE FUEL(S,GF,IUNIT,IUNITP,IGONE)
C       Handles fuel piping logic
      COMPLEX GF(25),S
      COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
      INTEGER SEGMN(25),SECTN(75,25),NOLINE(25),IENG(25),ITANK(25),
     *        LOPOLD(25),LOPEND(25)
      REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
      COMMON /PARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
     *              DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
     *              AREA(75,25),DIA(75,25),L,PIND(75,25),
     *              PCAP(75,25),AVGK(25),
     *              SEGMN,SECTN,NOLINE,IENG,ITANK,LOPOLD,LOPEND
      COMMON /FOPIPE/PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),
     *               PIPE4(75,25),PIPE5(75,25)
      CHARACTER*24 FUELIN,NAMLIN(2)
      COMMON /WCAOUT/NAMLIN
```

```
      CALL RLINE(TITL,SEGMN,SECIN,PIPE1,PIPE2,PIPE3,
     *  PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
     *  A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
     *  AVGK,MLINE)
      REWIND IUNITP
      WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
      WRITE(*,'(A\)')QUEST1(ITLIN)
      READ(*,'(A)')ANS
      IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y')  THEN
       CALL MODIFY(TITL,SEGMN,SECIN,PIPE1,PIPE2,PIPE3,
     *  PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
     *  A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
     *  AVGK,MLINE)
       REWIND IUNITP
       WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
      ENDIF
     ELSEIF(IGONE.EQ.0)  THEN
      IP=1
      ILINE=1
      DO 21 I=1,MLINE
       IT=ITANK(I)
       CALL ADMIT(S,GF,A(IT),AREA,CMAN,CTANK(IT),DPROR,
     *            L,LFLOW(IT),PMRAT,SEGMN,SECIN,
     *            SPLIT(I),LOPEND(I),PCAP,PIND,IENG,TFLOW,
     *            NOLINE,IP,ILINE)
       IP=IP+SPLIT(I)+1
21    CONTINUE
      RETURN
     ELSEIF(IGONE .EQ. 1)  THEN
      WRITE(*,'(A\)')QUEST2(ITLIN)
      READ(*,'(A)')ANS
      IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y')  THEN
       CALL MODIFY(TITL,SEGMN,SECIN,PIPE1,PIPE2,PIPE3,
     *  PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
     *  A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
     *  AVGK,MLINE)
       REWIND IUNITP
       WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
      ELSE
       WRITE(*,'(A\)')QUEST3(ITLIN)
       READ(*,'(A)')ANS
       IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y')  REWIND IUNIT
       CALL RLINE(TITL,SEGMN,SECIN,PIPE1,PIPE2,PIPE3,
     *  PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
     *  A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
     *  AVGK,MLINE)
       REWIND IUNITP
       WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
       WRITE(*,*)QUEST1(ITLIN)
       WRITE(*,'(A\)')'  if not, press enter key. '
       READ(*,'(A)')ANS
       WRITE(*,*)' '
       IF(ANS .EQ. 'Y' .OR. ANS .EQ. 'y') THEN
```

```
          CALL MODIFY(TITL,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,
     *    PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
     *    A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
     *    AVGK,MLINE)
          REWIND IUNITP
          WRITE(IUNITP)PIPE1,PIPE2,PIPE3,PIPE4,PIPE5
         ENDIF
        ENDIF
        IGONE=0
       ENDIF
       RETURN
       END
       SUBROUTINE GETLIN(IFULOX,IPLOT,IGO)
C        Determines line to be plotted
       COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
       INTEGER SEGMNF(25),SECINF(75,25),NOLINE(25),IENG(25),ITANK(25),
     *         LOPOLD(25),LOPEND(25)
       REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
       COMMON /PARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
     *               DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
     *               AREA(75,25),DIA(75,25),L,PIND(75,25),
     *               PCAP(75,25),AVGK(25),
     *               SEGMNF,SECINF,NOLINE,IENG,ITANK,LOPOLD,LOPEND
       COMMON /FOPIPE/PIPE1F(75,25),PIPE2F(75,25),PIPE3F(75,25),
     *              PIPE4F(75,25),PIPE5F(75,25)
       INTEGER SEGMN,SECTN(150)
       COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
     *              NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
     *              SEGMN,SECTN
       ILINE=0
       IP=1
       DO 22 I=1,MLINE
        IF(SPLIT(I).EQ.0.0)  THEN
         ILINE=ILINE+1
         NENG(ILINE)=IENG(IP)
         NTANK(ILINE)=ITANK(I)
         NLINE(ILINE)=IP
         NSP(ILINE)=ILINE
         NEND(ILINE)=IP
        ELSE
         DO 21 J=IP+1,IP+SPLIT(I)
          ILINE=ILINE+1
          NENG(ILINE)=IENG(J)
          NTANK(ILINE)=ITANK(I)
          NLINE(ILINE)=IP
          NSP(ILINE)=ILINE
          NEND(ILINE)=J
   21    CONTINUE
        ENDIF
        IP=IP+SPLIT(I)+1
   22 CONTINUE
   23 CONTINUE
      IF(ILINE.EQ.1)  THEN
```

```
      IF(IGO.EQ.0)  THEN
       IPLOT=1
      ELSE
       IPLOT=0
      ENDIF
      IGO=1
     ELSE
      IF(IFULOX.EQ.1)  THEN
       WRITE(*,*)' The following FUEL lines may be plotted'
       WRITE(*,'(/'' Line #    Tank #   Engine #''/)')
      ELSE
       WRITE(*,*)' The following LOX lines may be plotted'
       WRITE(*,'(/'' Line #    Tank #   Engine #''/)')
      ENDIF
      DO 24 I=1,ILINE
       WRITE(*,'(I5,I10,I11)')I,NTANK(I),NENG(I)
24    CONTINUE
25    CONTINUE
      WRITE(*,'(/'' Enter line # to be plotted, 0 will end plot ''\)')
      READ(*,*)IPLOT
      IF(IPLOT.LE.0)  RETURN
      IF(IPLOT.GT.ILINE)  THEN
       WRITE(*,*)'  You did not enter a valid line #. Try again'
       GO TO 25
      ENDIF
     ENDIF
     RETURN
     END
     SUBROUTINE GETZF(Y,Z,YF,ZF,IXMAX,IYMAX,IPTS,JPTS,IFULOX,IPLOT,IGO)
C      Determines pressure transfer function to be plotted
     COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
     INTEGER SEGMNF(25),SECINF(75,25),NOLINF(25),IENGF(25),ITANKF(25),
    *        LOPOLF(25),LOPENF(25)
     REAL KMANF(25),KTANKF(25),LFLOWF(25),LF(75,25)
     COMMON /PARAM/MLINEF,SPLITF(25),AF(25),CMANF(25),CTANKF(25),
    *              DENSF(25),KMANF,KTANKF,LFLOWF,VOLF(25),VOLMFF(25),
    *              AREAF(75,25),DIAF(75,25),LF,PINDF(75,25),
    *              PCAPF(75,25),AVGKF(25),
    *              SEGMNF,SECINF,NOLINF,IENGF,ITANKF,LOPOLF,LOPENF
     INTEGER SEGMN,SECIN(150)
     COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
    *             NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
    *             SEGMN,SECIN
     COMMON /INIVAL/NSEC(75,25),NPTS(25)
     INTEGER*4 IXMAX,IYMAX
     REAL YF(IYMAX),ZF(IXMAX,IYMAX),Y(IYMAX),Z(IYMAX)
     REWIND 17
     CALL GETLIN(IFULOX,IPLOT,IGO)
     JLIN=NSP(IPLOT)
     JPTS=NPTS(JLIN)
     DO 24 I=1,IPTS
      DO 23 J=1,ILINE
       READ(17)Y,Z
```

```
      IF(J.EQ.JLIN)  THEN
       DO 21 K=1,IYMAX
        ZF(I,K)=Z(K)
21     CONTINUE
       IF(I.EQ.1)  THEN
        DO 22 K=1,IYMAX
         YF(K)=Y(K)
22      CONTINUE
       ENDIF
      ENDIF
23   CONTINUE
24  CONTINUE
    RETURN
    END
    SUBROUTINE GINERT(BEND,X,Y)
C     Evaluates curve fit of inertance of bends
    DIMENSION B(3)
    DATA B/0.0,0.7877014E-02,-0.2814679E-04/
    A=B(1)+(B(2)+B(3)*BEND)*BEND
    Y=A*(X-1.0)**2
    RETURN
    END
    SUBROUTINE HHSECT(J,ITYPE,POINT,LEN,DIA,VOL)
C     Computes plot coordinates for Helmholtz resonator
    COMMON /PIPPXY/X,XH,XL,Y,YH,YL,XMIN,XMAX,YMIN,YMAX,SINA,COSA
    REAL LEN,POINT(8,200)
    INTEGER*2 ITYPE(200)
    XOLD=X
    XHOLD=XH
    XLOLD=XL
    YOLD=Y
    YHOLD=YH
    YLOLD=YL
    SINOLD=SINA
    COSOLD=COSA
    DIAM=SQRT((XH-XL)**2+(YH-YL)**2)
    CALL TSSECT(J,ITYPE,POINT,LEN,DIA)
    XC=0.5*(XOLD+X)
    YC=0.5*(YOLD+Y)
    XOLD=X
    YOLD=Y
    SINA=COSOLD
    COSA=-SINOLD
    X=XC+COSA*(LEN+0.5*DIAM)
    Y=YC+SINA*(LEN+0.5*DIAM)
    SIDE=VOL**0.3333333
    CALL STSECT(J,ITYPE,POINT,SIDE,SIDE)
    X=XOLD
    Y=YOLD
    SINA=SINOLD
    COSA=COSOLD
    DIAM=SQRT((XHOLD-XLOLD)**2+(YHOLD-YLOLD)**2)
    XH=X-0.5*SINA*DIAM
```

```
      XL=X+0.5*SINA*DIAM
      YH=Y+0.5*COSA*DIAM
      YL=Y-0.5*COSA*DIAM
      RETURN
      END
      SUBROUTINE LOWERW(LFREQ,HFREQ,ADMMAX)
C       Sets up lower plotting window
      COMMON /NOCOL/MODE,MODET,NTROWS,NTCOLS,NPROWS,NPCOLS
      COMMON /ADMCOL/ADMBAC,ADMLIN
      INTEGER ADMBAC,ADMLIN
      REAL LFREQ
      XMIN=LFREQ
      XMAX=HFREQ
      YMIN=0.0
      YMAX=ADMMAX
      XORG=XMIN
      YORG=YMIN
      XLEN=0.01*(XMAX-XMIN)
      YLEN=0.01*(YMAX-YMIN)
      XMIN=XMIN-XLEN
      XMAX=XMAX+XLEN
      YMIN=YMIN-YLEN
      YMAX=YMAX+YLEN
      JCOL1=150
      JCOL2=550
      IF(MODE.EQ.6)  THEN
       JROW1=20
       JROW2=79
      ELSE
       JROW1=40
       IF(MODE.EQ.16)  JROW2=134
       IF(MODE.EQ.18)  JROW2=199
      ENDIF
      YOVERX=1.0
      IOPT=0
      ASPECT=1.35
      CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,
     *           XORG,YORG,IOPT,YOVERX,ASPECT)
      IF(MODE.NE.6)  THEN
       CALL QPREG(0,ADMBAC)
      ENDIF
       CALL QSETUP(0,ADMLIN,-2,ADMLIN)
      RETURN
      END
      SUBROUTINE LOX(S,GOX,IUNIT,IUNITP,IGONE)
C       Handles lox piping logic
      COMPLEX GOX(25),S
      COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
      INTEGER SEGMN(25),SECIN(75,25),NOLINE(25),IENG(25),ITANK(25),
     *        LOPOLD(25),LOPEND(25)
      REAL KMAN(25),KTANK(25),LFLOW(25),L(75,25)
      COMMON /PARAM/MLINE,SPLIT(25),A(25),CMAN(25),CTANK(25),
     *              DENS(25),KMAN,KTANK,LFLOW,VOL(25),VOLMF(25),
```

```
     *                AREA(75,25),DIA(75,25),L,PIND(75,25),
     *                PCAP(75,25),AVGK(25),
     *                SEGMN,SECIN,NOLINE,IENG,ITANK,LOPOLD,LOPEND
      COMMON /FOPIPE/PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),
     *              PIPE4(75,25),PIPE5(75,25)
      CHARACTER*24 LOXIN,NAMLIN(2)
      COMMON /WCAOUT/NAMLIN
      CHARACTER*1 ANS
      IF(IGONE.EQ.2) THEN
       WRITE(*,'(A\)')' Is the lox file name LOX.RLN? (Y/N) '
       READ(*,'(A)')ANS
       IF(ANS.NE.'N'.AND.ANS.NE.'n')  THEN
        OPEN(UNIT=IUNIT,FILE='LOX.RLN')
        NAMLIN(2)='LOX.RLN'
       ELSE
        WRITE(*,'(A\)')'  Enter name of file with lox line data '
        READ(*,'(A)')LOXIN
        OPEN(UNIT=IUNIT,FILE=LOXIN)
        NAMLIN(2)=LOXIN
       ENDIF
       OPEN(IUNITP,FORM='UNFORMATTED')
      ENDIF
      CALL FULOX(S,GOX,SEGMN,SECIN,PIPE1,PIPE2,PIPE3,PIPE4,PIPE5,
     *  A,AREA,AVGK,CMAN,CTANK,DENS,DIA,IENG,IGONE,ITANK,
     *  IUNIT,IUNITP,KMAN,KTANK,L,LOPEND,LOPOLD,LFLOW,MLINE,NOLINE,PCAP,
     *  PIND,SPLIT,VOL,VOLMF,2)
      RETURN
      END
      SUBROUTINE MODENG(IUNIT,NAMENG)
C       Modifies engine parameters
      COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
      CHARACTER*24 NAMENG
      CHARACTER*8 NAME
      CHARACTER*1 ANS
      CHARACTER*8 VARL(3),VARU(3)
      DATA VARL/'tflow   ','pchmb   ','dpror   '/
      DATA VARU/'TFLOW   ','PCHMB   ','DPROR   '/
      DO 25 J=1,MENG
       WRITE(*,'(A,A,I3,A\)')' Do you wish to change flow conditions ',
     *                 'for engine #',J,'? '
       READ(*,'(A)')ANS
       IF(ANS.NE.'Y'.AND.ANS.NE.'y')  GO TO 25
   21  CONTINUE
       WRITE(*,*)' '
       WRITE(*,*)'   VARIABLE NAMES AND VALUES'
       WRITE(*,*)' '
       WRITE(*,'(A,1PE15.5)')' TFLOW - total mass flow (lbm/sec)      ',
     *                       TFLOW(J)
       WRITE(*,'(A,1PE15.5)')' PCHMB - chamber pressure (lbf/ft^2)    ',
     *                       PCHMB(J)
       WRITE(*,'(A,1PE15.5)')' DPROR - orfice pressure drop (lbf/ft^2)',
     *                       DPROR(J)
       WRITE(*,*)' '
```

```
      WRITE(*,*)'   Enter variable name and new value, or'
      WRITE(*,*)'         # to print variable names & values, or'
      WRITE(*,*)'         END when all changes have been made'
      WRITE(*,*)' '
   22 CONTINUE
      WRITE(*,'(A\)')'   Enter variable name and new value, END, or # '
      CALL ZREAD(NAME,VALUE)
      IF(NAME.EQ.'#')  GO TO 21
      IF(NAME.EQ.'END'.OR.NAME.EQ.'end')  GO TO 25
      DO 23 II=1,3
       I=II
       IF(NAME.EQ.VARU(I).OR.NAME.EQ.VARL(I))  GO TO 24
   23 CONTINUE
      WRITE(*,*)'     Invalid name, try again'
      GO TO 21
   24 CONTINUE
      IF(I.EQ. 1)  TFLOW(J)=VALUE
      IF(I.EQ. 2)  PCHMB(J)=VALUE
      IF(I.EQ. 3)  DPROR(J)=VALUE
      PMRAT(J)=PCHMB(J)/TFLOW(J)
      GO TO 22
   25 CONTINUE
      WRITE(*,'(A\)')' Do you wish to save these changes? Y or N '
      READ(*,'(A)')ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')  RETURN
      WRITE(*,'(A,A,A\)')' Do you wish to use file ',NAMENG,
     *                   '? Y or N '
      READ(*,'(A)')ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')  THEN
       WRITE(*,'(A\)')' Enter name of file to use '
       READ(*,'(A)')NAMENG
       CLOSE(UNIT=IUNIT)
       OPEN(UNIT=IUNIT,FILE=NAMENG)
      ELSE
       WRITE(*,'(A,A,A\)')' Do you wish to rewind ',NAMENG,
     *                    '? Y or N '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  REWIND IUNIT
      ENDIF
      WRITE(IUNIT,'(I5)')MENG
      DO 26 J=1,MENG
       WRITE(IUNIT,'(1P3E15.5)')TFLOW(J),PCHMB(J),DPROR(J)
   26 CONTINUE
      RETURN
      END
      SUBROUTINE MODIFY(TITL,SEGMN,SECIN,PIPE1,PIPE2,PIPE3,
     *  PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
     *  A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
     *  AVGK,MLINE)
C       Allows modifications to input data
      COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
      COMMON /TANK/MTANK
      CHARACTER*24 NAMLIN(2)
```

```
      COMMON /WCAOUT/NAMLIN
      REAL SPLIT(25),AVGK(25)
      REAL AREA(75,25),DIA(75,25),L(75,25),PIND(75,25),
     *     PCAP(75,25)
      REAL PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),PIPE4(75,25),
     *     PIPE5(75,25)
      INTEGER SEGMN(25),SECIN(75,25)
      INTEGER ITANK(25),IENG(25),LOPOLD(25),LOPEND(25),NOLINE(25)
      REAL A(25),CTANK(25),DENS(25),KTANK(25),CMAN(25),KMAN(25),
     *     LFLOW(25),VOL(25),VOLMF(25)
      CHARACTER*20 TITL
      CHARACTER*1 ANS
    1 FORMAT(1PE15.6)
    2 FORMAT(I5,1P5E15.6)
    3 FORMAT(' This segment is a bend of',1PE13.5,' deg and radius of',
     *       E13.5)
    4 FORMAT(' This segment is straight ',1PE13.5,' diameter pipe ',
     *       E13.5,' ft. long')
    5 FORMAT(' This segment is a manifold with',1PE13.5,' vol.',
     *       E13.5,' bulk modulus')
    6 FORMAT(' This segment is a pump with length =',1PE13.5,' dia =',
     *       E13.5/5X,'dp/dm =',E13.5,' capacitance =',E13.5,
     *       ' inductance =',E13.5)
    7 FORMAT(' This segment is a tuned pipe ',1PE13.5,' long & dia =',
     *       E13.5)
    8 FORMAT(' This segment is a Helmholtz resonator with'/5X,'length ='
     *       ,1PE13.5,' dia =',E13.5,' and vol =',E13.5)
    9 FORMAT(' This segment is a parallel resonator with'/5X,'length =',
     *       1PE13.5,' dia =',E13.5,' and vol =',E13.5)
   10 FORMAT(' This segment is a',1PE13.5,' long inline acc. with',
     *       ' diameter of',E13.5)
      IF(IUNIT.EQ.11)  THEN
       NAMNAM=1
      ELSE
       NAMNAM=2
      ENDIF
      WRITE(*,'(A\)')' Do you wish to change tank parameters? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       CALL MODTAN(MTANK,VOL,LFLOW,KTANK,DENS,A,CTANK)
      ENDIF
      WRITE(*,'(A\)')' Do you wish to change the pipe layout? '
      READ(*,'(A)')ANS
      IF(ANS.NE.'Y'.AND.ANS.NE.'y')  GO TO 28
      IP=0
      DO 27 M=1,MLINE
       IP=IP+1
       IT=ITANK(M)
       DO 26 IPP=IP,IP+SPLIT(M)
        I=0
        ISEGMN=SEGMN(IPP)
        DO 25 II=1,SEGMN(IPP)
         I=I+1
```

```
      IF(SECTN(I,IPP).EQ.0)  THEN
       WRITE(*,3)PIPE2(I,IPP),PIPE1(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.1)  THEN
       WRITE(*,4)PIPE2(I,IPP),PIPE1(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.2)  THEN
       WRITE(*,10)PIPE1(I,IPP),PIPE2(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.3)  THEN
       WRITE(*,7)PIPE1(I,IPP),PIPE2(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.4)  THEN
       WRITE(*,8)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.5)  THEN
       WRITE(*,9)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.6)  THEN
       WRITE(*,6)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP),
     *           PIPE4(I,IPP),PIPE5(I,IPP)
      ELSEIF(SECTN(I,IPP).EQ.7)  THEN
       WRITE(*,5)PIPE1(I,IPP),PIPE2(I,IPP)
      ENDIF
      WRITE(*,'(A\)')' You may keep (K), modify (Y), delete (D),',
     *       ' add before (B), or add after (A)? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'A'.OR.ANS.EQ.'a')  THEN
       I=I+1
       DO 21 III=ISEGMN,I,-1
        PIPE1(III+1,IPP)=PIPE1(III,IPP)
        PIPE2(III+1,IPP)=PIPE2(III,IPP)
        PIPE3(III+1,IPP)=PIPE3(III,IPP)
        PIPE4(III+1,IPP)=PIPE4(III,IPP)
        PIPE5(III+1,IPP)=PIPE5(III,IPP)
        L(III+1,IPP)=L(III,IPP)
        DIA(III+1,IPP)=DIA(III,IPP)
        AREA(III+1,IPP)=AREA(III,IPP)
        PCAP(III+1,IPP)=PCAP(III,IPP)
        PIND(III+1,IPP)=PIND(III,IPP)
        SECTN(III+1,IPP)=SECTN(III,IPP)
   21  CONTINUE
       ISEGMN=ISEGMN+1
       GO TO 24
      ELSEIF(ANS.EQ.'B'.OR.ANS.EQ.'b')  THEN
       DO 22 III=ISEGMN,I,-1
        PIPE1(III+1,IPP)=PIPE1(III,IPP)
        PIPE2(III+1,IPP)=PIPE2(III,IPP)
        PIPE3(III+1,IPP)=PIPE3(III,IPP)
        PIPE4(III+1,IPP)=PIPE4(III,IPP)
        PIPE5(III+1,IPP)=PIPE5(III,IPP)
        L(III+1,IPP)=L(III,IPP)
        DIA(III+1,IPP)=DIA(III,IPP)
        AREA(III+1,IPP)=AREA(III,IPP)
        PCAP(III+1,IPP)=PCAP(III,IPP)
        PIND(III+1,IPP)=PIND(III,IPP)
        SECTN(III+1,IPP)=SECTN(III,IPP)
   22  CONTINUE
       ISEGMN=ISEGMN+1
```

```
            GO TO 24
           ELSEIF(ANS.EQ.'D'.OR.ANS.EQ.'d')  THEN
            DO 23 III=I,ISEGMN
             PIPE1(III,IPP)=PIPE1(III+1,IPP)
             PIPE2(III,IPP)=PIPE2(III+1,IPP)
             PIPE3(III,IPP)=PIPE3(III+1,IPP)
             PIPE4(III,IPP)=PIPE4(III+1,IPP)
             PIPE5(III,IPP)=PIPE5(III+1,IPP)
             L(III,IPP)=L(III+1,IPP)
             DIA(III,IPP)=DIA(III+1,IPP)
             AREA(III,IPP)=AREA(III+1,IPP)
             PCAP(III,IPP)=PCAP(III+1,IPP)
             PIND(III,IPP)=PIND(III+1,IPP)
             SECIN(III,IPP)=SECIN(III+1,IPP)
      23     CONTINUE
             I=I-1
             ISEGMN=ISEGMN-1
             GO TO 25
           ELSEIF(ANS.NE.'Y'.AND.ANS.NE.'y')  THEN
            GO TO 25
           ENDIF
      24   CONTINUE
           WRITE(*,*)' Specify 0 for BEND,            1 for STRAIGHT pipe,'
           WRITE(*,*)'         2 for INLINE ACCUM.,   3 for TUNED STUB,'
           WRITE(*,*)'         4 for HELMHOLTZ RES.,  5 for PARALLEL RES.'
           WRITE(*,*)'         6 for PUMP,            7 for MANIFOLD'
           READ(*,*) SECT
           IF(SECT.LT.0.OR.SECT.GT.7)  GO TO 24
           SECIN(I,IPP)=SECT
           IF(SECT.EQ.0)  THEN
      C         bend in pipe
            WRITE(*,*)' RADIUS of bend along CL(ft), ANGLE of bend(deg),'
            WRITE(*,*)' DIAMETER(ft), and LENGTH(ft) beyond bend of pipe'
            READ(*,*)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP),PIPE4(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
        *              PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
        *              AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
        *              AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.1)  THEN
      C       straight section
            WRITE(*,*)' Specify LENGTH (ft) and DIAMETER (ft) of segment'
            READ(*,*) PIPE1(I,IPP),PIPE2(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
        *              PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
        *              AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
        *              AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.2)  THEN
      C       inline accumulator
            WRITE(*,*)' Specify LENGTH (ft) & DIAMETER (ft) of',
        *             ' accumulator '
            READ(*,*) PIPE1(I,IPP),PIPE2(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
        *              PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
```

```
     *                AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
     *                AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.3)  THEN
     C          tuned stub
            WRITE(*,*)' Specify LENGTH (ft) & DIAMETER (ft) of tuned stub'
            READ(*,*)PIPE1(I,IPP),PIPE2(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
     *                PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
     *                AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
     *                AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.4)  THEN
     C          helmholtz resonator
            WRITE(*,*)' Specify LENGTH (ft), DIAMETER (ft) ,VOLUME ',
     *                '(ft^3) of Helmholtz Resonator'
            READ(*,*)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
     *                PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
     *                AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
     *                AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.5)  THEN
     C          parallel resonator
            WRITE(*,*)' Specify LENGTH (ft), DIAMETER (ft) ,VOLUME ',
     *                '(ft^3) of Parallel Resonator'
            READ(*,*)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
     *                PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
     *                AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
     *                AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.6)  THEN
     C          pump
            WRITE(*,*)' Specify LENGTH (ft), DIAMETER (ft) ,dp/dm, CAP.',
     *                ' & IND. of pump'
            READ(*,*)PIPE1(I,IPP),PIPE2(I,IPP),PIPE3(I,IPP),
     *               PIPE4(I,IPP),PIPE5(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
     *                PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
     *                AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
     *                AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
           ELSEIF(SECT.EQ.7)  THEN
     C          manifold
            WRITE(*,*)' Specify VOLUME (ft^3) and BULK MODULUS (lbf/ft^2)'
            READ(*,*) PIPE1(I,IPP),PIPE2(I,IPP)
            CALL RTYPE(SECIN(I,IPP),PIPE1(I,IPP),PIPE2(I,IPP),
     *                PIPE3(I,IPP),PIPE4(I,IPP),PIPE5(I,IPP),L(I,IPP),
     *                AREA(I,IPP),DIA(I,IPP),PIND(I,IPP),PCAP(I,IPP),
     *                AVGK(M),DENS(IT),CMAN(IPP),KMAN(IPP),VOLMF(IPP))
            ENDIF
     25    CONTINUE
           SEGMN(IPP)=ISEGMN
     26    CONTINUE
           IF(SPLIT(M).NE.0.0)  THEN
            WRITE(*,'(A,I3)')' Maximum no. of iterations is set at ',
     *                       LOPOLD(M)
```

```
      WRITE(*,'(A\)')' Do you wish to change it? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       WRITE(*,'(A\)')' Enter maximum no. of iterations '
       READ(*,*)LOPOLD(M)
      ENDIF
      LOPEND(M)=LOPOLD(M)
      IP=IP+SPLIT(M)
     ENDIF
  27 CONTINUE
  28 CONTINUE
     WRITE(*,'(A\)')' Do you wish to save these changes? Y or N '
     READ(*,'(A)')ANS
     IF(ANS.NE.'Y'.AND.ANS.NE.'y')  RETURN
     WRITE(*,'(A,A,A\)')' Do you wish to use file ',NAMLIN(NAMNAM),
    *                    '? Y or N '
     READ(*,'(A)')ANS
     IF(ANS.NE.'Y'.AND.ANS.NE.'y')  THEN
      WRITE(*,'(A\)')' Enter name of file to use '
      READ(*,'(A)')NAMLIN(NAMNAM)
      CLOSE(UNIT=IUNIT)
      OPEN(UNIT=IUNIT,FILE=NAMLIN(NAMNAM))
     ELSE
      WRITE(*,'(A,A,A\)')' Do you wish to rewind ',NAMLIN(NAMNAM),
    *                     '? Y or N '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  REWIND IUNIT
     ENDIF
     IP=0
     WRITE(IUNIT,'(A)')TITL
     WRITE(IUNIT,2)MTANK
     DO 29 M=1,MTANK
      WRITE(IUNIT,1)VOL(M)
      WRITE(IUNIT,1)LFLOW(M)
      WRITE(IUNIT,1)KTANK(M)
      WRITE(IUNIT,1)DENS(M)
  29 CONTINUE
     WRITE(IUNIT,2)MLINE
     DO 33 M=1,MLINE
      IP=IP+1
      WRITE(IUNIT,2)ITANK(M)
      WRITE(IUNIT,2)IENG(IP)
      WRITE(IUNIT,2)SEGMN(IP)
      WRITE(IUNIT,2)SPLIT(M)
      DO 30 J=1,SEGMN(IP)
       WRITE(IUNIT,2)SECTN(J,IP),PIPE1(J,IP),PIPE2(J,IP),PIPE3(J,IP),
    *                PIPE4(J,IP),PIPE5(J,IP)
  30  CONTINUE
      IF(SPLIT(M).EQ.0)  GO TO 33
      DO 32 K=1,SPLIT(M)
       IP=IP+1
       WRITE(IUNIT,2)SEGMN(IP)
       WRITE(IUNIT,2)NOLINE(IP)
```

```
      IF(I.EQ. 2)  LFLOW(J)=VALUE
      IF(I.EQ. 3)  KTANK(J)=VALUE
      IF(I.EQ. 4)  DENS(J)=VALUE
      A(J)=SQRT(GRAV*KTANK(J)/DENS(J))
      CTANK(J)=DENS(J)*VOL(J)/KTANK(J)
      GO TO 22
   25 CONTINUE
      RETURN
      END
      SUBROUTINE PIPPLOT(SEGMN,SECTN,PIPE1,PIPE2,PIPE3,PIPE4,ILOX,
     *                   ITANK,IENG)
C       Supervises plot of piping layout
      COMMON /ARCCON/XC,YC,RAD,ANG,ANGLE
      COMMON /PIPPXY/X,XH,XL,Y,YH,YL,XMIN,XMAX,YMIN,YMAX,SINA,COSA
      EXTERNAL XFUN,YFUN
      INTEGER*2 SEGMN,SECTN(75),ITYPE(200)
      REAL PIPE1(75),PIPE2(75),PIPE3(75),PIPE4(75)
      REAL POINT(8,200),XP(2),YP(2)
      ANG=0.0
      ANGLE=0.0
      COSA=1.0
      SINA=0.0
      X=0.0
      XH=0.0
      XL=0.0
      Y=0.0
      IF(SECTN(1).EQ.0)  THEN
       YH=Y+0.5*PIPE3(1)
       YL=Y-0.5*PIPE3(1)
      ELSEIF(SECTN(1).GE.3.AND.SECTN(1).LE.5)  THEN
       IF(SECTN(2).EQ.0)  THEN
        YH=Y+0.5*PIPE3(2)
        YL=Y-0.5*PIPE3(2)
       ELSE
        YH=Y+0.5*PIPE2(2)
        YL=Y-0.5*PIPE2(2)
       ENDIF
      ELSE
       YH=Y+0.5*PIPE2(1)
       YL=Y-0.5*PIPE2(1)
      ENDIF
      J=0
      XMIN=0.0
      XMAX=0.0
      YMIN=AMIN1(Y,YL,YH)
      YMAX=AMAX1(Y,YL,YH)
      DO 21 I=1,SEGMN
       IF(SECTN(I).EQ.0)  THEN
C        bend
        CALL BNSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I),PIPE3(I),PIPE4(I))
       ELSEIF(SECTN(I).EQ.1)  THEN
C        straight section
        CALL STSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I))
```

```
      ELSEIF(SECTN(I).EQ.2)  THEN
C        inline accumulator
       CALL STSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I))
      ELSEIF(SECTN(I).EQ.3)  THEN
C        tuned stub
       CALL TSSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I))
      ELSEIF(SECTN(I).EQ.4)  THEN
C        helmholtz resonator
       CALL HHSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I),PIPE3(I))
      ELSEIF(SECTN(I).EQ.5)  THEN
C        parallel resonator
       CALL PLSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I),PIPE3(I))
      ELSEIF(SECTN(I).EQ.6)  THEN
C        pump
       CALL STSECT(J,ITYPE,POINT,PIPE1(I),PIPE2(I))
      ENDIF
   21 CONTINUE
      XRANGE=XMAX-XMIN
      YRANGE=YMAX-YMIN
      XMIN=XMIN-0.05*XRANGE
      XMAX=XMAX+0.05*XRANGE
      YMIN=YMIN-0.05*YRANGE
      YMAX=YMAX+0.05*YRANGE
      CALL UPPERW(XMIN,YMIN,XMAX,YMAX,ILOX,ITANK,IENG)
      DO 22 I=1,J
       IF(ITYPE(I).EQ.0)  THEN
C        bend
        XC=POINT(1,I)
        YC=POINT(2,I)
        X1=POINT(3,I)
        Y1=POINT(4,I)
        RAD=POINT(5,I)
        IF(X1.GT.Y1)  THEN
         X1=3.14159+X1
         Y1=3.14159+Y1
         CALL QCURV(XFUN,YFUN,Y1,X1)
        ELSE
         CALL QCURV(XFUN,YFUN,X1,Y1)
        ENDIF
       ELSE
C        all except bend
        X0=POINT(1,I)
        Y0=POINT(2,I)
        X1=POINT(3,I)
        Y1=POINT(4,I)
        X2=POINT(5,I)
        Y2=POINT(6,I)
        X3=POINT(7,I)
        Y3=POINT(8,I)
        XP(1)=X0
        YP(1)=Y0
        XP(2)=X1
        YP(2)=Y1
```

```
      CALL QTABL(1,2,XP,YP)
      XP(1)=X2
      YP(1)=Y2
      XP(2)=X3
      YP(2)=Y3
      CALL QTABL(1,2,XP,YP)
      XP(1)=X0
      YP(1)=Y0
      XP(2)=X2
      YP(2)=Y2
      CALL QTABL(1,2,XP,YP)
      XP(1)=X1
      YP(1)=Y1
      XP(2)=X3
      YP(2)=Y3
      CALL QTABL(1,2,XP,YP)
     ENDIF
   22 CONTINUE
     RETURN
     END
     SUBROUTINE PLOTSU(X,Y,Z,XF,YF,ZF,IPTS,IXMAX,IYMAX,IFULOX,MLINE)
C      Supervises the surface plot
     INTEGER SEGMN,SECIN(150)
     COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
    *              NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
    *              SEGMN,SECIN
     CHARACTER*40 TITLE
     CHARACTER*20 TITL
     INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
     CHARACTER*2 AP
     COMMON /WCATIT/TITLE,TITL,IHR,IMIN,AP,IYR,IMON,IDAY
     COMMON /FACTOR/SFAC
     INTEGER*4 IXMAX,IYMAX
     REAL XF(IXMAX),YF(IYMAX),ZF(IXMAX,IYMAX)
     REAL X(IPTS,IYMAX),Y(IPTS,IYMAX),Z(IPTS,IYMAX)
     INTEGER*2 IWRK1(640),IWRK2(640)
     CHARACTER*1 ANS
     CHARACTER*38 PIPING
     CHARACTER*45 LEGEND
     CHARACTER*58 LEGENDR,LEGENDH
     CHARACTER*4 FOPIPE(2)
     DATA FOPIPE/'FUEL',' LOX'/
     DATA LEGEND/'Pressure Transfer Function = f(freq,distance)'/
     DATA LEGENDR/'Pressure Transfer Function = f(freq(rad/sec),distanc
    *e(ft))'/
     DATA LEGENDH/' Pressure Transfer Function = f(freq(Hertz),distance
    *(ft)) '/
     DATA ASPECT/1.35/
     DATA ICOLR/4/,IFIL/3/,ILIN/1/
    1 FORMAT(' Current view is PHI =',F8.3,'   THETA =',F8.3)
    2 FORMAT(' Current BACKGROUND COLOR = ',I2,' LINE COLOR = ',I2,
    *       ' FILL COLOR = ',I2)
    3 FORMAT(1X,A4,' Piping - Tank #',I3,3X,'Engine #',I3)
```

```
      IGOO=0
      CALL QRMODE(MODET,NCOLT)
      CALL QVIDBD(IBOARD)
      IF(IBOARD.LT.1.OR.IBOARD.GT.3)  THEN
       WRITE(*,*)' Graphics board not installed!'
       RETURN
      ENDIF
      IF(IBOARD.EQ.1)  MODE=6
      IF(IBOARD.EQ.2)  MODE=16
      IF(IBOARD.EQ.3)  MODE=18
      IWIRE=0
      IF(IBOARD.NE.1)  THEN
       WRITE(*,'(A\)')' Do you want a wire-frame drawing? '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  IWIRE=1
      ENDIF
   21 CONTINUE
      CALL GETZF(Y,Z,YF,ZF,IXMAX,IYMAX,IPTS,JPTS,IFULOX,IPLOT,IGOO)
      IF(IPLOT.EQ.0)  RETURN
      WRITE(PIPING,3)FOPIPE(IFULOX),NTANK(IPLOT),NENG(IPLOT)
      XMIN=XF(1)
      XMAX=XF(IPTS)
      YMIN=YF(1)
      YMAX=YF(JPTS)
      ZMIN=ZF(1,1)
      ZMAX=ZF(1,1)
      DO 22 J=1,JPTS
      DO 22 I=1,IPTS
       IF(ZMIN.GT.ZF(I,J))  ZMIN=ZF(I,J)
       IF(ZMAX.LT.ZF(I,J))  ZMAX=ZF(I,J)
   22 CONTINUE
      YLEN=YF(JPTS)-YF(1)
      XLEN=XF(IPTS)-XF(1)
      ZLEN=ZMAX-ZMIN
      IF(XLEN.EQ.0.0.OR.YLEN.EQ.0.0.OR.ZLEN.EQ.0.0)  STOP 1
      XYZLEN=AMAX1(XLEN,YLEN,ZLEN)
      XFAC=XYZLEN/XLEN
      XINV=1.0/XFAC
      YFAC=XYZLEN/YLEN
      YINV=1.0/YFAC
      ZFAC=XYZLEN/ZLEN
      ZINV=1.0/ZFAC
      DO 23 J=1,JPTS
      DO 23 I=1,IPTS
       X(I,J)=XF(I)*XFAC
       Y(I,J)=YF(J)*YFAC
       Z(I,J)=ZF(I,J)*ZFAC
   23 CONTINUE
      XMIN=XMIN*XFAC
      XMAX=XMAX*XFAC
      YMIN=YMIN*YFAC
      YMAX=YMAX*YFAC
      ZMIN=ZMIN*ZFAC
```

```
      ZMAX=ZMAX*ZFAC
      XMAJ=0.2*(XMAX-XMIN)
      YMAJ=0.2*(YMAX-YMIN)
      ZMAJ=0.2*(ZMAX-ZMIN)
      P=-45.0
      T=30.0
      CALL Q3DROT(X,Y,Z,IPTS,JPTS,P,T)
   24 CONTINUE
      CALL QSMODE(MODE)
      IF(IBOARD.NE.1)  CALL QPREG(0,ICOLR)
      CALL WINDOW(MODE,ASPECT,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX)
      CALL Q3DXAX(XMIN,XMAX,XMAJ,0,-1,2,YMIN,YMAX,ZMIN,XINV)
      CALL Q3DYAX(YMIN,YMAX,YMAJ,0,-1,2,XMAX,XMIN,ZMIN,YINV)
      CALL Q3DZAX(ZMIN,ZMAX,ZMAJ,0,-1,2,XMIN,YMIN,ZINV)
      IF(MODE.EQ.6)  THEN
       CALL QPTXT(40,TITLE,7,17,23)
       CALL QPTXT(45,LEGEND,7,15,22)
        CALL QPTXT(38,PIPING,7,18,21)
      ELSEIF(MODE.EQ.16)  THEN
       CALL QPTXT(40,TITLE,7,17,23)
       IF(SFAC.EQ.1.0)  THEN
        CALL QPTXT(58,LEGENDR,7,8,22)
       ELSE
        CALL QPTXT(58,LEGENDH,7,8,22)
       ENDIF
        CALL QPTXT(38,PIPING,7,18,21)
      ELSE
       CALL QPTXT(40,TITLE,7,17,27)
       IF(SFAC.EQ.1.0)  THEN
        CALL QPTXT(58,LEGENDR,7,8,26)
       ELSE
        CALL QPTXT(58,LEGENDH,7,8,26)
        CALL QPTXT(38,PIPING,7,18,25)
       ENDIF
      ENDIF
      IF(IBOARD.EQ.1.OR.IWIRE.EQ.1)  THEN
       CALL Q3DSTK(X,Y,IPTS,JPTS,IWRK1,IWRK2,640,1)
      ELSE
       CALL Q3DFIL(X,Y,IPTS,JPTS,IFIL,ILIN)
      ENDIF
   25 CONTINUE
      CALL QONKEY(IKEY)
      IF(IKEY.EQ.0)  GO TO 25
      CALL QINKEY(IEXTEN,IKEY)
      CALL QSMODE(MODET)
      IGO=0
      WRITE(*,1)P,T
      WRITE(*,'(A\)')' Do you wish another view? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       WRITE(*,'(A\)')'  Enter new viewing angles PHI & THETA. '
       READ(*,*)P,T
       CALL Q3DINV(X,Y,Z,IPTS,JPTS)
```

```
      CALL Q3DROT(X,Y,Z,IPTS,JPTS,P,T)
      IGO=1
     ENDIF
     IF(IBOARD.NE.1)  THEN
      WRITE(*,2)ICOLR,ILIN,IFIL
      WRITE(*,'(A\)')' Do you wish another color? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       WRITE(*,*)' Enter color number (0-63) for BACKGROUND, LINE,
     * and FILL '
       WRITE(*,*)'   4,1,3 will give the default colors '
       WRITE(*,'(A\)')'    0,7,0 will give black & white '
       READ(*,*)ICOLR,ILIN,IFIL
       IGO=1
      ENDIF
      IWR=0
      IF(IWIRE.EQ.0)  THEN
       WRITE(*,'(A\)')' Do you want a wire-frame drawing? '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
        IWR=1
        IGO=1
       ENDIF
      ELSE
       WRITE(*,'(A\)')' Do you want a filled drawing? '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
        IWR=2
        IGO=1
       ENDIF
      ENDIF
      IF(IWR.EQ.1)  IWIRE=1
      IF(IWR.EQ.2)  IWIRE=0
     ENDIF
     IF(IGO.NE.0)  GO TO 24
     IF(MLINE.GT.1)  THEN
      WRITE(*,'(A\)')' Do you want another surface plot? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  GO TO 21
     ENDIF
     RETURN
     END
     SUBROUTINE PLSECT(J,ITYPE,POINT,LEN,DIA,VOL)
C      Computes plot coordinates for parallel resonator
     COMMON /PIPPXY/X,XH,XL,Y,YH,YL,XMIN,XMAX,YMIN,YMAX,SINA,COSA
     COMMON /ARCCON/XC,YC,RAD,ANG,ANGLE
     REAL LEN,POINT(8,200)
     INTEGER*2 ITYPE(200)
     XOLD=X
     XHOLD=XH
     XLOLD=XL
     YOLD=Y
     YHOLD=YH
```

```
      YLOLD=YL
      ANGOLD=ANG
      ANGSAV=ANGLE
      SINOLD=SINA
      COSOLD=COSA
      DIAM=SQRT((XH-XL)**2+(YH-YL)**2)
      CALL STSECT(J,ITYPE,POINT,DIA,DIAM)
      XC=0.5*(XHOLD+XH)
      XHC=XHOLD
      XLC=XL
      YC=0.5*(YHOLD+YH)
      YHC=YHOLD
      YLC=YL
      PLEN=LEN-2.0*DIA
      PDIA=(VOL-2.0*DIA*DIAM)/PLEN
      CALL STSECT(J,ITYPE,POINT,PLEN,PDIA)
      CALL STSECT(J,ITYPE,POINT,DIA,DIAM)
      XSAV=X
      XHSAV=XH
      XLSAV=XL
      YSAV=Y
      YHSAV=YH
      YLSAV=YL
      SINA=COSOLD
      COSA=-SINOLD
      RADIUS=DIA
      TURN=-90.0
      SIDE=LEN-5.0*DIA
      ANG=ANG+1.5708
      ANGLE=ANGLE+90.0
      X=XC
      Y=YC
      XH=XHC
      XL=XLC
      YH=YHC
      YL=YLC
      CALL BNSECT(J,ITYPE,POINT,RADIUS,TURN,DIA,DIA)
      CALL STSECT(J,ITYPE,POINT,SIDE,DIA)
      CALL BNSECT(J,ITYPE,POINT,RADIUS,TURN,DIA,DIA)
      X=XSAV
      Y=YSAV
      XH=XHSAV
      XL=XLSAV
      YH=YHSAV
      YL=YLSAV
      ANG=ANGOLD
      ANGLE=ANGSAV
      SINA=SINOLD
      COSA=COSOLD
      RETURN
      END
      SUBROUTINE PLTCON(X,Y,Z,XF,YF,ZF,IPTS,IXMAX,IYMAX,IFULOX,MLINE)
C       Supervises plot of contour plot
```

```
      INTEGER SEGMN,SECTN(150)
      COMMON /SETUP/PIPE1(150),PIPE2(150),PIPE3(150),PIPE4(150),
     *              NENG(25),NTANK(25),NLINE(25),NSP(25),NEND(25),ILINE,
     *              SEGMN,SECTN
      CHARACTER*40 TITLE
      CHARACTER*20 TITL
      INTEGER*2 IHR,IMIN,IYR,IMON,IDAY
      CHARACTER*2 AP
      COMMON /WCATIT/TITLE,TITL,IHR,IMIN,AP,IYR,IMON,IDAY
      COMMON /FACTOR/SFAC
      INTEGER*4 IXMAX,IYMAX
      REAL XF(IXMAX),YF(IYMAX),ZF(IXMAX,IYMAX)
      REAL X(IPTS),Y(IYMAX),Z(IPTS,IYMAX),CONS(10)
      INTEGER*2 LABL(10)
      CHARACTER*38 PIPING
      CHARACTER*4 FOPIPE(2)
      DATA FOPIPE/'FUEL',' LOX'/
      DATA ASPECT/1.35/
      DATA LABL/1,0,0,0,1,0,0,0,1,0/
      DATA ICOLR/4/,IFIL/3/,ILIN/1/
    1 FORMAT(1X,A4,' Piping - Tank #',I3,3X,'Engine #',I3)
    2 FORMAT(' Current BACKGROUND COLOR = ',I2,' LINE COLOR = ',I2,
     *       ' FILL COLOR = ',I2)
      IGO=0
      CALL QRMODE(MODET,NCOLT)
      CALL QVIDBD(IBOARD)
      IF(IBOARD.LT.1.OR.IBOARD.GT.3)  THEN
       WRITE(*,*)' Graphics board not installed!'
       RETURN
      ENDIF
      IF(IBOARD.EQ.1)  MODE=6
      IF(IBOARD.EQ.2)  MODE=16
      IF(IBOARD.EQ.3)  MODE=18
   21 CONTINUE
      CALL GETZF(Y,Z,YF,ZF,IXMAX,IYMAX,IPTS,JPTS,IFULOX,IPLOT,IGO)
      IF(IPLOT.EQ.0)  RETURN
      WRITE(PIPING,1)FOPIPE(IFULOX),NTANK(IPLOT),NENG(IPLOT)
      XMIN=XF(1)
      XMAX=XF(IPTS)
      YMIN=YF(1)
      YMAX=YF(JPTS)
      ZMIN=ZF(1,1)
      ZMAX=ZF(1,1)
      DO 22 J=1,JPTS
       Y(J)=YF(J)
      DO 22 I=1,IPTS
       IF(J.EQ.1)  X(I)=XF(I)
       Z(I,J)=ZF(I,J)
       IF(ZMIN.GT.Z(I,J))  ZMIN=Z(I,J)
       IF(ZMAX.LT.Z(I,J))  ZMAX=Z(I,J)
   22 CONTINUE
      ZLEN=0.1*(ZMAX-ZMIN)
      DO 23 I=1,9
```

```
      CONS(I)=I*ZLEN
   23 CONTINUE
      XMAJ=0.2*(XMAX-XMIN)
      YMAJ=0.2*(YMAX-YMIN)
   24 CONTINUE
      CALL QSMODE(MODE)
      IDEF=2
      IF(IBOARD.NE.1)  THEN
       IDEF=2
       CALL QPREG(0,ICOLR)
      ENDIF
      CALL QCTRDE(MODE,ILIN,IFIL,ILIN,1)
      JCOL1=100
      JCOL2=450
      JROW1=40
      IF(MODE.EQ.6)  JROW1=60
      JROW2=169
      IF(MODE.EQ.16)  JROW2=300
      IF(MODE.EQ.18)  JROW2=409
      XORG=XMIN
      YORG=YMIN
      YOVERX=1.0
      IOPT=0
      IF(MODE.NE.18)  THEN
       CALL QPTXT(40,TITLE,7,17,23)
       CALL QPTXT(38,PIPING,7,18,22)
      ELSE
       CALL QPTXT(40,TITLE,7,17,27)
       CALL QPTXT(38,PIPING,7,18,26)
      ENDIF
      CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,
     *           XORG,YORG,IOPT,YOVERX,ASPECT)
      CALL QXAXIS(XMIN,XMAX,XMAJ,0,-1,2)
      CALL QYAXIS(YMIN,YMAX,YMAJ,0,-1,2)
      IF(SFAC.EQ.1)  THEN
       CALL QPTXTA(17,'Frequency-rad/sec',7)
      ELSE
       CALL QPTXTA(17,' Frequency-Hertz ',7)
      ENDIF
      CALL QPTXTD(7,'X - ft.',7)
      CALL QCNTOU(ASPECT,X,Y,Z,CONS,LABL,IPTS,JPTS,9,IDEF)
   25 CONTINUE
      CALL QONKEY(IKEY)
      IF(IKEY.EQ.0)  GO TO 25
      CALL QINKEY(IEXTEN,IKEY)
      CALL QSMODE(MODET)
      IF(IBOARD.NE.1)  THEN
       WRITE(*,2)ICOLR,ILIN,IFIL
       WRITE(*,'(A\)')' Do you wish another color? '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
        WRITE(*,*)' Enter color number (0-63) for BACKGROUND, LINE,
     * and FILL '
```

```
            WRITE(*,*)'   4,1,3 will give the default colors '
            WRITE(*,'(A\)')'    0,7,7 will give black & white '
            READ(*,*)ICOLR,ILIN,IFIL
            GO TO 24
           ENDIF
          ENDIF
          IF(MLINE.GT.1)  THEN
           WRITE(*,'(A\)')' Do you want another contour plot? '
           READ(*,'(A)')ANS
           IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  GO TO 21
          ENDIF
          RETURN
          END
          SUBROUTINE RLINE(TITL,SEGMN,SECTN,PIPE1,PIPE2,PIPE3,
         *  PIPE4,PIPE5,L,AREA,DIA,PIND,PCAP,LOPEND,LOPOLD,SPLIT,IUNIT,
         *  A,CMAN,CTANK,DENS,KMAN,KTANK,LFLOW,VOL,VOLMF,NOLINE,IENG,ITANK,
         *  AVGK,MLINE)
C         Reads fuel or lox file.
          COMMON /EPARAM/MENG,TFLOW(25),PCHMB(25),DPROR(25),PMRAT(25)
          COMMON /TANK/MTANK
          REAL SPLIT(25),AVGK(25)
          REAL AREA(75,25),DIA(75,25),L(75,25),PIND(75,25),
         *     PCAP(75,25)
          REAL PIPE1(75,25),PIPE2(75,25),PIPE3(75,25),PIPE4(75,25),
         *     PIPE5(75,25)
          INTEGER SEGMN(25),SECTN(75,25)
          INTEGER ITANK(25),IENG(25),LOPOLD(25),LOPEND(25),NOLINE(25)
          REAL A(25),CMAN(25),CTANK(25),DENS(25),KMAN(25),KTANK(25),
         *     LFLOW(25),VOL(25),VOLMF(25)
          CHARACTER*20 TITL
          CHARACTER*1 ANS
          READ(IUNIT,'(A)')TITL
          CALL TANKNO(MTANK,VOL,LFLOW,KTANK,DENS,A,CTANK,IUNIT)
          READ(IUNIT,*)MLINE
          IF(MLINE.GT.25)  THEN
           WRITE(*,*)' Number of lines must be less than 25'
           STOP
          ENDIF
          IF(MLINE.LE.0)  MLINE=1
          DO 20 M=1,25
           IENG(M)=0
       20 CONTINUE
          M=0
          DO 24 MM=1,MLINE
           M=M+1
           READ(IUNIT,*)ITANK(MM),IENG(M)
           IF(ITANK(MM).GT.MTANK)  THEN
            WRITE(*,*)'  Invalid tank number.'
            STOP
           ENDIF
           IF(IENG(M).GT.MENG)  THEN
            WRITE(*,*)'  Invalid engine number.'
            STOP
```

```
      ENDIF
      IT=ITANK(MM)
      IE=IENG(M)
      LOPOLD(MM)=20
      LOPEND(MM)=1
      AVGK(MM)=0.0
      DIVAVG=0.0
      READ(IUNIT,*)SEGMN(M),SPLIT(MM)
      DO 21 I=1,SEGMN(M)
       READ(IUNIT,*)SECIN(I,M),PIPE1(I,M),PIPE2(I,M),PIPE3(I,M),
     *              PIPE4(I,M),PIPE5(I,M)
       IF(SECIN(I,M).NE.7)  GO TO 21
       AVGK(MM)=AVGK(MM)+PIPE2(I,M)
       DIVAVG=DIVAVG+1
   21 CONTINUE
      IF(SPLIT(MM).EQ.0)  THEN
       AVGK(MM)=KTANK(IT)
       NOLINE(M)=1
       GO TO 24
      ENDIF
C           split pipe
      DO 23 J=1,SPLIT(MM)
       M=M+1
       READ(IUNIT,*)SEGMN(M),NOLINE(M),IENG(M)
       IF(IENG(M).GT.MENG)  THEN
        WRITE(*,*)'  Invalid engine number.'
        STOP
       ENDIF
       IE=IENG(M)
       IF(NOLINE(M).EQ.0)  NOLINE(M)=1
       DO 22 I=1,SEGMN(M)
        READ(IUNIT,*)SECIN(I,M),PIPE1(I,M),PIPE2(I,M),PIPE3(I,M),
     *              PIPE4(I,M),PIPE5(I,M)
        IF(SECIN(I,M).NE.7)  GO TO 22
        AVGK(MM)=AVGK(MM)+PIPE2(I,M)*NOLINE(M)
        DIVAVG=DIVAVG+NOLINE(M)
   22   CONTINUE
   23  CONTINUE
      WRITE(*,'(A,I3)')' Max. no. of iterations is set at ',
     *                  LOPOLD(MM)
      WRITE(*,'(A\)')' Do you wish to change it? '
      READ(*,'(A)')ANS
      IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
       WRITE(*,'(A\)')' Enter maximum no. of iterations '
       READ(*,*)LOPOLD(MM)
      ENDIF
      LOPEND(MM)=LOPOLD(MM)
      IF(DIVAVG.LE.0.0)  DIVAVG=1.0
      AVGK(MM)=KTANK(IT)+AVGK(MM)/DIVAVG
   24 CONTINUE
      M=0
      DO 28 MM=1,MLINE
       M=M+1
```

```
      IT=ITANK(MM)
      IE=IENG(M)
      DO 25 I=1,SEGMN(M)
       CALL RTYPE(SECIN(I,M),PIPE1(I,M),PIPE2(I,M),
     *            PIPE3(I,M),PIPE4(I,M),PIPE5(I,M),L(I,M),AREA(I,M),
     *            DIA(I,M),PIND(I,M),PCAP(I,M),AVGK(MM),DENS(IT),
     *            CMAN(M),KMAN(M),VOLMF(M))
   25 CONTINUE
      IF(SPLIT(MM).EQ.0)  GO TO 28
      DO 27 J=1,SPLIT(MM)
       M=M+1
       IE=IENG(M)
       DO 26 I=1,SEGMN(M)
        CALL RTYPE(SECIN(I,M),PIPE1(I,M),PIPE2(I,M),
     *             PIPE3(I,M),PIPE4(I,M),PIPE5(I,M),L(I,M),AREA(I,M),
     *             DIA(I,M),PIND(I,M),PCAP(I,M),AVGK(MM),DENS(IT),
     *             CMAN(M),KMAN(M),VOLMF(M))
   26 CONTINUE
   27 CONTINUE
   28 CONTINUE
      RETURN
      END
      SUBROUTINE RTYPE(SECIN,PIPE1,PIPE2,PIPE3,PIPE4,PIPE5,L,AREA,DIA,
     *                 PIND,PCAP,AVGK,DENS,CMAN,KMAN,VOLMF)
C       Stores values for different types of piping
      INTEGER SECIN
      REAL L,KMAN
      DATA GRAV/32.2/,PI/3.141593/
      IF(SECIN.EQ.0)  THEN
       CALL BENDS(PIPE1,PIPE2,PIPE3,PIPE4,VALUE,DIME)
        AREAB=0.785398*DIME**2
        L=VALUE
        AREA=AREAB
        DIA=DIME
       ELSEIF(SECIN.EQ.1)  THEN
C            straight section
        VALUE=PIPE1
        DIME=PIPE2
        AREAB=0.785398*DIME**2
        L=VALUE
        AREA=AREAB
        DIA=DIME
       ELSEIF(SECIN.EQ.2)  THEN
C           inline accumulator
C         PIPE1 - LEN
C         PIPE2 - DIA
C         PIPE3 - DEN
C         PIPE4 - K
        L=PIPE1
        DIA=PIPE2
        AREA=0.25*PI*PIPE2**2
        IF(PIPE3.EQ.0.0)  PIPE3=DENS
        IF(PIPE4.EQ.0.0)  PIPE4=AVGK
```

```
          PCAP=PIPE3*L*AREA/PIPE4
         ELSEIF(SECIN.EQ.3)  THEN
C             tuned stub - suppresses omega = (PI/2)/(L*SQRT(PIND*PCAP))
C          PIPE1 - LEN
C          PIPE2 - DIA
C          PIPE3 - DEN
C          PIPE4 - K
          L=PIPE1
          DIA=PIPE2
          AREA=0.25*PI*DIA**2
          IF(PIPE3.EQ.0.0)  PIPE3=DENS
          IF(PIPE4.EQ.0.0)  PIPE4=AVGK
          PCAP=PIPE3*L*AREA/PIPE4
          PIND=L/(AREA*GRAV)
         ELSEIF(SECIN.EQ.4.OR.SECIN.EQ.5)  THEN
C             helmholtz resonator or parallel resonator
C               suppresses omega = 1/SQRT(PIND*PCAP)
C          PIPE1 - LEN
C          PIPE2 - DIA
C          PIPE3 - VOL
C          PIPE4 - DEN
C          PIPE5 - K
          L=PIPE1
          DIA=PIPE2
          AREA=PIPE3
          IF(PIPE4.EQ.0.0)  PIPE4=DENS
          IF(PIPE5.EQ.0.0)  PIPE5=AVGK
          PCAP=PIPE4*AREA/PIPE5
          PIND=L/(0.25*PI*DIA**2*GRAV)
         ELSEIF(SECIN.EQ.6)  THEN
C             pump
C          PIPE1 - LEN
C          PIPE2 - DIA
C          PIPE3 - DP/DM
C          PIPE4 - IND
C          PIPE5 - CAP
          L=PIPE1
          DIA=PIPE2
          AREA=PIPE3
          PCAP=PIPE4
          PIND=PIPE5
         ELSEIF(SECIN.EQ.7)  THEN
C             manifold
C          PIPE1 - VOLMF
C          PIPE2 - KMAN
          VOLMF=PIPE1
          KMAN=PIPE2
          CMAN=DENS*VOLMF/KMAN
          L=VOLMF
          DIA=CMAN
         ENDIF
        RETURN
        END
```

```
      SUBROUTINE SETPLT
C       Sets up the plot environment
      COMMON /WCAPAS/IFRST
      COMMON /NOCOL/MODE,MODET,NTROWS,NTCOLS,NPROWS,NPCOLS
      COMMON /ADMCOL/ADMBAC,ADMLIN
      INTEGER ADMBAC,ADMLIN
      CHARACTER*1 ANS
      DATA ITIM/0/
      IF(ITIM.EQ.0)  THEN
       ITIM=1
       ADMBAC=4
       ADMLIN=1
      ENDIF
      CALL QRMODE(MODET,NCOLT)
      CALL QVIDBD(IBOARD)
      IF(IBOARD.LT.1.OR.IBOARD.GT.3)  THEN
       WRITE(*,*)' Graphics board not installed!'
       RETURN
      ENDIF
      IF(IBOARD.EQ.1)  THEN
       MODE=6
       NPROWS=200
       NTROWS=25
      ENDIF
      IF(IBOARD.EQ.2)  THEN
       MODE=16
       NPROWS=350
       NTROWS=25
      ENDIF
      IF(IBOARD.EQ.3)  THEN
       MODE=18
       NPROWS=480
       NTROWS=25
      ENDIF
      IFRST=0
      NTCOLS=NCOLT
      NPCOLS=640
      IF(MODE.NE.6)  THEN
       WRITE(*,'(A\)')' Do you wish change colors of admittance? '
       READ(*,'(A)')ANS
       IF(ANS.EQ.'Y'.OR.ANS.EQ.'y')  THEN
        WRITE(*,*)' Enter no. of background color and no. of line color'
        WRITE(*,*)'   4,1 will give the default colors '
        WRITE(*,'(A\)')'    0,7 will give black & white '
        READ(*,*)ADMBAC,ADMLIN
       ENDIF
      ENDIF
      CALL QSMODE(MODE)
      RETURN
      END
      SUBROUTINE STSECT(J,ITYPE,POINT,LEN,DIA)
C       Computes plot coordinates for a straight section
      COMMON /PIPPXY/X,XH,XL,Y,YH,YL,XMIN,XMAX,YMIN,YMAX,SINA,COSA
```

```
      REAL LEN,POINT(8,200)
      INTEGER*2 ITYPE(200)
      J=J+1
      ITYPE(J)=1
      XH=X-0.5*SINA*DIA
      XL=X+0.5*SINA*DIA
      YH=Y+0.5*COSA*DIA
      YL=Y-0.5*COSA*DIA
      POINT(1,J)=XH
      POINT(2,J)=YH
      POINT(3,J)=XL
      POINT(4,J)=YL
      X=X+COSA*LEN
      XH=X-0.5*SINA*DIA
      XL=X+0.5*SINA*DIA
      Y=Y+SINA*LEN
      YH=Y+0.5*COSA*DIA
      YL=Y-0.5*COSA*DIA
      POINT(5,J)=XH
      POINT(6,J)=YH
      POINT(7,J)=XL
      POINT(8,J)=YL
      XMIN=AMIN1(X,XL,XH,XMIN)
      XMAX=AMAX1(X,XL,XH,XMAX)
      YMIN=AMIN1(Y,YL,YH,YMIN)
      YMAX=AMAX1(Y,YL,YH,YMAX)
      RETURN
      END
      SUBROUTINE TANKNO(MTANK,VOL,LFLOW,KTANK,DENS,A,CTANK,IUNIT)
C       Reads tank parameters
      REAL VOL(25),LFLOW(25),KTANK(25),DENS(25),A(25),CTANK(25)
      DATA GRAV/32.2/
      READ(IUNIT,*)MTANK
      IF(MTANK.GT.25)  THEN
       WRITE(*,*)' Number of tanks must be less than 25'
       STOP
      ENDIF
      IF(MTANK.LE.0)  MTANK=1
      DO 21 I=1,MTANK
      READ(IUNIT,*)VOL(I),LFLOW(I),KTANK(I),DENS(I)
      A(I)=SQRT(GRAV*KTANK(I)/DENS(I))
      CTANK(I)=DENS(I)*VOL(I)/KTANK(I)
   21 CONTINUE
      RETURN
      END
      SUBROUTINE TSSECT(J,ITYPE,POINT,LEN,DIA)
C       Computes plot coordinates for a tuned stub
      COMMON /PIPPXY/X,XH,XL,Y,YH,YL,XMIN,XMAX,YMIN,YMAX,SINA,COSA
      REAL LEN,POINT(8,200)
      INTEGER*2 ITYPE(200)
      J=J+1
      ITYPE(J)=1
      DIAM=SQRT((XH-XL)**2+(YH-YL)**2)
```

```
      XH=X-SINA*(LEN+0.5*DIAM)
      YH=Y+COSA*(LEN+0.5*DIAM)
      POINT(1,J)=XH
      POINT(2,J)=YH
      POINT(3,J)=XL
      POINT(4,J)=YL
      X=X+COSA*DIA
      XH=X-SINA*(LEN+0.5*DIAM)
      XL=XL+COSA*DIA
      Y=Y+SINA*DIA
      YH=Y+COSA*(LEN+0.5*DIAM)
      YL=YL+SINA*DIA
      POINT(5,J)=XH
      POINT(6,J)=YH
      POINT(7,J)=XL
      POINT(8,J)=YL
      XMIN=AMIN1(X,XL,XH,XMIN)
      XMAX=AMAX1(X,XL,XH,XMAX)
      YMIN=AMIN1(Y,YL,YH,YMIN)
      YMAX=AMAX1(Y,YL,YH,YMAX)
      RETURN
      END
      SUBROUTINE UPPERW(X0,Y0,X1,Y1,IFULOX,ITANK,IENG)
C       Sets up upper plotting window
      COMMON /NOCOL/MODE,MODET,NTROWS,NTCOLS,NPROWS,NPCOLS
      COMMON /ADMCOL/ADMBAC,ADMLIN
      INTEGER ADMBAC,ADMLIN
      CHARACTER*36 FULOX
    1 FORMAT('FUEL Piping - Tank # ',I2,'  Engine # ',I2)
    2 FORMAT(' LOX Piping - Tank # ',I2,'  Engine # ',I2)
      XMIN=X0
      XMAX=X1
      YMIN=Y0
      YMAX=Y1
      JCOL1=100
      JCOL2=550
      IF(MODE.EQ.6)  THEN
       JROW1=100
       JROW2=179
      ELSEIF(MODE.EQ.16)  THEN
       JROW1=214
       JROW2=309
      ELSEIF(MODE.EQ.18)  THEN
       JROW1=244
       JROW2=449
      ENDIF
      XORG=XMIN
      YORG=YMIN
      YOVERX=1.0
      IOPT=1
      ASPECT=1.35
      YMAX0=YMAX
      CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,
```

```
     *           XORG,YORG,IOPT,YOVERX,ASPECT)
      IF(IOPT.GE.0)  GO TO 21
      IOPT=1
      CHANGE=(YMAX-YMIN)/(YMAX0-YMIN)
      JCOL2=JCOL1+0.98*CHANGE*(JCOL2-JCOL1)
      YMAX=YMAX0
      CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,
     *           XORG,YORG,IOPT,YOVERX,ASPECT)
   21 CONTINUE
      IF(MODE.NE.6)  THEN
       CALL QPREG(0,ADMBAC)
      ENDIF
      CALL QSETUP(0,ADMLIN,-2,ADMLIN)
      IF(IFULOX.EQ.1)  THEN
C         fuel line
       WRITE(FULOX,1)ITANK,IENG
      ELSE
C         lox line
       WRITE(FULOX,2)ITANK,IENG
      ENDIF
      IF(MODE.NE.18)  THEN
       CALL QPTXT(36,FULOX,7,24,23)
      ELSE
       CALL QPTXT(36,FULOX,7,24,27)
      ENDIF
      RETURN
      END
      SUBROUTINE WINDOW(MODE,XSCALE,XST,XFIN,YST,YFIN,ZST,ZFIN)
C       Sets up window for surface plot
      CALL Q3DWIN(XST,XFIN,YST,YFIN,ZST,ZFIN,XMIN,XMAX,YMIN,YMAX)
      JCOL1=100
      JCOL2=450
      JROW1=40
      JROW2=169
      IF(MODE.EQ.16)  JROW2=319
      IF(MODE.EQ.18)  JROW2=409
      XORG=XMIN
      YORG=YMIN
      YOVERX=1.0
      IOPT=0
      ASPECT=XSCALE
      CALL QPLOT(JCOL1,JCOL2,JROW1,JROW2,XMIN,XMAX,YMIN,YMAX,
     *           XORG,YORG,IOPT,YOVERX,ASPECT)
      RETURN
      END
      FUNCTION XFUN(T)
C       Parametric function for plotting of bends
      COMMON /ARCCON/XC,YC,RAD,ANG,ANGLE
      XFUN=XC+RAD*SIN(T)
      RETURN
      END
      FUNCTION YFUN(T)
C       Parametric function for plotting of bends
```

```
      COMMON /ARCCON/XC,YC,RAD,ANG,ANGLE
      YFUN=YC-RAD*COS(T)
      RETURN
      END
      SUBROUTINE ZREAD(NAME,VALUE)
C       Reads input for input modification
      CHARACTER*1 NAME(8)
      CHARACTER*1 CARD(80),PLUS,MINUS,PERIOD,LE,E,NUMBER(10)
      CHARACTER*1 LEND(3),CEND(3),POUND,QUEST,BLK,COMMA
      CHARACTER*1 LTIT(5),CTIT(5)
      CHARACTER*80 DCARD
      EQUIVALENCE (CARD(1),DCARD)
      DATA PLUS/'+'/,MINUS/'-'/,PERIOD/'.'/,LE/'e'/,E/'E'/,BLK/' '/
      DATA NUMBER/'0','1','2','3','4','5','6','7','8','9'/,COMMA/','/
      DATA LEND/'e','n','d'/,CEND/'E','N','D'/,POUND/'#'/,QUEST/'?'/
      DATA LTIT/'t','i','t','l','e'/,CTIT/'T','I','T','L','E'/
    1 FORMAT(A)
      DO 21 I=1,8
       NAME(I)=BLK
   21 CONTINUE
      READ(*,1)DCARD
      IF(CARD(1).EQ.POUND)  THEN
       NAME(1)=POUND
       RETURN
      ENDIF
      IF(CARD(1).EQ.QUEST)  THEN
       NAME(1)=QUEST
       RETURN
      ENDIF
      DO 22 I=1,3
       IF(CARD(I).NE.LEND(I).AND.CARD(I).NE.CEND(I))  GO TO 23
       NAME(I)=CEND(I)
   22 CONTINUE
      RETURN
   23 CONTINUE
      DO 24 I=1,5
       IF(CARD(I).NE.LTIT(I).AND.CARD(I).NE.CTIT(I))  GO TO 25
       NAME(I)=CTIT(I)
   24 CONTINUE
      RETURN
   25 CONTINUE
      DO 26 I=1,8
       II=I
       IF(CARD(I).EQ.BLK.OR.CARD(I).EQ.COMMA)  GO TO 27
       NAME(I)=CARD(I)
   26 CONTINUE
   27 CONTINUE
      DO 28 I=II,80
       ID=I
       IF(CARD(I).NE.BLK.AND.CARD(I).NE.COMMA)  GO TO 29
   28 CONTINUE
      VALUE=0.0
      WRITE(*,*)'   No value given, ZERO assumed'
```

```
      RETURN
   29 CONTINUE
      SIGN=1.0
      IF(CARD(ID).EQ.MINUS)  THEN
       SIGN=-1.0
       ID=ID+1
      ELSEIF(CARD(ID).EQ.PLUS)  THEN
       ID=ID+1
      ENDIF
      WHOLE=0.0
      DO 32 I=ID,80
       II=I
       IF(CARD(I).EQ.PERIOD)  GO TO 31
       IF(CARD(I).EQ.PLUS)  GO TO 38
       IF(CARD(I).EQ.MINUS)  GO TO 38
       IF(CARD(I).EQ.E.OR.CARD(I).EQ.LE)  GO TO 37
       DO 30 J=1,10
        JJ=J-1
        IF(CARD(I).EQ.NUMBER(J))  GO TO 31
   30 CONTINUE
       VALUE=SIGN*WHOLE
       IF(CARD(I).EQ.BLK)  RETURN
       WRITE(*,*)'  Input error, value set to ZERO'
       VALUE=0.0
       RETURN
   31 CONTINUE
       WHOLE=WHOLE*10.0+JJ
   32 CONTINUE
      VALUE=SIGN*WHOLE
      RETURN
   33 CONTINUE
      ID=II+1
      FRACT=0.0
      ICOUNT=0
      DO 36 I=ID,80
       ICOUNT=ICOUNT+1
       II=I
       IF(CARD(I).EQ.PERIOD)  THEN
        WRITE(*,*)'  Input error, value set to ZERO'
        VALUE=0.0
        RETURN
       ENDIF
       IF(CARD(I).EQ.PLUS)  GO TO 38
       IF(CARD(I).EQ.MINUS)  GO TO 38
       IF(CARD(I).EQ.E.OR.CARD(I).EQ.LE)  GO TO 37
       DO 34 J=1,10
        JJ=J-1
        IF(CARD(I).EQ.NUMBER(J))  GO TO 35
    34 CONTINUE
       VALUE=SIGN*(WHOLE+FRACT)
       IF(CARD(I).EQ.BLK)  RETURN
       WRITE(*,*)'  Input error, value set to ZERO'
       VALUE=0.0
```

```
      RETURN
35 CONTINUE
      FRACT=FRACT+JJ/10.0**ICOUNT
36 CONTINUE
   VALUE=SIGN*(WHOLE+FRACT)
   RETURN
37 CONTINUE
   II=II+1
38 CONTINUE
   VALUE=SIGN*(WHOLE+FRACT)
   SIGN=1.0
   IF(CARD(II).EQ.MINUS)  THEN
    SIGN=-1.0
    II=II+1
   ELSEIF(CARD(II).EQ.PLUS)  THEN
    II=II+1
   ENDIF
   WHOLE=0.0
   DO 41 I=II,80
    DO 39 J=1,10
     JJ=J-1
     IF(CARD(I).EQ.NUMBER(J))  GO TO 40
39 CONTINUE
    VALUE=VALUE*10.0**(SIGN*WHOLE)
    IF(CARD(I).EQ.BLK)  RETURN
    WRITE(*,*)'  Input error, value set to ZERO'
    VALUE=0.0
    RETURN
40 CONTINUE
    WHOLE=WHOLE*10.0+JJ
41 CONTINUE
   VALUE=VALUE*10.0**(SIGN*WHOLE)
   RETURN
   END
```