Generalized Fluid System Simulation Program (GFSSP)
Supplementary Documentation for Version 605

Alok Majumdar, Andre Leclair, and Ric Moore
ER43/Thermal Analysis Branch
NASA/Marshall Space Flight Center
Huntsville, Alabama – 35812

June 2014

Summary

Since the release of GFSSP Version 604 and the User's Manual in December 2013[1], there were several enhancements to the code which led to the development of Version 605. It was decided that Version 605 will be the final release of Version 6; all future enhancements will be part of Version 7 and beyond. Therefore, this supplementary document to the GFSSP User's Manual was prepared to describe the additional features of Version 605 which include: 1. Saturation Property Calculation for User Specified Fluid Property Table Look-up, 2. Provision of Fluid Property Call from User Subroutine, 3. Improved Efficiency, 4. Improvements in the Graphical User Interface, and 5. Five Additional Example Problems to demonstrate the code's application for *i*) Rapid Valve Opening to cause Fluid Transient, *ii*) Saturation Property Calculation for User Specified Fluid Property Table Look-up, *iii*) No Vent Chill and Fill of a Cryogenic Propellant Tank, *iv*) Self-Pressurization of a Cryogenic Propellant Tank with Multi-Layer Insulation, and *v*) Ballistic Calculation of a Solid Propellant Rocket Motor. In addition, there are major revisions to the source code to improve the readability and reduction in lines of code. This improvement was made possible by the introduction of universal property call subroutines.

Topics

1.0	**New Features**
1.1	Saturation Property Table for User Fluid
1.2	Call of Property Routine from User Subroutine
1.3	Improved Efficiency
2.0	**VTASC Upgrades**
3.0	**New Example Problems**
3.1	Chilldown of Cryogenic Transfer Line (Revised Example 14)
3.2	Fluid Transients in Pipes Due to Sudden Opening of Valve (Example 26)
3.3	Boiling Water Reactor (Example 27)
3.4	No-Vent Tank Chill & Fill Model (Example 28)
3.5	Self-Pressurization of a Cryogenic Propellant Tank Due to Boil-off (Example 29)
3.6	Modeling Solid Propellant Ballistic with GFSSP (Example 30)
4.0	**References**
	Appendix
A.1	Instructions to convert REFPROP data to GFSSP Property Table
A.2	User Subroutine for Example 26
A.3	User Subroutine for Example 28
A.4	User Subroutine for Example 29
A.5	User Subroutine for Example 30

Acknowledgement

The authors would like to acknowledge the support of MSFC's NIRPS (National Institute of Rocket Propulsion System) and SLS, KSC's LSP (Launch Service Program) and GRC's CPST (Cryogenic Propellant Storage and Transfer) Projects. The encouragement and support of Tom Brown (NIRPS), Wendy Cruit/Tony Lavoie (SLS-Stages), Tom Byrd (LES), Danny Garcia (Tech Transfer) and Melissa Van Dyke of MSFC, Paul Schallhorn of KSC, and Jeff Moder of GRC are gratefully acknowledged. The authors would also like to acknowledge the contribution of Alak Bandyopadhyay of Alabama A&M University for the development of the Example Problem on Fluid Transient (Example 26), Johnny Maroney of Jacobs/ESSA for his contribution to the Example Problem on No-Vent Tank Fill (Example 28), and Juan Valenzuela, John Bailey and Ali Hedayat of MSFC for the development of the Example Problem on Self Pressurization of Cryogenic Propellant Tank (Example 29).

1. New Features

This section describes the new features of Version 605. The primary purpose of Version 605 is to extend the capability of Look-Up Tables of thermodynamic properties. GFSSP is integrated with thermodynamic programs that provide fluid properties for sub-cooled, superheated, and saturation states. For fluids that are not included in the thermodynamic property program, look-up property tables can be provided. The look-up property tables of Versions 604 and earlier can only handle the sub-cooled and superheated states. The primary purpose of Version 605 was to extend the look-up tables to handle saturated states. This involves a) generation of a property table using REFPROP, a thermodynamic property program [2] that is widely used, and b) modifications of the Fortran source code to read in an additional property table containing saturation data for both saturated liquid and saturated vapor states. Section 1.1 describes this feature. Version 605 also allows users to make calls for fluid properties from User Subroutines. Four universal property call routines have been developed and are discussed in Section 1.2. There has been improvement in computational efficiency by modifying the frequently-called indexing subroutine INDEXI. This is further explained in Section 1.3

1.1 Saturation Property Calculation for User Specified Fluid

GFSSP has the capability to define a User Fluid with property interpolation tables. These tables provide seven fluid properties as functions of pressure and temperature. Beginning with v605, saturated fluid properties and phase change may be modeled with a User Fluid, provided that an eighth table of saturation properties is also supplied. A maximum of three User Fluids is permitted in a model.

Figure 1.1.1 shows the VTASC Fluid Options dialog as set up to define water as a User Fluid. The seven property filenames and fluid molecular weight must be provided. The eighth file providing saturation properties is optional and can be added by checking the Phase Change box.

The seven properties that must be provided are Thermal Conductivity, Density, Absolute Viscosity, Specific Heat Ratio, Enthalpy, Entropy, and Specific Heat. The required units depend on whether the user has developed a model in English or SI units and are shown in Table 1.1.1.

The format of the seven property tables is:

NP, NT
T(1), T(2), T(3),…, T(NT)
P(1), PROP(1,1), PROP(1,2), PROP(1,3),…, PROP(1,NT)
P(2), PROP(2,1), PROP(2,2), PROP(2,3),…, PROP(2,NT)
…
P(NP), PROP(NP,1), PROP(NP,2), PROP(NP,3),…, PROP(NP,NT)

NP is the number of pressures, and NT is the number of temperatures. There can be a maximum of 301 pressures and 301 temperatures in the property file. The pressures and temperatures must be the same in all seven files. Figure 1.1.2 shows a portion of the density file for water.

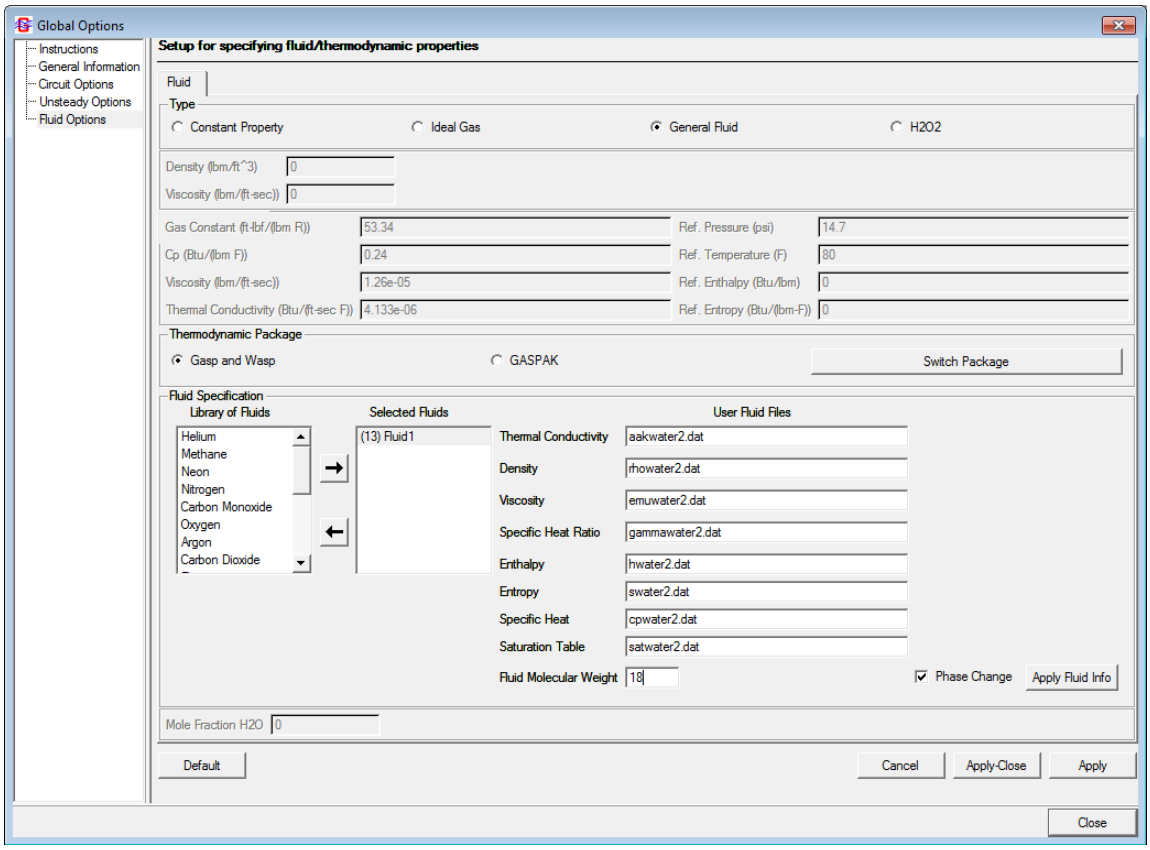
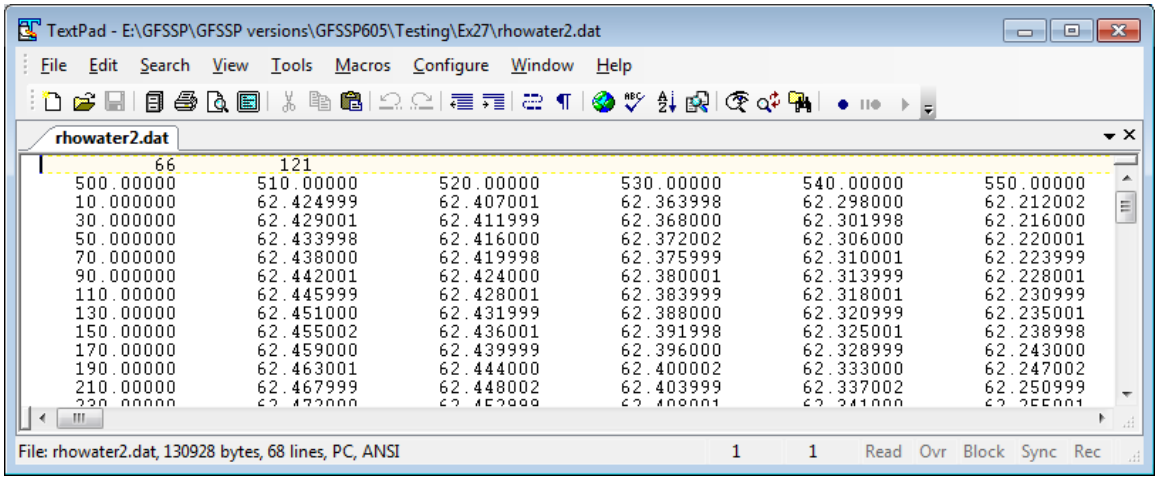


Figure 1.1.1 Defining a User Fluid in the VTASC Fluid Options Dialog

Table 1.1.1 User Fluid Properties and Units

Property	English Units	SI Units
Pressure (P)	psia	kPa
Temperature (T)	°R	K
Conductivity (k)	BTU/ft-s-R	W/m-K
Density (ρ)	lb/ft^3	kg/m^3
Viscosity (μ)	lb/ft-s	$N\text{-}s/m^2$
Specific Heat Ratio (γ)	Dimensionless	Dimensionless
Enthalpy (H)	BTU/lb	kJ/kg
Entropy (S)	BTU/lb-R	kJ/kg-K
Specific Heat (Cp)	BTU/lb-R	kJ/kg-K



TextPad - E:\GFSSP\GFSSP versions\GFSSP605\Testing\Ex27\rhowater2.dat

File Edit Search View Tools Macros Configure Window Help

rhowater2.dat

66	121				
500.00000	510.00000	520.00000	530.00000	540.00000	550.00000
10.000000	62.424999	62.407001	62.363998	62.298000	62.212002
30.000000	62.429001	62.411999	62.368000	62.301998	62.216000
50.000000	62.433998	62.416000	62.372002	62.306000	62.220001
70.000000	62.438000	62.419998	62.375999	62.310001	62.223999
90.000000	62.442001	62.424000	62.380001	62.313999	62.228001
110.00000	62.445999	62.428001	62.383999	62.318001	62.230999
130.00000	62.451000	62.431999	62.388000	62.320999	62.235001
150.00000	62.455002	62.436001	62.391998	62.325001	62.238998
170.00000	62.459000	62.439999	62.396000	62.328999	62.243000
190.00000	62.463001	62.444000	62.400002	62.333000	62.247002
210.00000	62.467999	62.448002	62.403999	62.337002	62.250999

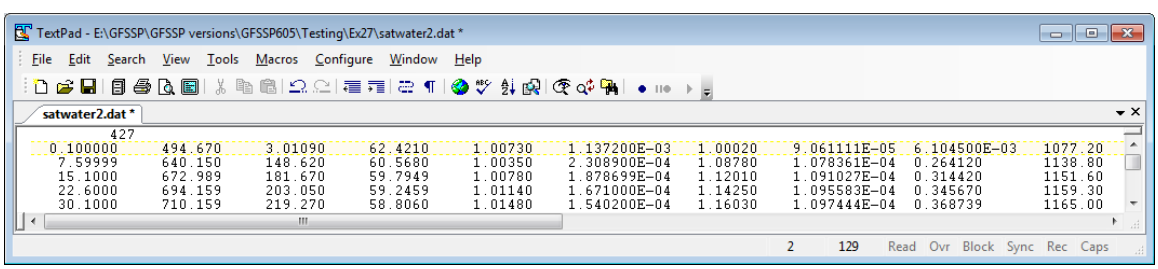
File: rhowater2.dat, 130928 bytes, 68 lines, PC, ANSI 1 1 Read Ovr Block Sync Rec

Figure 1.1.2 User Fluid Property File for Water Density

The format of the optional eighth property file for saturation properties is:

NPSAT
PSAT(1), TSAT, (H, ρ, Cp, μ, γ, k, S)LIQ, (H, ρ, Cp, μ, γ, K, S)VAP
PSAT(2), TSAT, (H, ρ, Cp, μ, γ, k, S)LIQ, (H, ρ, Cp, μ, γ, K, S)VAP
…
PSAT(NPSAT), TSAT, (H, ρ, Cp, μ, γ, k, S)LIQ, (H, ρ, Cp, μ, γ, K, S)VAP

NPSAT is the number of saturation pressures; the maximum is 500. The saturation pressures do not need to be the same as the pressures in the seven property tables. Each line gives a saturation pressure, the corresponding saturation temperature, and the seven properties of the saturated liquid and saturated vapor. Figure 1.1.3 shows a portion of the saturated property file for water.



TextPad - E:\GFSSP\GFSSP versions\GFSSP605\Testing\Ex27\satwater2.dat *

File Edit Search View Tools Macros Configure Window Help

satwater2.dat *

427									
0.100000	494.670	3.01090	62.4210	1.00730	1.137200E-03	1.00020	9.061111E-05	6.104500E-03	1077.20
7.59999	640.150	148.620	60.5680	1.00350	2.308900E-04	1.08780	1.078361E-04	0.264120	1138.80
15.1000	672.989	181.670	59.7949	1.00780	1.878699E-04	1.12010	1.091027E-04	0.314420	1151.60
22.6000	694.159	203.050	59.2459	1.01140	1.671000E-04	1.14250	1.095583E-04	0.345670	1159.30
30.1000	710.159	219.270	58.8060	1.01480	1.540200E-04	1.16030	1.097444E-04	0.368739	1165.00

2 129 Read Ovr Block Sync Rec Caps

Figure 1.1.3 User Fluid Saturated Property File

A common source of fluid properties is the REFPROP program. Utility programs with instructions (Appendix A.1) for converting output from REFPROP into GFSSP's User Fluid format are provided in the User Fluid Utilities sub-folder of the GFSSP installation folder.

General guidelines for developing User Fluid tables:

- Interpolation is linear with respect to pressure and temperature. Many pressure points are recommended for compressible fluids, to ensure accuracy of density interpolation.
- User Fluids cannot be used with the Second Law energy equation, so entropy is a print-out variable only. If entropy is unknown, the user may place dummy values in the table.
- If viscosity, specific heat, and thermal conductivity are not known as functions of pressure and/or temperature, the user may use the same constant value at all pressure and/or temperature points in the table.
- If enthalpy is not known, the user may construct enthalpy tables by integrating the specific heat over temperature.

1.2 Provision of Fluid Property Call from User Subroutine

GFSSP calculates fluid properties in every node as functions of (1) Pressure and Temperature, (2) Pressure and Enthalpy, or (3) Pressure and Entropy. In v605, most of the property package calls were standardized by placing them in one of three universal property call subroutines: PROPS_PT, PROPS_PH, and PROPS_PS. An additional subroutine called PROPS_PSATX is available to users who desire saturation properties at a given pressure. These subroutines call the GASP/WASP programs, RP-1 interpolation tables, and User Fluid interpolation tables. At this time, they do not call the GASPAK property program.

An advantage of the universal property call subroutines is that it provides a utility for writing User Subroutines. For example, when calculating a convection coefficient using a film temperature (mean of the fluid and solid node temperatures), the user will want properties at a different temperature than the current node temperature. These can be provided with a call to PROPS_PT. Two-phase convection coefficient correlations may require properties of the saturated liquid and vapor phases, which can be provided with calls to PROPS_PH, PROPS_PS, or PROPS_PSATX.

Subroutine PROPS_PT returns single-phase fluid properties as a function of Pressure and Temperature. It does not return saturation properties. The calling statement for PROPS_PT is:

```
      CALL PROPS_PT(I_NFLUID, Z_P, Z_T, Z_RHO, Z_H, Z_CP, Z_CV,
     +              Z_S, Z_GAMMA, Z_MU, Z_K, I_KR, Z_XV)
```

Input:
I_NFLUID: Integer ID number of the fluid
Z_P: Pressure
Z_T: Temperature

Output:
Z_RHO: Density
Z_H: Enthalpy
Z_CP: Specific heat at constant pressure
Z_CV: Specific heat at constant volume
Z_S: Entropy
Z_GAMMA: Ratio of specific heats
Z_MU: Viscosity
Z_K: Thermal conductivity
I_KR: Integer code for the fluid phase (0, unknown; 1, saturated; 2, liquid; 3, gas)
Z_XV: Quality (vapor mass fraction)

The units of the input and output properties are the same as GFSSP's internal units, and are shown in Table 1.2.1. Table 1.2.2 gives the fluid ID numbers that are recognized by the PROPS subroutines. An incorrect ID number will generate an error message and stop the run.

Table 1.2.1 Fluid Properties and Units

Property	English Units
Pressure (P)	Psf
Temperature (T)	°R
Conductivity (k)	BTU/ft-s-R
Density (ρ)	lb/ft^3
Viscosity (μ)	lb/ft-s
Specific Heat Ratio (γ)	Dimensionless
Enthalpy (H)	BTU/lb
Entropy (S)	BTU/lb-R
Specific Heat (Cp)	BTU/lb-R
Specific Heat (Cv)	BTU/lb-R

Subroutine PROPS_PH returns fluid properties as a function of Pressure and Enthalpy. The calling statement for PROPS_PH is:

```
 CALL PROPS_PH(I_NFLUID, Z_P, Z_T, Z_RHO, Z_H, Z_CP, Z_CV,
+        Z_S, Z_GAMMA, Z_MU, Z_K, I_KR, Z_XV,
+        Z_RHOL, Z_HL, Z_CPL, Z_CVL, Z_SL, Z_GAMMAL, Z_MUL, Z_KL,
+        Z_RHOV, Z_HV, Z_CPV, Z_CVV, Z_SV, Z_GAMMAV, Z_MUV, Z_KV)
```

Table 1.2.2 Fluid ID Numbers and Critical Pressures

ID Number	Fluid	Pcrit (psia)
1	GASP He	33.0
2	GASP CH4	671.1
3	GASP Ne	384.9
4	GASP N2	495.6
5	GASP CO	507.4
6	GASP O2	737.2
7	GASP Ar	705.6
8	GASP CO2	1070.9
9	GASP F2	756.4
10	GASP H2	187.5
11	WASP H2O	3204.0
12	RP-1 Tables	
37	User Fluid 1 Tables	
38	User Fluid 2 Tables	
39	User Fluid 3 Tables	

Input:
I_NFLUID: Integer ID number of the fluid
Z_P: Pressure
Z_H: Enthalpy

The output is similar to the output of PROPS_PT, except that there are additional liquid (suffix L) and vapor (suffix V) property values returned when the fluid is saturated. If the input Enthalpy falls under the saturation dome at the given Pressure, variable I_KR will be returned as 1 (saturated), and the properties will be those of a homogeneous two-phase mixture with a quality of Z_XV. If the fluid is single-phase at the given Pressure and Enthalpy, variable I_KR will be returned as 2 (liquid) or 3 (gas), and the saturated liquid and vapor property values will be zero.

Subroutine PROPS_PS returns fluid properties as a function of Pressure and Entropy. The calling statement for PROPS_PS is:

```
      CALL PROPS_PS(I_NFLUID, Z_P, Z_T, Z_RHO, Z_H, Z_CP, Z_CV,
     +        Z_S, Z_GAMMA, Z_MU, Z_K, I_KR, Z_XV,
     +        Z_RHOL, Z_HL, Z_CPL, Z_CVL, Z_SL, Z_GAMMAL, Z_MUL, Z_KL,
     +        Z_RHOV, Z_HV, Z_CPV, Z_CVV, Z_SV, Z_GAMMAV, Z_MUV, Z_KV)
```

Input:
I_NFLUID: Integer ID number of the fluid
Z_P: Pressure
Z_S: Entropy

The output and saturation functionality is similar to that of PROPS_PH. The RP-1 and User Fluid interpolation table subroutines do not work as functions of entropy, so calls to PROPS_PS for these fluids will generate an error message and stop the run.

Subroutine PROPS_PSATX returns fluid properties at a given saturation Pressure and Quality (vapor mass fraction). The calling statement for PROPS_PSATX is:

```
      CALL PROPS_PSATX(I_NFLUID, Z_P, Z_T, Z_RHO, Z_H, Z_CP, Z_CV,
     +        Z_S, Z_GAMMA, Z_MU, Z_K, I_KR, Z_XV,
     +        Z_RHOL, Z_HL, Z_CPL, Z_CVL, Z_SL, Z_GAMMAL, Z_MUL, Z_KL,
     +        Z_RHOV, Z_HV, Z_CPV, Z_CVV, Z_SV, Z_GAMMAV, Z_MUV, Z_KV)
```

Input:
I_NFLUID: Integer ID number of the fluid
Z_P: Saturation Pressure
Z_XV: Quality (vapor mass fraction)

The output is similar to saturated output from PROPS_PH and PROPS_PS. If the user only desires the liquid (suffix L) and vapor (suffix V) properties, an input Quality of zero may be used, and the Quality-weighted properties ignored. At this time, this subroutine only works for GASP/WASP fluids; calls with RP-1 or User Fluids will generate an error message and stop the run. Providing a saturation pressure greater than the critical pressure will also generate an error message and stop the run. Critical pressures for the GASP/WASP fluids are listed in Table 1.2.2.

<u>Cleaning up of the Source Code</u>

After the three property call subroutines had been written and tested in a driver program, they were integrated into the GFSSP source code. There are seven GFSSP subroutines with property calls: INIT, DENSITY, BOUND, MIXTEMP, EQNSHM, MIXPROP, and CONVHC. With the use of these three subroutines, these seven existing subroutines have been rewritten reduce the number of lines of code and make it easier to read and debug the code. The total number of lines of fluid property code in these seven subroutines was reduced from 1654 to 724 (about 56%).

1.3 Improved Efficiency

GFSSP frequently requires the index of an array to access properties of a node or branch. This is often called “finding the pointer to a node”. The utility subroutine to accomplish this is INDEXI.

```
CALL INDEXI(NUMBER,NODE,NNODES,IPN)
```

Variable NUMBER is the node number, as seen on the VTASC canvas and in the input file. NODE is the array of these node numbers, of size NNODES. The subroutine returns the integer variable IPN (“Integer Pointer to Node”), which is the array index

needed to access node properties, such as fluid temperature TF(IPN) or density RHO(IPN).

The pointer to a branch, IPB, can be found by modifying the call statement to use the array of branch numbers, IBRANCH, of size NBR.

```
CALL INDEXI(NUMBER,IBRANCH,NBR,IPB)
```

In v605 subroutine INDEXI has been modified to improve the efficiency of the code. The search loop over array NODE now features an early exit as soon as the correct node is found. This can decrease run time in transient models with large numbers of nodes. The subroutine also checks for incorrect input, as may happen when INDEXI is called from a user subroutine. If the node cannot be found, an error message is printed and the run is stopped.

```
      SUBROUTINE INDEXI(NUMBER,NODE,NNODES,IPN)
      DIMENSION NODE(1)
      DO II = 1,NNODES
         IF (NUMBER .EQ. NODE(II)) THEN
            IPN = II
            RETURN
         END IF
      END DO
      PRINT *, 'ERROR.  Can''t find ID# ', NUMBER
      STOP
      END
```

Throughout the older portions of the GFSSP source code, more than 100 inefficient node and branch search loops that were hard-coded have also been replaced by calls to the new INDEXI.

Although the modifications to subroutine INDEXI have had little effect in small GFSSP models, there is a noticeable decrease in running time for transient models with large number of nodes and branches. Example 12 (Helium Pressurization of LOx and RP-1 Propellant Tanks) has 65 nodes and 64 branches. On a certain desktop PC, a ten-second simulation takes 1004 seconds of computational time with Version 604. On the same machine with Version 605, the run requires 852 seconds, an improvement of 15%.

2.0 VTASC Upgrade

Multiple upgrades to VTASC, GFSSP's graphical user interface, have been incorporated for Version 605. These include changes to the User Options menus, the ability to place descriptive text on the canvas, a change in the way branch connecting lines are drawn, and additional checks for proper input.

The User Options menus, accessed from VTASC's Edit menu, have been modified to allow for new features, improve usability, and prevent user error. There are new tool tips throughout the menus, especially on the Solution Control and Fluid Options pages. The Circuit Options page now includes a checkbox for DFLI (Differential Formulation for Longitudinal Inertia). When checked (the default setting), GFSSP calculates the inertia term of the momentum equation of a branch ij as:

$$\dot{m}_{ij} d\left(\frac{\dot{m}}{\rho A}\right)$$

When unchecked, the inertia term is calculated as:

$$\dot{m}_{ij}(u_{upstream} - u_{ij})$$

The Unsteady Options page has been modified to allow the user to select both a Pressure Regulator and a Flow Regulator in the same model. The only restriction is that there may be only one Option 1 (Iterative Method) Regulator of any type within a model; the new VTASC includes checks to prevent the user from selecting more than one.

The Fluid Options page includes a new Phase Change checkbox for User Defined Fluids. When checked, the user supplies an eighth fluid property file with saturated fluid properties. New tool tips explain the format and units of the User Fluid property files.

The text description of a node or branch can now be placed on the canvas by selecting the Show Description checkbox on the node or branch property dialog. By default, node descriptions are placed below a node; branch descriptions, above. Right-clicking a node or branch brings up a menu which includes the option of relocating the description. Another change to the canvas is that branch connecting lines are now drawn so that the arrow always points to the center of the node. When the user adjusts the location of nodes and branches, it is not necessary to reconnect a branch to a different node handle.

Finally, VTASC Version 605 includes new checks to reduce user error. The Solution Control page verifies that values for the relaxation parameters are reasonable, and the Second Law formulation of the energy equation is disabled when a model includes a User

Fluid. The Mixture Enthalpy 2 option is disabled when the GASPAK fluid property package is selected, and the Mixture Enthalpy 1 option is disabled when Pressurization Advanced Option is selected. VTASC prevents the user from changing the ID number of a node or branch being used by an Advanced Option such as a Pressure or Flow Regulator.

3.0 New Example Problems

In this section several new example problems are described. These additional example problems are part of the Version 605 installation. The chilldown of a cryogenic pipeline (Example 14) from previous versions of GFSSP has been revised. The revised Example 14 (Section 3.1) describes the modeling of a National Bureau of Standards (NBS) Test Setup for experiments on the chilldown of a cryogenic transfer line. Five additional example problems (Examples 26 thru 30) are added in Version 605. Example 26 (Section 3.2) describes a fluid transient problem of sudden valve opening. The predictions of pressure history have been compared with test data. Example 27 (Section 3.3) illustrates the new feature of Version 605. It describes a boiling water reactor problem where GFSSP reads a saturation property table generated by REFPROP [2]. Example 28 (Section 3.4) describes a no-vent fill model of a cryogenic propellant tank. The prediction of propellant loss has been validated by comparing with test data. Section 3.5 describes the modeling of self-pressurization of a cryogenic propellant tank due to boil-off (Example 29). GFSSP's prediction of pressure history has been validated by comparing with test data. Example 30 (Section 3.6) illustrates the use of GFSSP in predicting ballistics of solid rocket propellant motor. User subroutines were needed to run several example problems. The user subroutine code is provided in the Appendix.

3.1 Chilldown of a Cryogenic Pipeline (Revised Example 14)

Problem Considered:
For this example, the chilldown of cryogenic pipeline to validate GFSSP's transient conjugate heat transfer capability has been selected. In the 1960s the National Bureau of Standards (NBS) conducted a series of chilldown experiments on a cryogenic transfer line [3]. The test setup (Figure 3.1.1) is a vacuum-jacketed 200 foot long copper pipe of 5/8 inch inner diameter. A pressurized 80 gallon dewar feeds liquid hydrogen into the pipe that is initially at ambient temperature. The wall temperature is measured at four thermocouple stations at distances of 20, 80, 141, and 198 feet from the inlet.
When the fluid touches the relatively warm pipe walls, heat transfer causes the liquid cryogen to boil and the pipe wall temperature to decrease. Eventually the pipe chills down to the liquid temperature, and the liquid front gradually travels further down the pipeline. At the outlet of the pipeline, vapor exits to the atmosphere.

The NBS experiments were conducted with liquid hydrogen (LH2) and liquid nitrogen (LN2) at various driving pressures with saturated and sub-cooled fluid. This example problem models one of the tests with an inlet boundary of saturated LH2 at 74.97 psia and -411 °F.

GFSSP Model
Figure 3.1.2 shows the GFSSP model of the chilldown experiment. The pipeline has been discretized into 30 pipe branches, each 80 inches long. The fluid nodes are connected to 31 solid nodes representing the mass of the copper pipe. The fluid and solid nodes are initially at the ambient temperature of 44 °F. Boundary node 1 represents the storage dewar at 74.97 psia and -411 °F. Boundary node 33 is the ambient atmosphere in

Boulder, Colorado at 12.05 psia. Restriction 12 represents the inlet valve, which opens during the first 0.05 seconds of the simulation. Restriction 3233 represents the minor loss of the pipeline exit, with a K factor of 1.0.

The solid nodes are connected to the fluid nodes by Fluid-to-Solid Conductors, which model convection from the fluid to the pipe wall. The built-in Miropolskii correlation is used to calculate the convection coefficient for the two-phase flow. Because the pipe is vacuum-jacketed, heat transfer between the pipe walls and the ambient is assumed negligible.

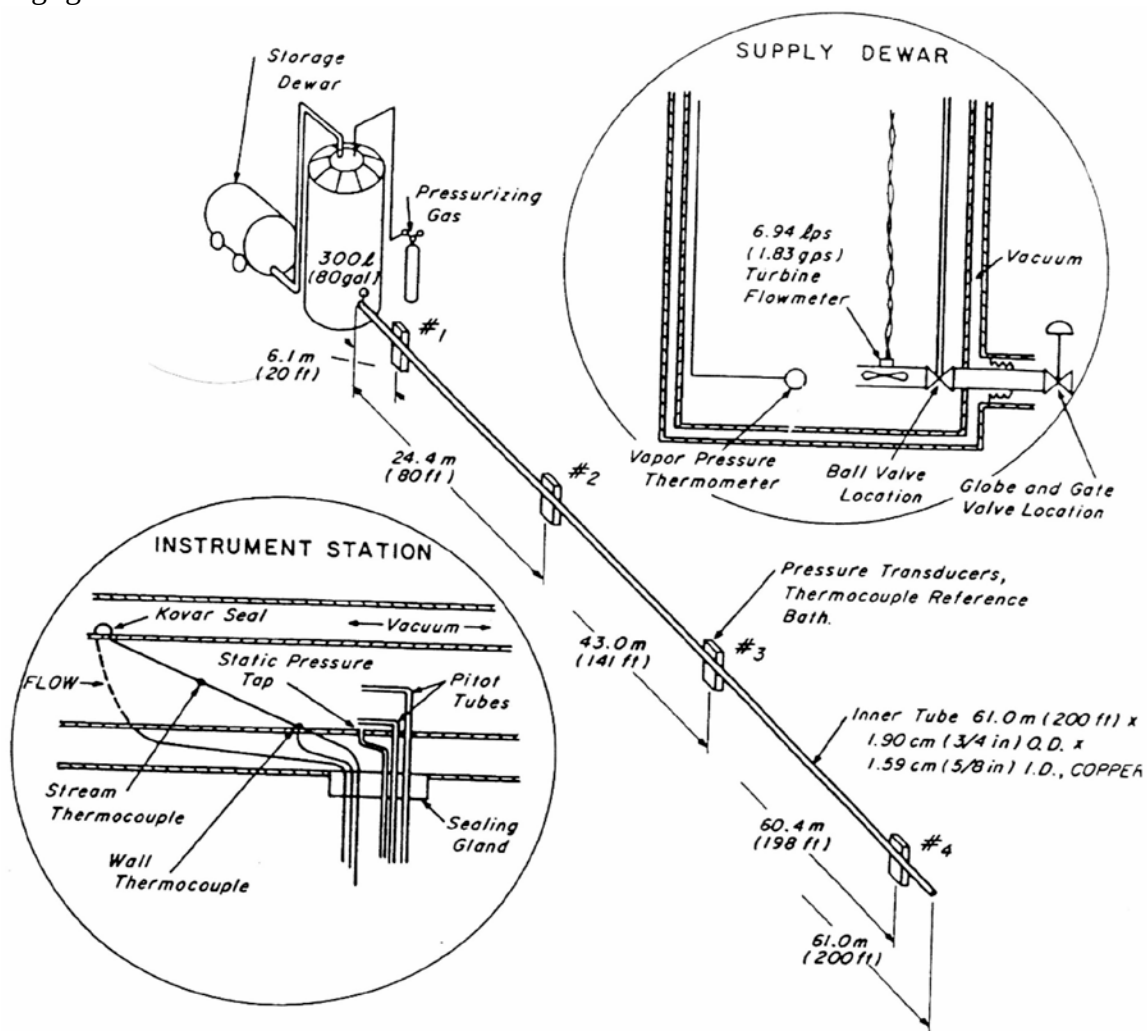


Figure 3.1.1 NBS Test Set-up of Cryogenic Transfer Line

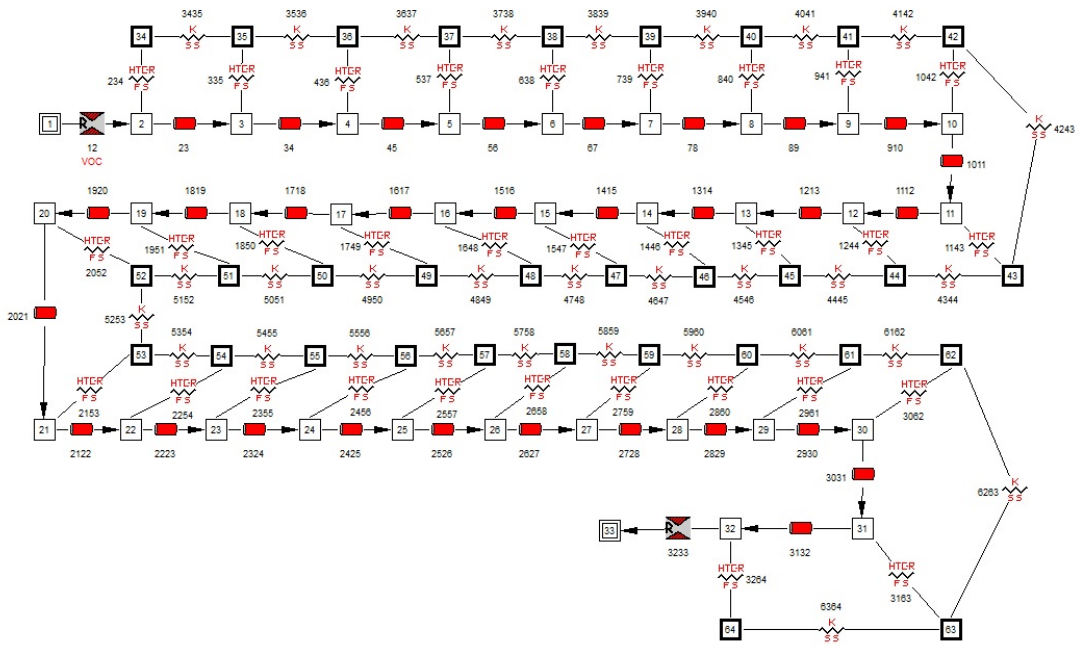


Figure 3.1.2 GFSSP Model of the Cryogenic Pipeline

Results

Figure 3.1.3 shows GFSSP's predicted solid temperatures at each of the four stations as functions of time. These compare well to the measured temperatures. At this driving pressure, the pipeline chills down in about 70 seconds.

Figure 3.1.4 shows GFSSP's predicted flow rate at the entrance (orange) and mid-section (green) of the pipeline. It is noted that the flow rate at the entrance is initially very unstable. This is caused by rapid boil-off of the cryogen, leading to pressure spikes that temporarily reverse the flow rate. Such behavior was also observed during the NBS experiments. The pressure spikes during start-up are seen in Figure 3.1.5, plotting the pressure near the entrance (orange) and mid-section (green) of the pipe.

Figure 3.1.6 shows the predicted quality (vapor mass fraction) at the nodes corresponding to the four experimental measurement stations. The flow is initially all vapor, but then transitions to a saturated liquid-vapor mixture, and eventually becomes pure liquid flow at the upstream stations shortly after their pipe section has completely chilled down. Cross et al [4] have generated an analytical solution to the chilldown problem by assuming constant properties and a constant convection coefficient. Figure 3.1.7 shows how a modified version of GFSSP's Example 14 compares to the analytical solution.

Figure 3.1.3 GFSSP's Predicted Solid Temperatures (°F) Compared to Measurements

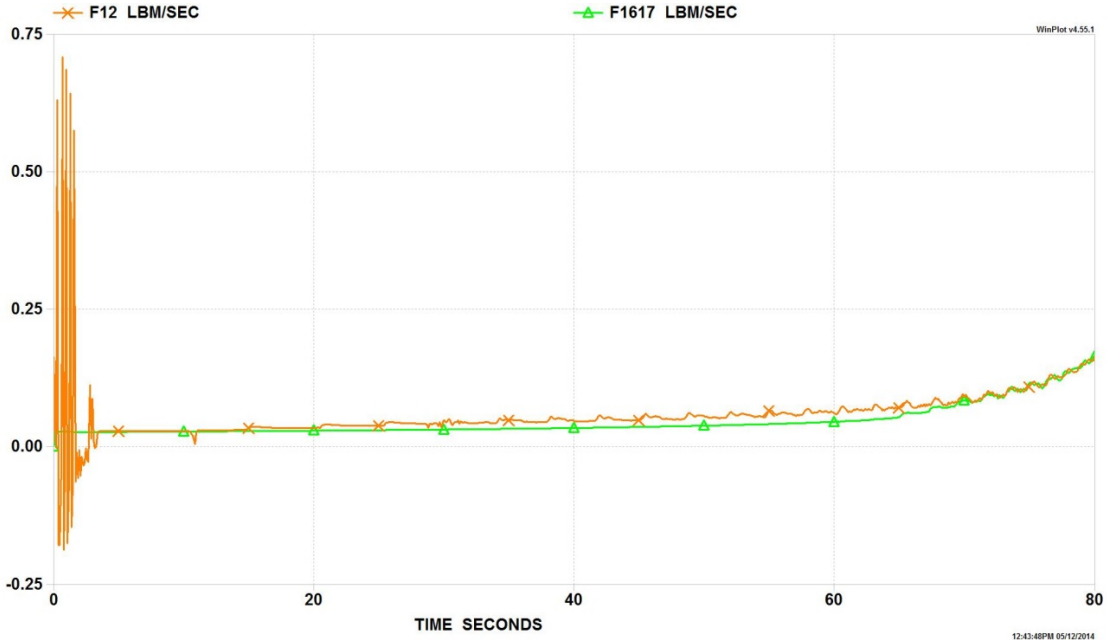


Figure 3.1.4 Predicted Flow Rate (lb/s) at the Pipe Entrance (Orange) and Middle (Green)

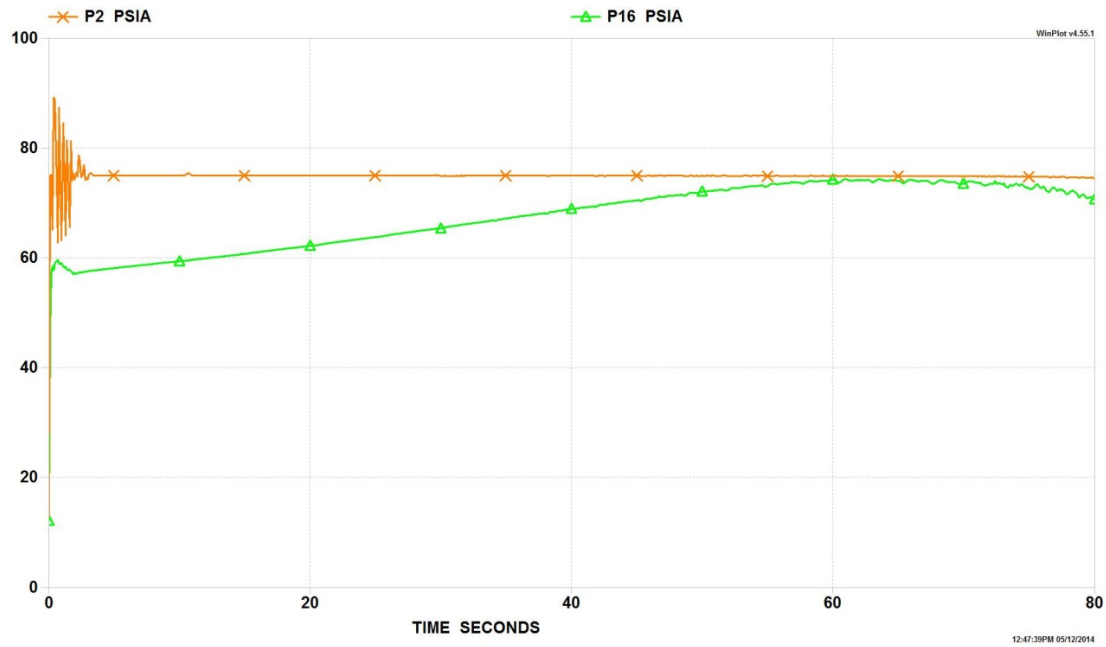


Figure 3.1.5 Predicted Pressure (psia) near the Pipe Entrance (Orange) and Middle (Green)

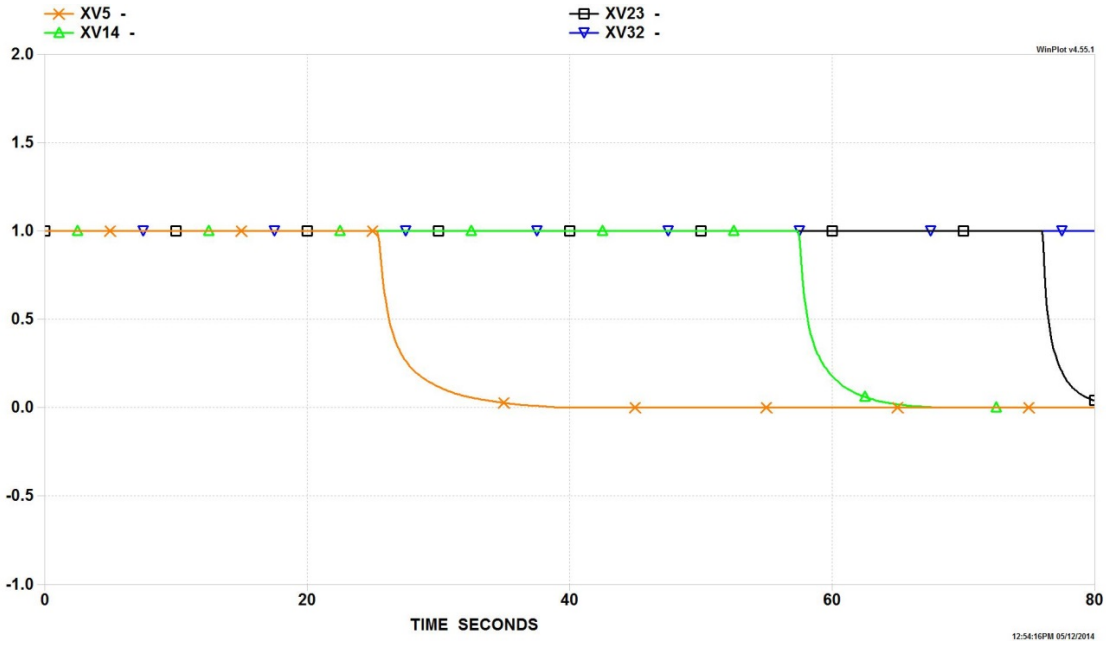


Figure 3.1.6 Predicted Quality (Vapor Mass Fraction) at the Four Measurement Stations

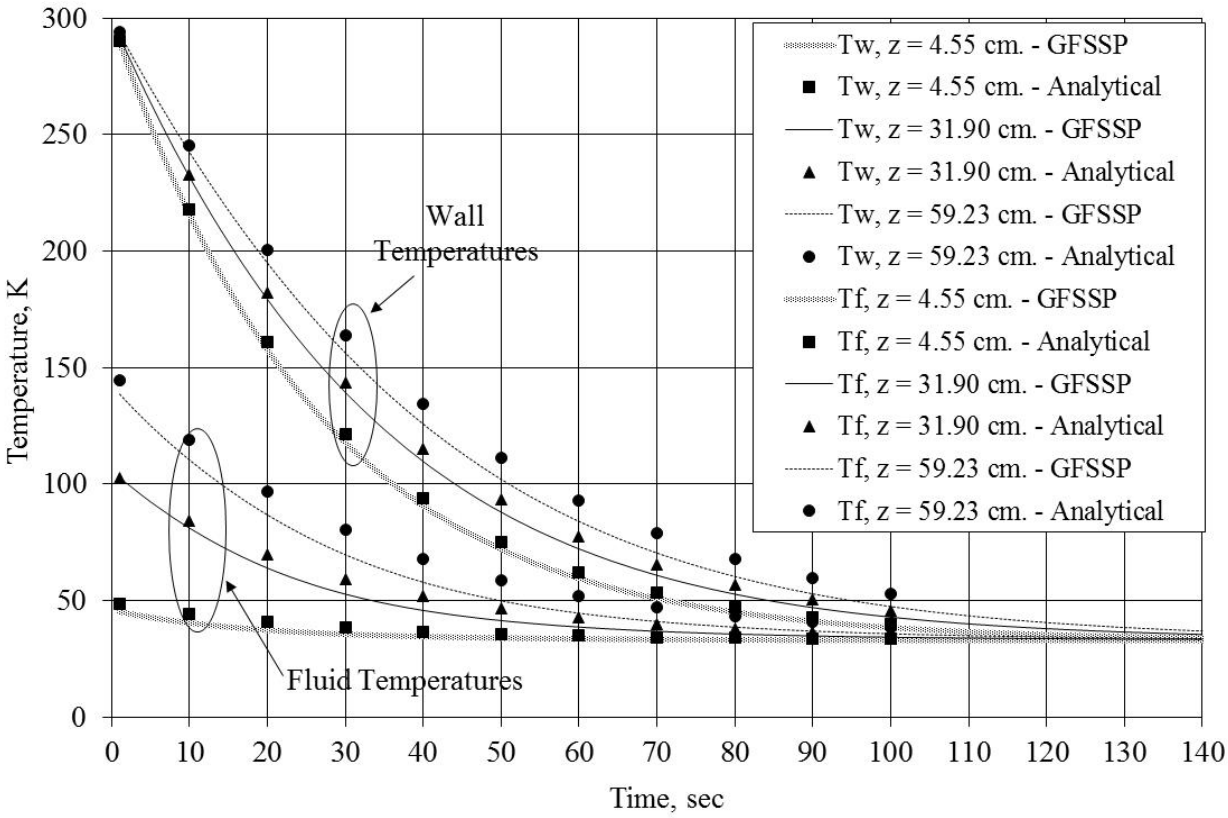


Figure 3.1.7 GFSSP's Predicted Wall and Fluid Temperatures Compared to Analytical Solution by Cross et al.

3.2 Fluid Transients in Pipes Due to Sudden Opening of Valve (EXAMPLE 26)

Problem Description

This example deals with water hammer in a pipe with entrapped air. This problem is different from example 15 as the flow regulating valve is suddenly opened as compared to sudden closing in the case of example 15, and the results are validated against experimental data available in the literature [5]. A long pipe of diameter 1.025 inches is attached to a reservoir of water at one end and closed at the other end with some entrapped air in the other end. A ball valve separates the water from the air as shown in Figure 3.2.1 below. The ball valve is closed until about 0.15 second, and then gradually opens to 100% at about 0.4 second. This example has been set up according to the experimental study done by Lee et al [5]. The two most important controlling parameters for this problem are the reservoir pressure (p_R) and the fractional air length present in the pipe as compared to the total pipe length ($\alpha_g = L_g/L_T$). The initial length for the water volume in the pipe (L_l) is fixed to 20 ft, and initial length of air column in the pipe (L_g) varies from a low of 1.23 ft to 16.23 ft, the value of α ranging from 0.0579 to 0.448 respectively. The ratio of reservoir pressure to the initial pressure of the entrapped air ($P_R = p_R/p_{atm}$) varies in the range of 2 to 7, i.e. the reservoir pressure (p_R) range being 29.4

psi to 102.9 psi. The objective of this study is to predict the transient pressure at different points along the length of the pipe.

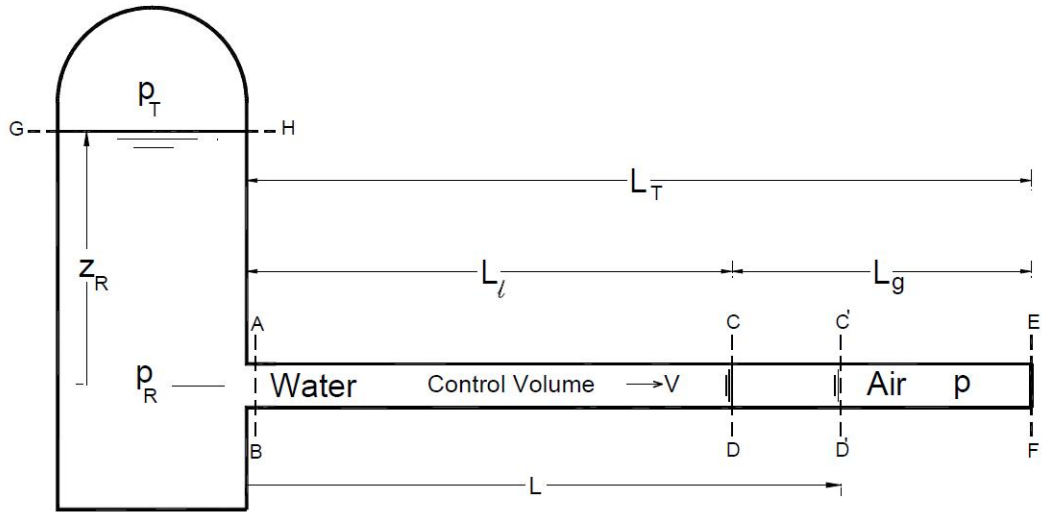


Figure 3.2.1 Schematic of the Water Pipe with Entrapped Air

GFSSP Model

The GFSSP model to represent the flow of water in the pipe is shown in Figure 3.2.2. The pipe sector of length 20 feet (only the water column) is divided into ten uniform pipe segments and one restriction separating twelve nodes. Boundary Node 1 represents the tank (reservoir). A user subroutine interfaces Node 12 to an unseen pseudo control volume containing air only. The pseudo control volume has a fixed mass of air, but the volume changes as it is pressurized owing to the fluctuation of pressure at node 12. Thereby the volume of node 12 changes as the volume of the imaginary control volume changes. The volume change in node 12 is computed by a volume balance between the volume of water and the volume of the entrapped air. The total volume ($V_{tot} = V_{air} + V_{12}$) remains constant, and must be equal to the initial total volume (since the pipe is closed at the other end). The equation of state for water (for node 12) and entrapped air as given below:

(for water)

$$p_{12}V_{12} = Z_{12}(m_{12}R_{12}T_{12}) \qquad (3.2.1)$$

(for air)

$$p_{air}V_{air} = m_{air}R_{air}T_{air} \qquad (3.2.2)$$

Volume balance:

$$(V_{tot})^0 = V_{air} + V_{12} \qquad (3.2.3)$$

Also, across the water-air interface, due to mechanical and thermal equilibrium,

$$p_{air} = p_{12} \text{ and } T_{air} = T_{12} \qquad (3.2.4)$$

The volume of node 12 at any instant can be obtained by combining all the four equations given above.

$$V_{12} = (V_{tot})^0 /(1 + \beta), \text{ where } \beta = \frac{m_{air} R_{air}}{m_{12} R_{12} Z_{12}} \qquad (3.2.5)$$

where V_{air} and V_{12} are the volume of entrapped air and volume of water in node 12.

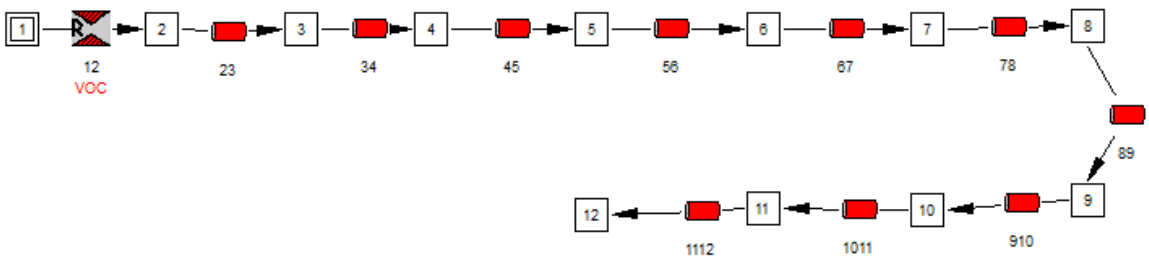


Figure 3.2.2 GFSSP Model of Sudden Valve Opening Experiment of Lee & Martin[5]

Along with the volume adjustment for node 12, there would be a momentum source added to this node subject to air pressure from the entrapped air, and this is done through Equation 3.2.6, as given below.

$$\text{Momentum Source} = -\frac{1}{g_c}\rho_{12}\frac{(V_{12} - V_{12}^*)}{\Delta\tau}u_{12} \qquad (3.2.6)$$

Where, u_{12} is the velocity at the last node, and V_{12} and V_{12}^* are the volume of the 12th node at the current and previous time steps respectively.

The volume source for node 12 is implemented in the subroutine **SORCEM** in the user subroutine through the variable VOLUME(IPN). The momentum source term given in Equation 3.2.6 is prescribed in subroutine **SORCEF** through the variable TERM100 for the last branch (branch 1112). The user subroutine (Ex26.for) is attached in Appendix of this documentation.

The boundary condition at node 1 is provided through the history file Ex26hs1.dat. The ball valve opening control is provided through the area change of the control valve over time through the data file, Ex26vlv.txt.

RESULTS

For the numerical solution a time step of 0.01 s has been used. The operating conditions are: PR = 7, and α_g = 0.45). The GFSSP input file (Ex26.dat) and output file (Ex26.out) are attached in appendix T. Figure 3.2.3 compares GFSSP's predicted pressure at node 12 with that of the experimental data points of Lee and Martin. The predicted results compared very well, and even though the peak pressure amplitude differs by about 7%, the frequencies of pressure oscillations matched very well.

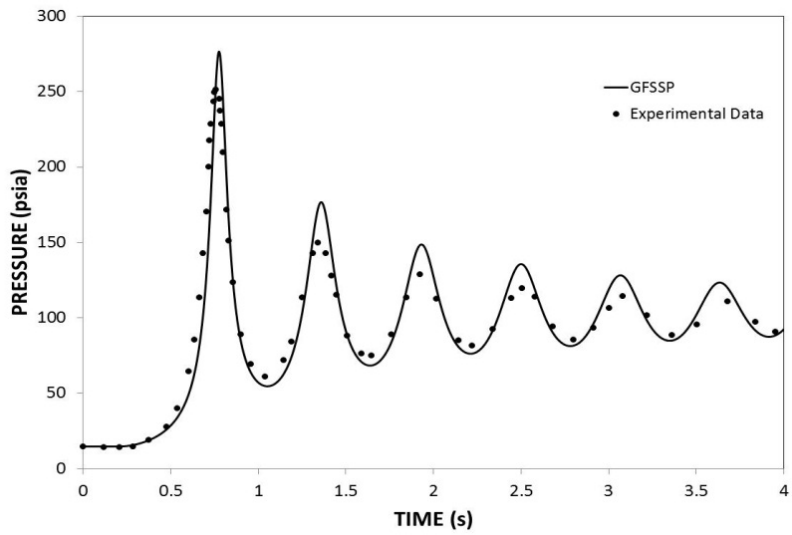


Figure 3.2.3 Comparison of GFSSP and Experimental Data

A Fast Fourier Transform has been conducted in the numerical model to predict the different modal frequencies of the pressure transient and also compared with the experimental data. More details of this example problem is available in Reference [6].

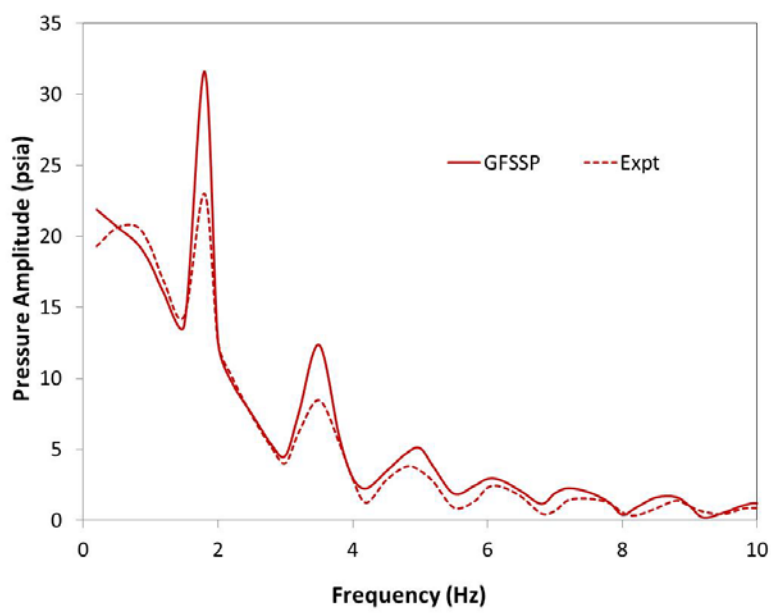


Figure 3.2.4 Fast Fourier Transform for Modal Frequencies

Fast Fourier Transform: A fast Fourier transform algorithm based on Danielson and Lanczos [11] has been used to transform the time domain pressure oscillations into the frequency domain. The computer program “FFT_Multiple.for” has been generated based on this algorithm and can be found in the Ex26 folder of the GFSSP installation directory. The time domain data are saved in a column format with time as an independent variable in the first column and multiple pressures (or some other variables in time domain) in the successive columns. This input data file is called “p_t.txt”, and the results are saved in the file “p_f.xls”. For better accuracy, it is expected that the user will supply N (the number of data) as a power in 2, i.e. $N = 2^m$, for any integer m. However, the present code is capable of taking any number of input data, and computes the number of data to be selected from the input data as the largest possible power of 2.

3.3 Boiling Water Reactor (Example 27)

Problem Considered

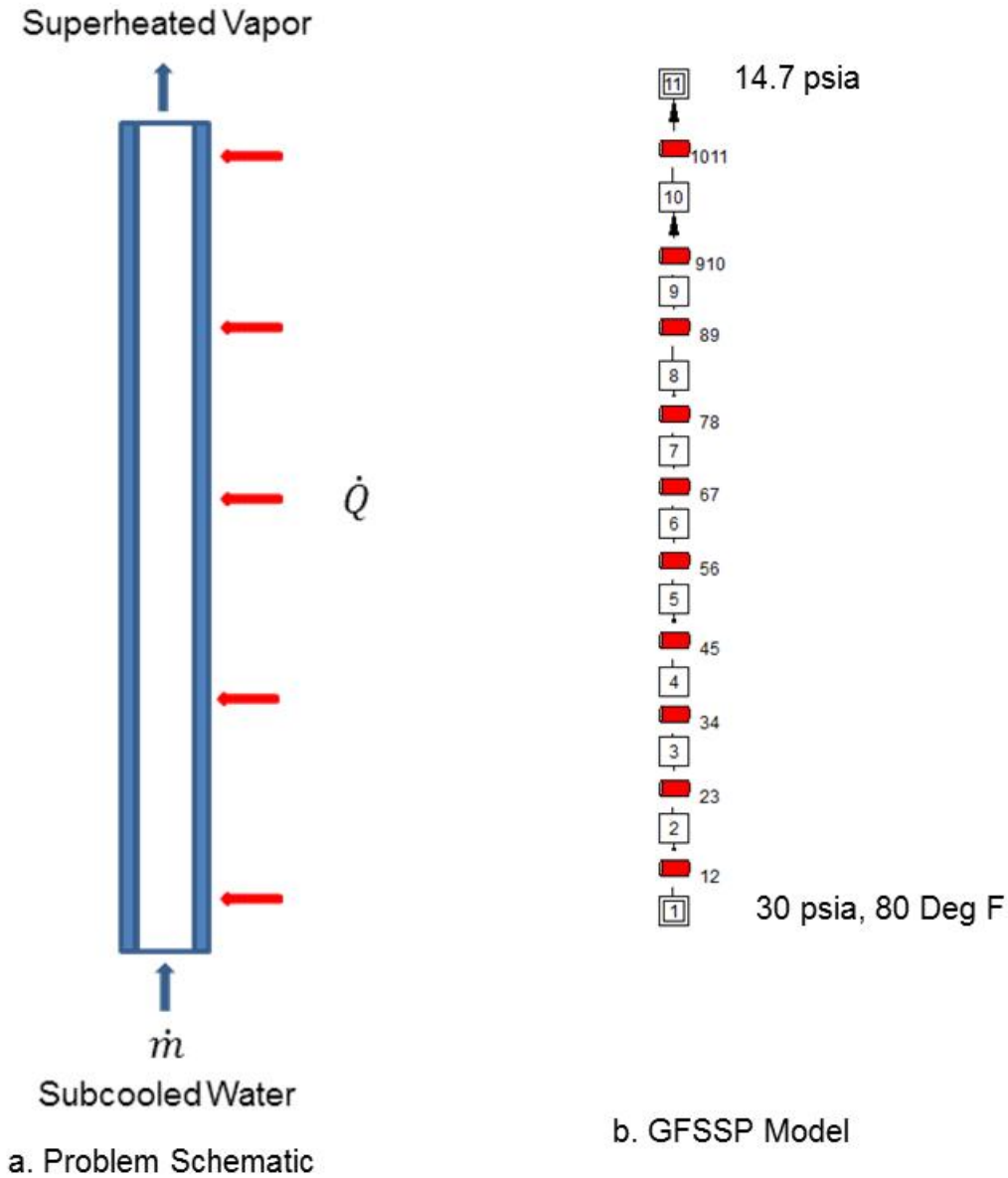


Figure 3.3.1 Schematic and GFSSP Model of Boiling Water Reactor

The purpose of this example is to demonstrate the phase change of water in a vertical tube being heated from outside. This example also demonstrates the use of the Saturated Fluid Property Table for modeling phase change, and the results of the Fluid Property Table have been verified by comparing with the results obtained with WASP, GFSSP's built-in thermodynamic property program for water.

Figure 3.3.1a shows the schematic of the problem considered. A 6-foot long, 1-inch diameter vertical water tube is being heated uniformly along the tube. The water enters the tube in a subcooled state and exits as superheated vapor. The pressures at the inlet and outlet have been prescribed. The model calculates flowrate through the tube, as well as the distribution of pressure, temperature, quality, and other thermo-physical properties of water in the tube.

GFSSP Model

The water tube has been discretized into ten equal segments resulting in ten branches, nine internal nodes, and two boundary nodes where pressures were prescibed (Figure 3.3.1b). Steady-state flow is assumed, and the flowrate is calculated by solving the momentum conservation equation that accounts for the pressure difference between inlet and outlet boundaries, frictional resistance, fluid inertia, and gravity. The external heat load was specified as 130 Btu/lbm in each of the nine internal nodes. The calculated flowrate was 0.113 lbm/sec. Therefore, the total applied heat load was 132.21 Btu/sec or 140 kW. The water enters the reactor at a highly subcooled state (80°F; saturation temperature at 30 psia is 250.3 °F). The water leaves as superheated vapor at 354.4 °F (saturation temperature in Node 10 at 15.94 psia is 216.1 °F). In the GFSSP model, gravity and inertia were activated in all fluid branches. Inertia was activated due to the large change in velocity along the pipe.

The GFSSP model was run with two different fluid options. First, the model was run with the built-in GASP/WASP property package using water (Ex27_with_WASP.vts). The same model was then run with the User Fluid Property Table Option (Ex27_with_Property_Table.vts). In order to run with the property table option, eight property tables are necessary. They are:

1. aakwater2.dat – Conductivity in Btu/sec-ft-R
2. rhowater2.dat – Density in lbm/ft^3
3. emuwater2.dat – Absolute Viscosity in lbm/ft-sec
4. gammawater2.dat – Specific Heat Ratio
5. hwater2.dat – Enthalpy in Btu/lbm
6. swater2.dat – Entropy in Btu/lbm-R
7. cpwater2.dat – Specific Heat in Btu/lbm-R
8. satwater2.dat – Saturation Table for Pressure (psia), Temperature (R) and enthalpy, density, specific heat, absolute viscosity, specific heat ratio, conductivity and entropy of liquid and vapor

Figure 3.3.2 shows the Fluid Option window with eight property tables listed. It may be noted that the saturation table is required when “Phase Change” box is checked.

The method of generating property tables using the NIST-developed property program REFPROP[2] is described in Appendix A.1 of this report.

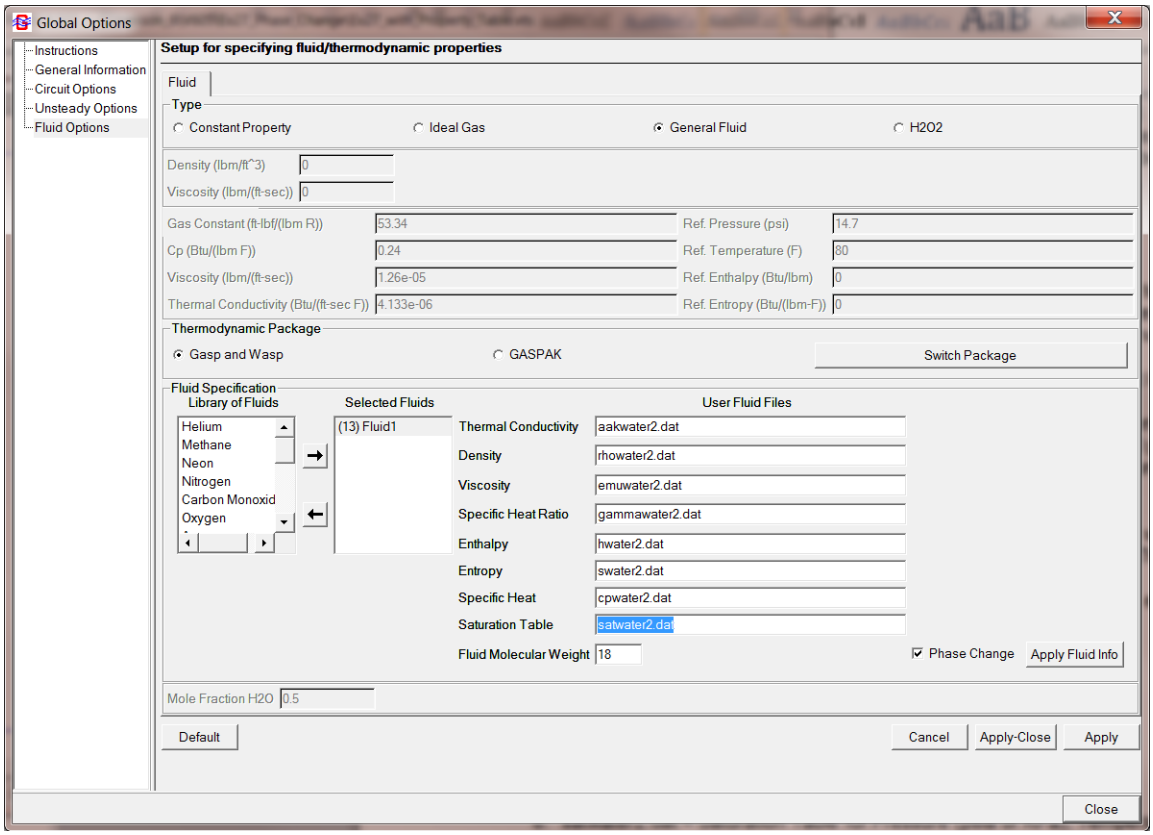


Figure 3.3.2 Fluid Option in VTASC for User Specified Fluid

Results

Figure 3.3.3 shows the distribution of pressure and temperature along the length of the pipe. The distribution of quality and density is shown in Figure 3.3.4. The distribution of velocity is shown in Figure 3.3.5. The pressure drops continuously from inlet to outlet due to frictional and gravitational effects. The temperature, however, first rises steadily until it reaches the saturation temperature. After reaching the boiling temperature, the temperature slightly drops due to the drop in saturation temperature at lower pressure. Once the water completely transforms to steam, temperature increases monotonically. The quality increases continuously in the tube. The quality increases monotonically from 0 to 1 (Figure 3.3.4); the variation of density is very significant due to phase change and is plotted in logarithmic scale. The increase of velocity in Figure 3.3.5 is another evidence of phase change as the mass flowrate is the same at all sections of the tube, but the density decreases as water turns into vapor.

The comparison between the two models is shown in Table 3.3.1.

Table 3.3.1 Comparison between solutions with WASP and Property Table generated by REFPROP

Variables	WASP	Property Table	% Difference
Flowrate (lbm/sec)	0.1125	0.1131	0.5333
P_2 (psia)	29.745	29.747	0.00672
P_5 (psia)	18.517	18.536	0.103
P_{10} (psia)	15.946	15.935	0.069
T_2 (°F)	209.99	209.59	0.191
T_5 (°F)	223.88	223.02	0.384
T_{10} (°F)	354.39	352.55	0.519
X_2 **	0.00	0.00	0.00
X_5	0.3906	0.3909	0.077
X_{10}	1.00	1.0	0.00

** x is the vapor mass fraction or quality

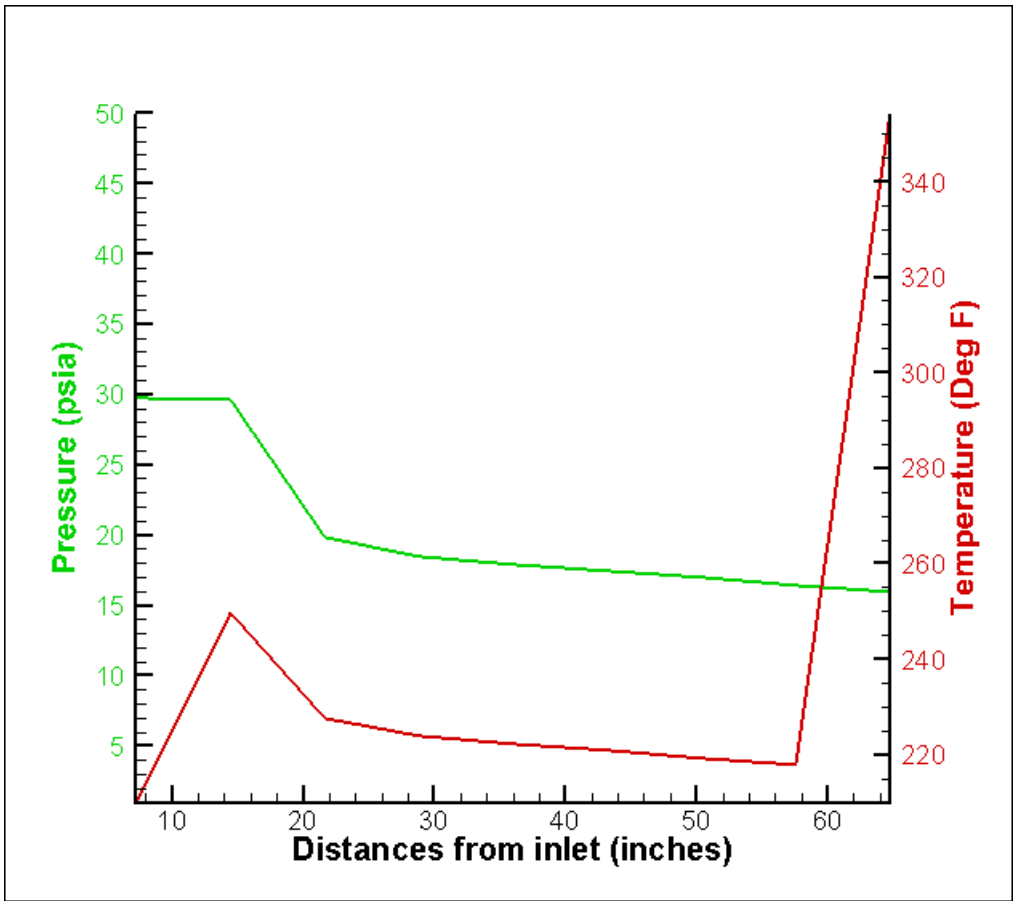


Figure 3.3.3 Pressure and Temperature distribution in the vertical tube

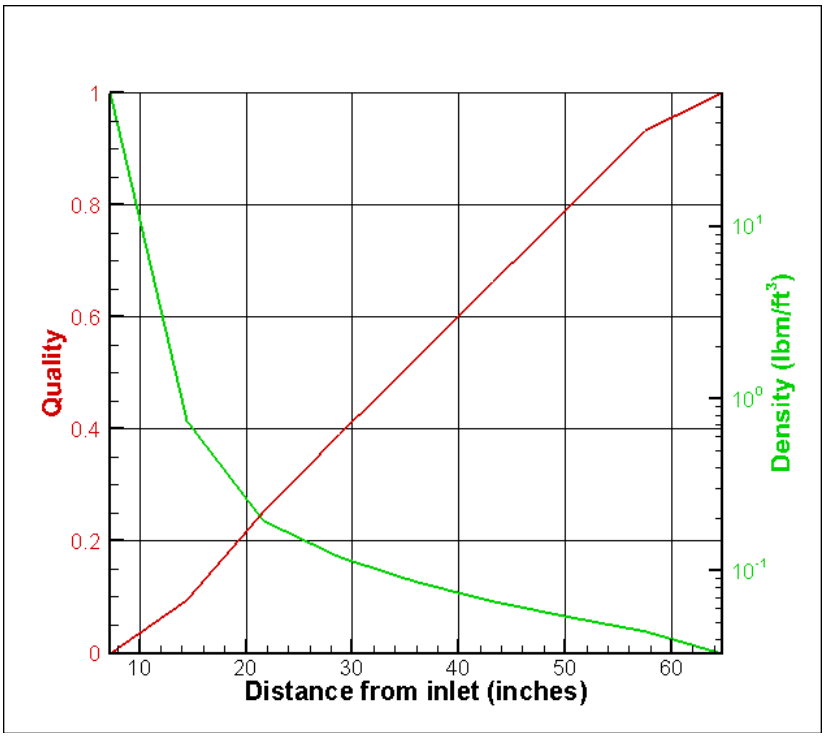


Figure 3.3.4 Distribution of Quality and Density in the Vertical Tube.

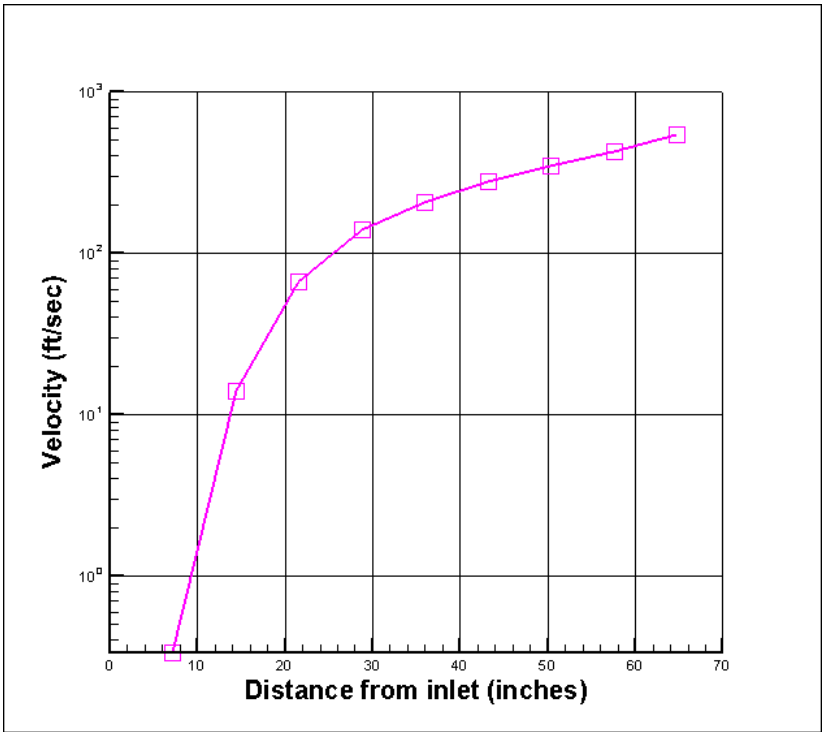


Figure 3.3.5 Distribution of Velocity in the Vertical Tube

Comparison of GFSSP results with hand calculation

The outlet temperature can be calculated by a simple energy balance for a control volume for a given mass flowrate. The energy balance equation can be written as:

$$\dot{m}\left[C_{p,liq}\left(T_{sat}-T_{in}\right)+h_{fg}+C_{p,vap}\left(T_{out}-T_{sat}\right)\right]=\dot{Q} \qquad (3.3.1)$$

From Equation 1, T_{out} can be expressed as:

$$T_{out}=T_{sat}+\frac{1}{C_{p,vap}}\left[\frac{\dot{Q}}{\dot{m}}-C_{p,liq}(T_{sat}-T_{in})-h_{fg}\right] \qquad (3.3.2)$$

From REFPROP, the saturation properties at an average pressure of 22.35 psia were evaluated:

$$T_{sat}=233.89\ °F;\ h_{fg}=956.69\ \frac{Btu}{lb_m}\ ;\ C_{p,liq}=1.0113\ \frac{Btu}{lb_m{}^{\circ}F}\ ;\ C_{p,vap}=0.5103\ \frac{Btu}{lb_m{}^{\circ}F}$$

From Equation 2, $T_{out}=346.92\ °F$, and GFSSP calculates $T_{10}=354.39\ °F$.

The 7.47 degree difference between hand calculation and GFSSP calculation can be attributed to the use of average pressure and the assumption of constant specific heat in the hand calculation.

3.4 No-Vent Tank Chill & Fill Model (Example 28)

Problem Considered

The purpose of this example is to demonstrate the simulation of the no-vent chill and fill method of chilling and filling a cryogenic tank. The practice of tank chilldown in micro-gravity environment is quite different than tank chilldown on the ground. On the ground, under normal gravity, a vent valve on top of the tank can be kept open to vent the vapor generated during the chilling process. The tank pressure can be kept close to atmospheric pressure while the tank is chilling down. In a micro-gravity environment, due to the absence of stratification, such practice may result in dumping a large amount of precious propellant overboard. The intent of the no-vent chill and fill method is to minimize the loss of propellant during chilldown of a propellant tank in a micro-gravity environment. The no-vent chill and fill method consists of a repeated cyclic process of charge, hold, and vent.

During the charge cycle, a small quantity of liquid cryogen is injected into the evacuated tank. Some type of spray nozzle is usually used to break the incoming liquid into droplets. Initially, the liquid flashes due to the low tank pressure, and then the remaining liquid droplets evaporate as they contact warm hydrogen vapor or the tank wall. During the hold period, the circulating flow pattern induced from the spray nozzles provides convective heat transfer from cold vapor to the tank wall. The primary mode of heat transfer during the hold is convection. At the completion of the hold period, the pressure

has risen considerably and the tank is ready to be vented. Since venting occurs as an isentropic blow-down, some additional cooling may be recovered by stage-wise venting. The key parameters of this method are (1) charge magnitude, (2) spray system selection, (3) mass flow rate, (4) hold duration, (5) acceleration environment, (6) desired tank wall temperature, and (7) maximum operating pressure. A reliable and inexpensive mathematical model will help designers to perform a large amount of calculations to optimize the key parameters. A GFSSP model was developed to simulate chilldown of the LH2 tank at the K-site Test Facility [7] and numerical predictions were compared with test data.

The test set-up at the K-site Test Facility, shown in Figure 3.4.1, consists of a test tank, spray system, test tank valves, instrumentation, and the vacuum chamber.

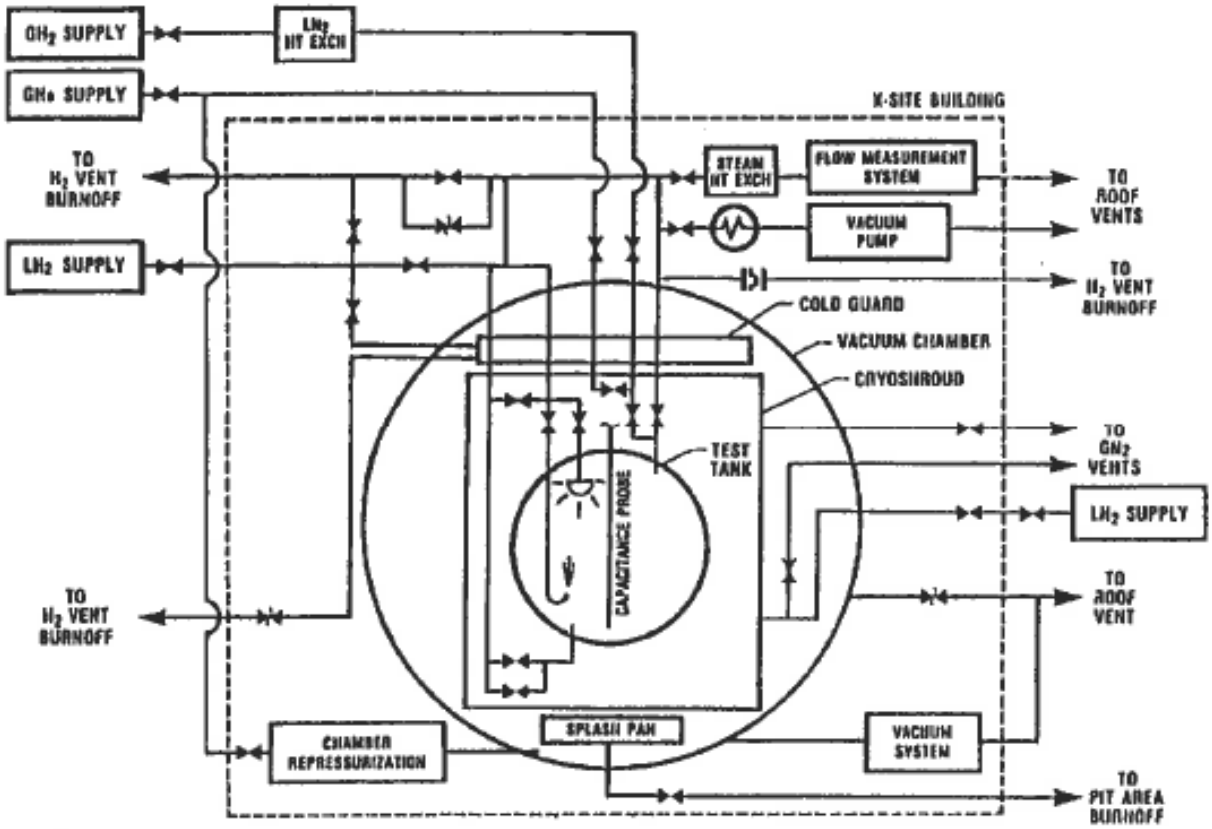


Figure 3.4.1 K-Site Test Set-Up for No-Vent Fill Experiment

The test tank selected was ellipsoidal with an 87 inch major diameter and a 1.2 to 1 major to minor axis ratio. The two ends are joined by a short 1.5 inch cylindrical section. The tank is made of 2219 aluminum chemically milled to a nominal thickness of 0.087 inches. Thicker sections exist where they were required for manufacturing (mainly weld lands). The tank has a 28.35 inch access flange on the top. The tank weighs 329.25 pounds, and the tank's volume is 175 ft^3. The tank was originally designed for a maximum operating pressure of 80 psia. Prior to the start of testing the tank was requalified by pneumatic test for a maximum operating pressure of 50 psia. The tank is covered with a blanket of 34 layers of multi-layer insulation (MLI) made with double aluminized Mylar and silk net spacers, and is

supported by 12 fiberglass epoxy struts. The test environment ambient temperature was uniform and maintained at 530R ± 1R by an electrically heated shroud located outside the tank and inside the vacuum chamber.

GFSSP Model

A nine node tank (Figure 3.4.2) was built to simulate the test as described above. Boundary Node 13 represents the Supply Tank which is supplying hydrogen at -420F. The total flowrate was evenly distributed through Branches 131, 132, 133, 134, 135, 136, 137, 138, and 139. Figure 3 shows the transformation of the actual tank configuration to the model configuration, where the tank geometry was assumed to consist of nine volumes or "tank slices". The total volume and surface area of heat transfer between solid and fluid are identical between the actual and model configurations. Nodes 1 through 9 represent the inside tank volume where propellants reside, transfer from one control volume to another, exchange heat with neighboring solid nodes, and change phases from liquid to vapor and vice versa. The mass and energy conservation equations are solved in the nodes, connected by branches in which the momentum equation is solved. Node 12 is another boundary node that represents the tank outlet. Nodes 10 and 11, branches 910, 1011, and 1112 represent the vent line. Nodes 1 through 9 are connected to metal solid nodes 14 through 22 through fluid to solid conductors that allow convective heat transfer between solid and fluid nodes. The model neglects axial conduction of heat.

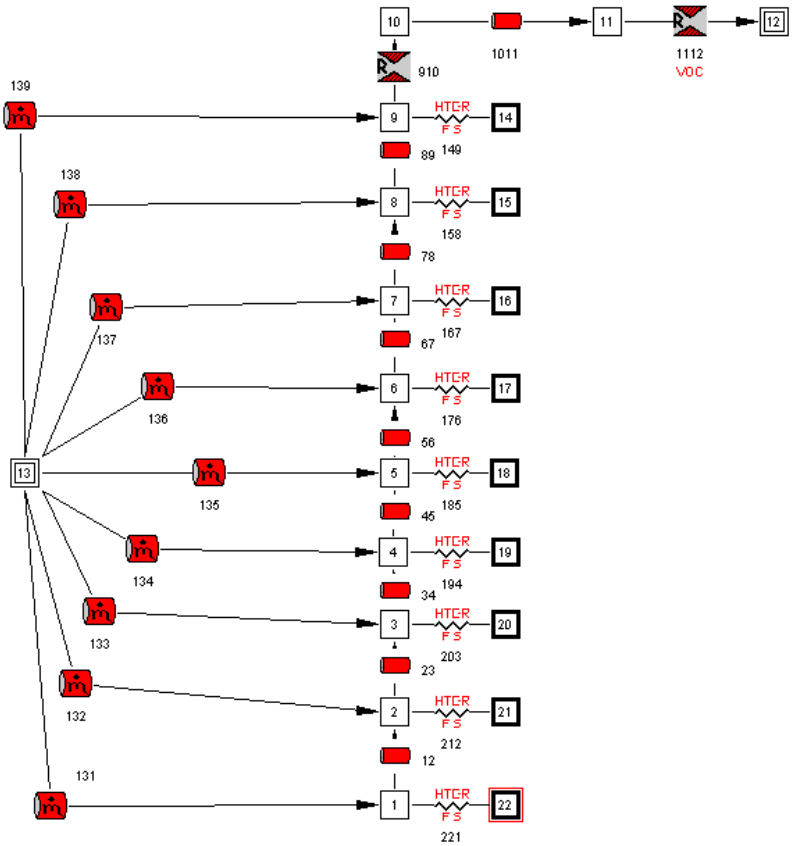


Figure 3.4.2 GFSSP Nine Node Tank Model

Nodes 1 through 9 represent the volume of the test tank with initial pressure set to 1.97 psia and initial temperature set to -19.57 °F. The lengths and diameters of each branch that represent the tank are given in Table 3.4.1. Summing the individual tank slice volumes yields a total tank volume of 301,836.93 in^3. Branch 1112, modeled as a restriction, simulates the vent valve open and close. Node 12 is a boundary node which represents the vent to ambient at Pressure equal to 14.7 psia and Temperature equal to 60F. Conductors 149, 158, 167, 176, 185, 194, 203, 212, and 221 represent the heat transfer from the hydrogen fluid to the aluminum 2219 tank wall. The heat transfer area is the surface area of each tank slice as given in Table 2. Summing the individual heat transfer areas yields a total model tank heat transfer area of 21,599.43 in^2. The heat transfer coefficients were calculated from a free convection correlation [6]. Nodes 14 through 22 represent the tank material and mass which are Aluminum 2219 and the masses given in Table 3. Summing the individual tank slice masses yields a total model tank mass of 329.247 lb. The initial solid wall temperature was set to -19.57 °F.

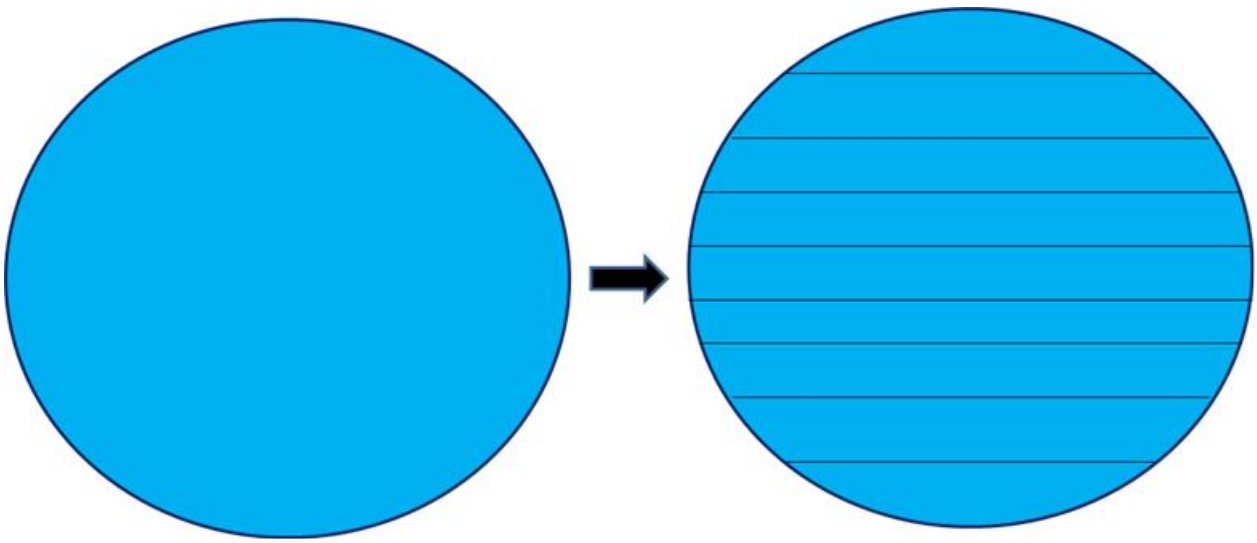


Figure 3.4.3 Conversion of LH2 Tank Geometry

Table 3.4.1. Internal Model Tank Parameters

Branch Number	Branch Length (in)	Branch Diameter (in)	Branch Volume (in^3)
12	11.063	38.59	4,118.72π
23	11.063	62.71	10,876.43π
34	11.063	74.61	15,395.96π
45	11.063	79.88	17,647.73π
56	11.063	79.88	17,647.73π
67	11.063	74.61	15,395.96π
78	11.063	62.71	10,876.43π
89	11.063	38.59	4,118.72π

Table 3.4.2 Tank Slice Heat Transfer Area

Branch	Area (in^2)
221	462.27
212	1684.92
203	2952.94
194	3715.48
185	3968.21
176	3715.48
167	2952.94
158	1684.92
149	462.27

Table 3.4.3 Tank Slice Mass Distribution

Node	Mass (lb)	Node	Mass (lb)
14	7.046	19	56.636
15	25.683	20	45.013
16	45.013	21	25.683
17	56.636	22	7.046
18	60.491		

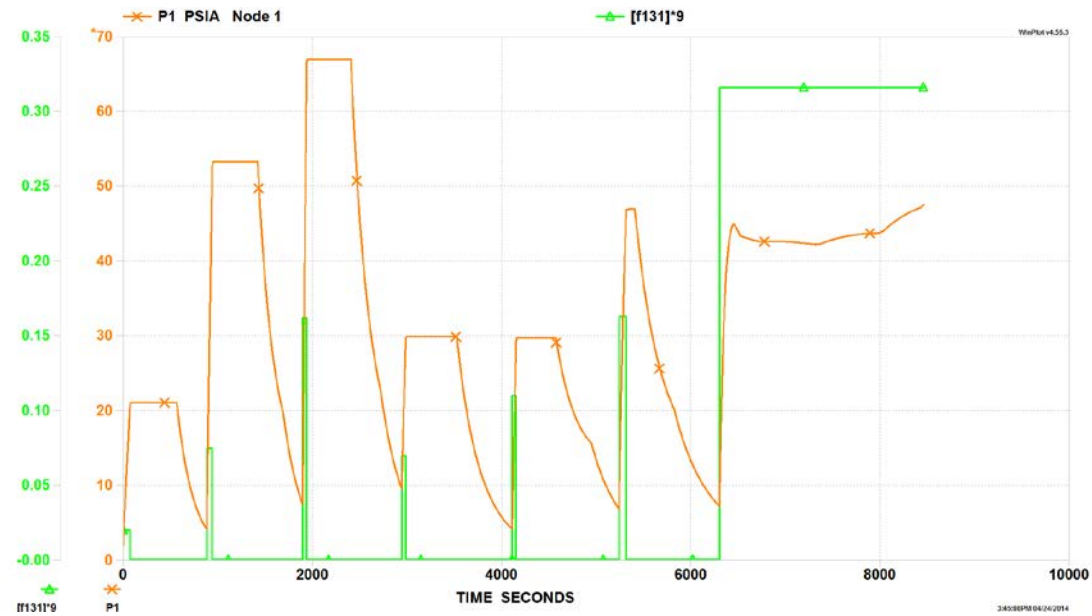


Figure 3.4.4 Specified Inlet Flowrate (green) and Predicted Pressure (Orange) History

Results

Figure 3.4.4 shows the specified inlet flowrate and predicted pressure history. During the charging period, the pressure increases rapidly due to evaporation. During the hold period, pressure remains constant. During venting, pressure in the tank reduces rapidly. During the charging period, the flowrate was assumed constant and was obtained from the reported test data. Figure 3.4.5 shows the predicted mass history of hydrogen during the operation. There is very little hydrogen during the chilling process because of venting.

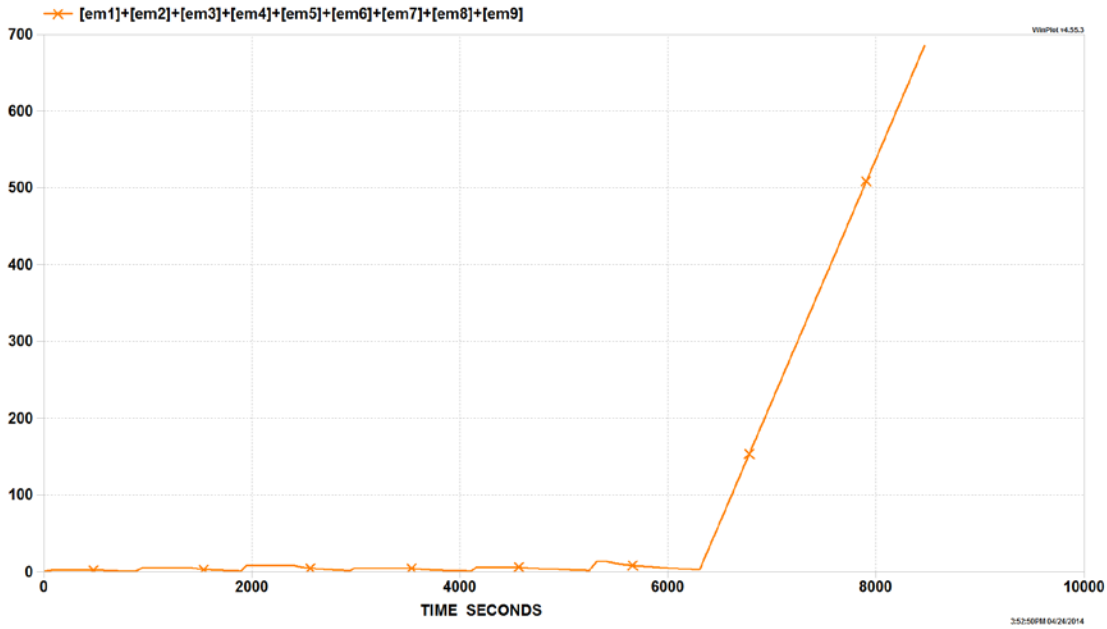


Figure 3.4.5 Predicted Hydrogen Mass History in the Tank

Figure 3.4.6 shows the predicted vent flowrate history during tank chilldown. Vent flowrate reaches a peak value at the opening of the vent valve and diminishes as tank pressure reduces. Figure 3.4.7 shows vapor quality at all 9 nodes during the process. As expected, liquid first forms at the bottom node while the remaining nodes remain superheated. The sudden drop in the quality in the bottom node is due to the blow-down effect. Predicted propellant loss agrees quite well with estimated propellant loss during the test:

- Predicted – 32.5 lbs (9-node model) & 33.5 lbs (1-node model)
- Test – 32 lbs

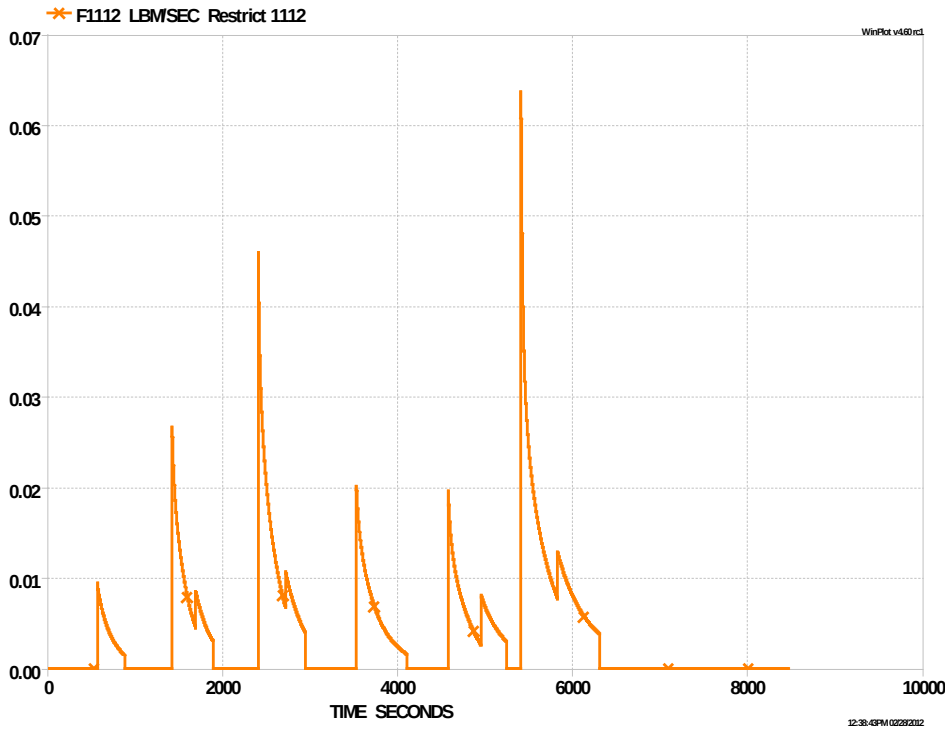


Figure 3.4.6 Predicted Vent Flowrate History during Tank Chilldown

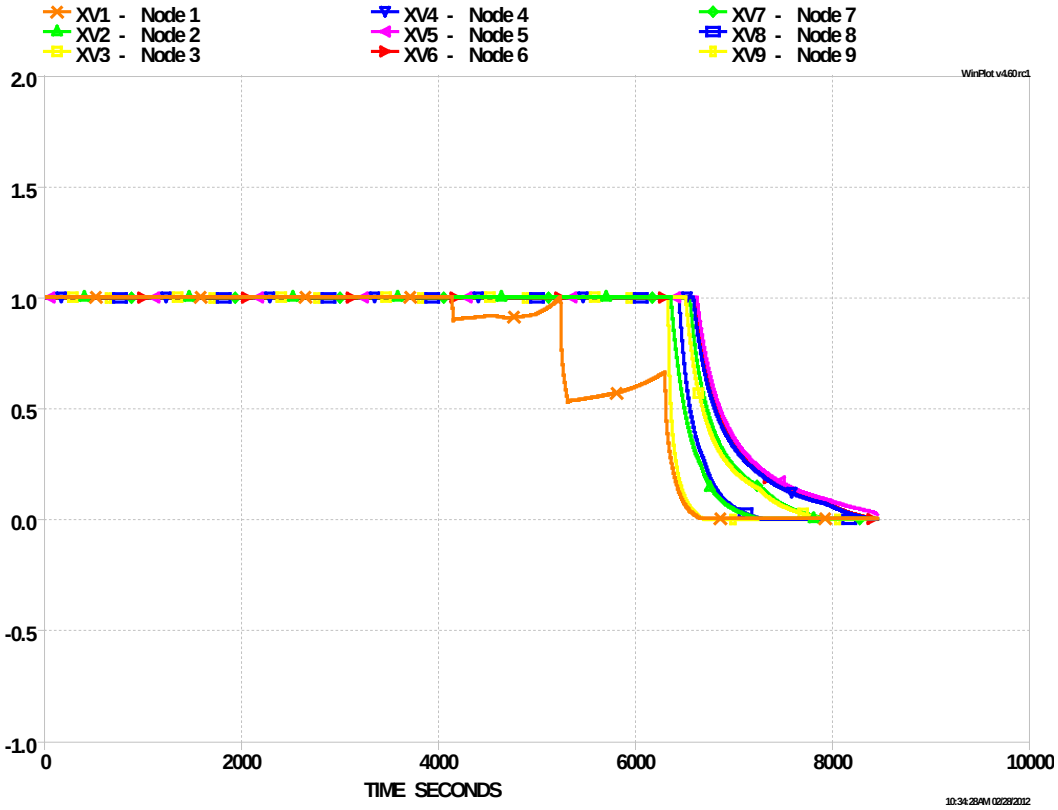


Figure 3.4.7 Predicted Vapor Quality during Tank Chilldown

3.5 Self-Pressurization of a Cryogenic Propellant Tank Due to Boil-off (Example 29)

Problem Considered

The purpose of this example is to demonstrate the simulation of self-pressurization of a Liquid Hydrogen Tank performed under the Multi-Purpose Hydrogen Test Bed (MHTB) program [8]. The purpose of the MHTB program is to test a Thermodynamic Vent System (TVS) to reduce boil-off in a Cryogenic Propellant Tank for long term storage of propellant in space as shown in Figure 3.5.1.

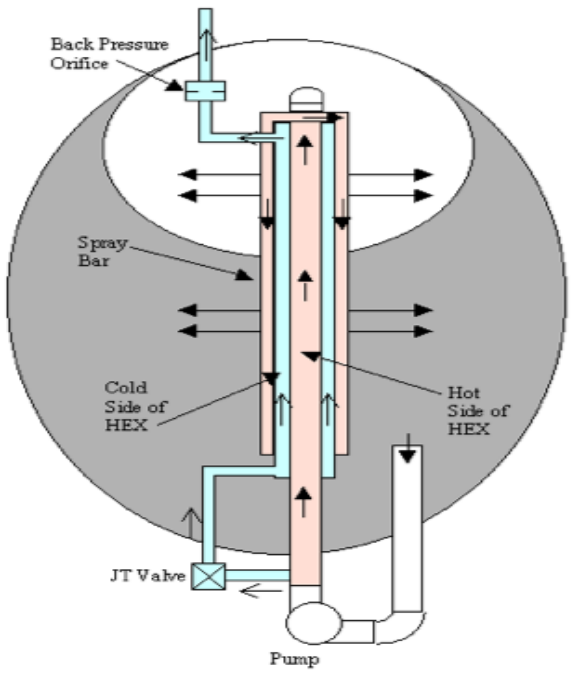


Figure 3.5.1 Thermodynamic Vent System in MHTB Tank

The MHTB 5083 aluminum tank is cylindrical in shape with a height and diameter of 10 feet and elliptic domes in both ends as shown in Figure 3.5.2. It has an internal volume of 639 ft^3 and surface area of 379 ft^2. Initially the tank is allowed to self- pressurize due to boil-off and by not allowing the vapor to vent. Once the pressure reaches the maximum allowable pressure, liquid hydrogen is introduced into the tank through the spray bar. The pressure starts falling due to heat transfer, and when the pressure reaches the minimum allowable pressure, the spray is stopped and the tank is allowed to self-pressurize and thus TVS cycle continues. The purpose of the GFSSP model is to simulate the initial self-pressurization when ullage pressure rises from the initial tank pressure to the upper bound pressure when the spray starts. The GFSSP model results were then compared with the test data. A 50% Fill Level case was modeled to simulate the self-pressurization test.

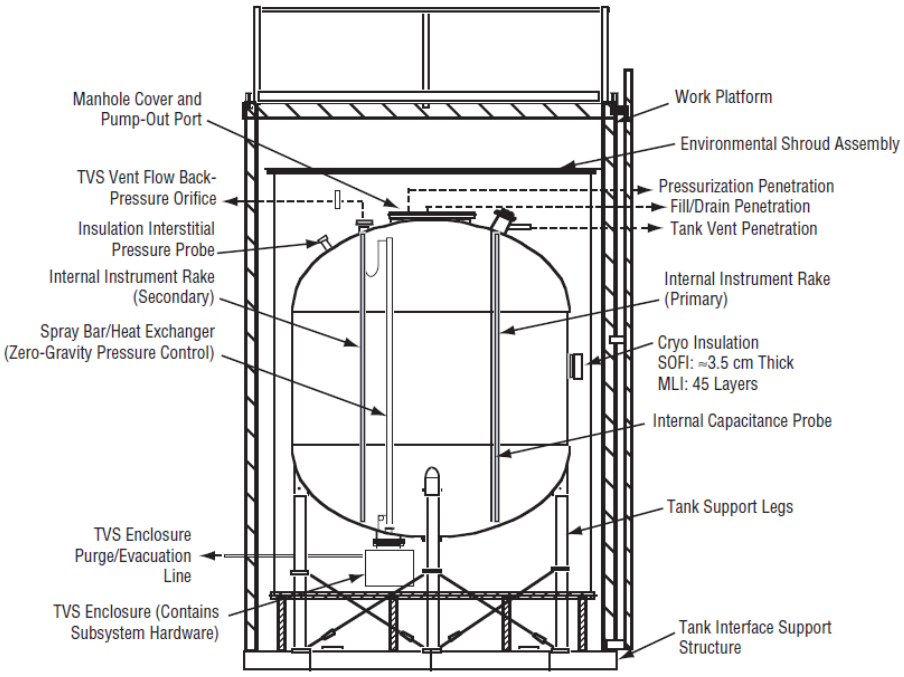


Figure 3.5.2 MHTB Test Tank and Supporting Hardware Schematic

GFSSP Model

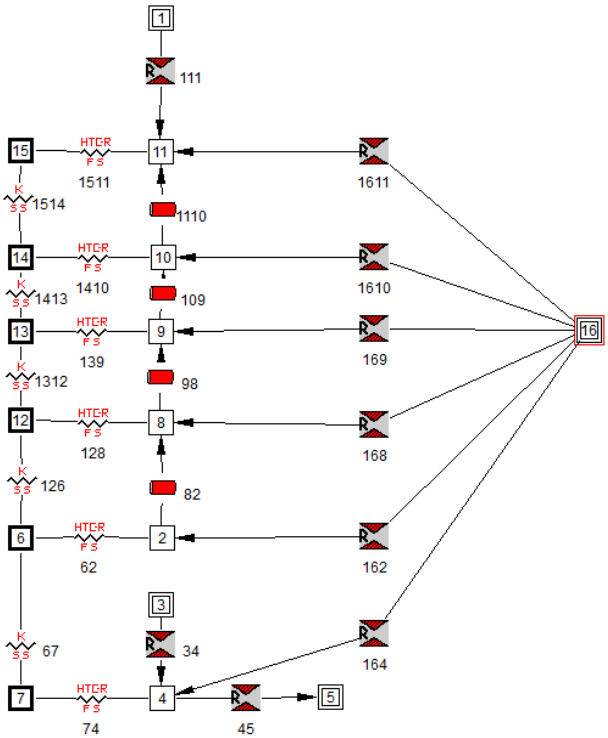


Figure 3.5.3 GFSSP Model

Figure 3.5.3 shows the GFSSP model of self-pressurization in the MHTB Tank at the 50% fill level. Node 4 represents liquid hydrogen; Nodes 2, 8, 9, 10, and 11 represent the ullage

at different fill levels. Node 3 is a pseudo-boundary node separating liquid hydrogen from vapor hydrogen in the ullage space. Branches 45, 164, 162, 168, 169, 1610 and 1611 are for introducing liquid hydrogen into the tank through the TVS spray bar. These branches are currently inactive during self-pressurization of the tank. Node 7, 6, 12, 13, 14 and 15 are solid nodes representing the aluminum tank wall. Solid Node 7 is connected with liquid hydrogen stored in Fluid Node 4. In this model, heat leak through insulation is calculated in the User Subroutine and applied in the solid nodes as a source term.

User Subroutine

In this model a User Subroutine was used a) to model evaporative mass transfer at the liquid-vapor interface, b) to calculate the heat transfer coefficient between the wall and the fluid nodes, and c) to calculate heat transfer through the MLI blankets.

<u>Evaporative Mass Transfer at Liquid-Vapor Interface</u>

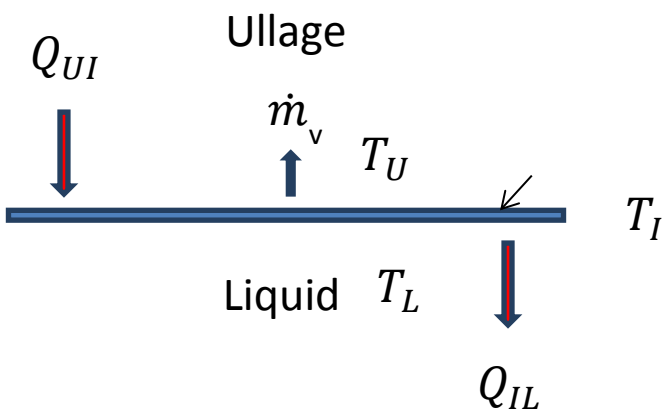


Figure 3.5.4 Schematic of Evaporative Mass Transfer Process at Liquid-Vapor Interface

Figure 3.5.4 shows the evaporative mass transfer process at the liquid-vapor interface. It is assumed that evaporation takes place at the interface in a thin film which contains saturated vapor at ullage pressure. The ullage contains superheated vapor at temperature, T_U. The interface temperature T_I is the saturation temperature at ullage pressure. The Ullage to Interface heat transfer, Q_{UI} can be expressed as:

$$Q_{UI} = h_{UI}A(T_U - T_I), \qquad (3.5.1)$$

where

$$h_{UI} = \frac{k_U}{h_L}, \qquad (3.5.2)$$

k_U is the conductivity of vapor in the ullage and h_L is the length scale.
The Interface to Liquid heat transfer, Q_{IL} can be expressed as:

$$Q_{IL} = h_{IL}A(T_I - T_L) \qquad (3.5.3)$$

Where

$$h_{IL} = \frac{k_L}{h_L} \qquad (3.5.4)$$

k_L is the conductivity of vapor in the ullage and h_L is the length scale.
The evaporative mass transfer is given by:

$$\dot{m} = \frac{Q_{UI} - Q_{IL}}{h_{fg}}, \qquad (3.5.5)$$

where h_{fg} is the enthalpy of evaporation.

$\dot{m}$ was computed in Subroutine SORCEM. The variation of h_{fg} and T_I due to pressure change was neglected in this model.

<u>Heat transfer thru MLI blankets</u>

Heat Transfer thru MLI can be expressed by the Modified Lockheed Equation [9]:

$$q = \left[\frac{C_s(0.017 + 7.0E-6*(800.0 - T_{avg}) + 2.28E-2*\ln(T_{avg}))(N*)^{2.63}(T_h - T_c)}{N_s} + \frac{C_r \varepsilon (T_h^{4.67} - T_c^{4.67})}{N_s} + \frac{C_g P (T_h^{0.52} - T_c^{0.52})}{N_s} \right] \quad (3.5.6)$$

The actual heat transfer, however, was calculated by introducing a Degradation Factor, D_f, which is typically in the order of 3 to 5 for liquid hydrogen tank. The heat transfer rate thru MLI was expressed as:

$$q_{MLI} = D_f q \quad (3.5.7)$$

Where,

Constants

C_s = 2.4E-4
C_r = 4.944E-10
C_g = 14600.

Variables and units

q heat flux through MLI, W/m²
T_h hot boundary temperature, K
T_c cold boundary temperature, K
T_{avg} average of hot and cold boundary temperatures, K

N* MLI layer density, layers/cm
N_s number of MLI layers
ε MLI layer emissivity, ε = 0.031
P interstitial gas pressure, torr

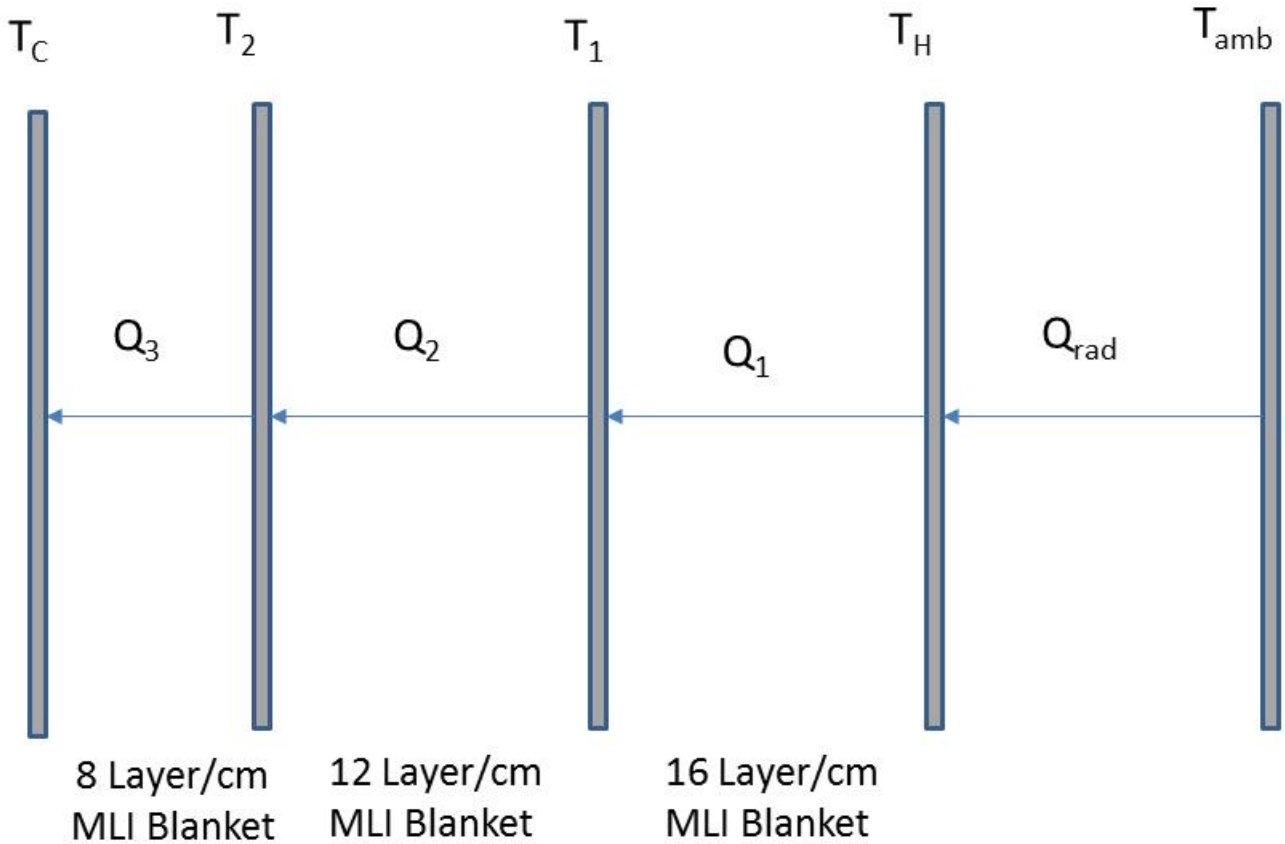


Figure 3.5.5 Multi-Layer Insulation Modeling Methodology

Typically, several MLI blankets constitute MLI insulation. The mathematical modeling methodology is shown in Figure 3.5.5. According to the law of energy conservation:

$Q_{rad} = Q_1 = Q_2 = Q_3$,

where radiative heat transfer is given as:

$$q = \frac{\sigma\left(T_{amb}^4 - T_{outer}^4\right)}{\frac{1}{\varepsilon_{MLI}} + \frac{1}{\varepsilon_{shrd}} - 1} \qquad (3.5.8)$$

The law of energy conservation can also be expressed as:

$Q_2(T_1, T_2) - Q_3(T_2, T_c) = 0$ (3.5.9)

$Q_1(T_h, T_1) - Q_2(T_1, T_2) = 0$ (3.5.10)

$Q_{rad}(T_{amb}, T_H) - Q_1(T_H, T_1)$ (3.5.11)

These three equations [(3.5.9), (3.5.10), and (3.5.11)] are the governing equations to calculate temperature at the outer boundary and two intermediate temperatures by the

Newton-Raphson method. A subroutine MLI_HEAT_RATE was developed to solve these equations. Figure 3.5.6 shows the flowchart of the subroutine which was called from Subroutine SORCETS.

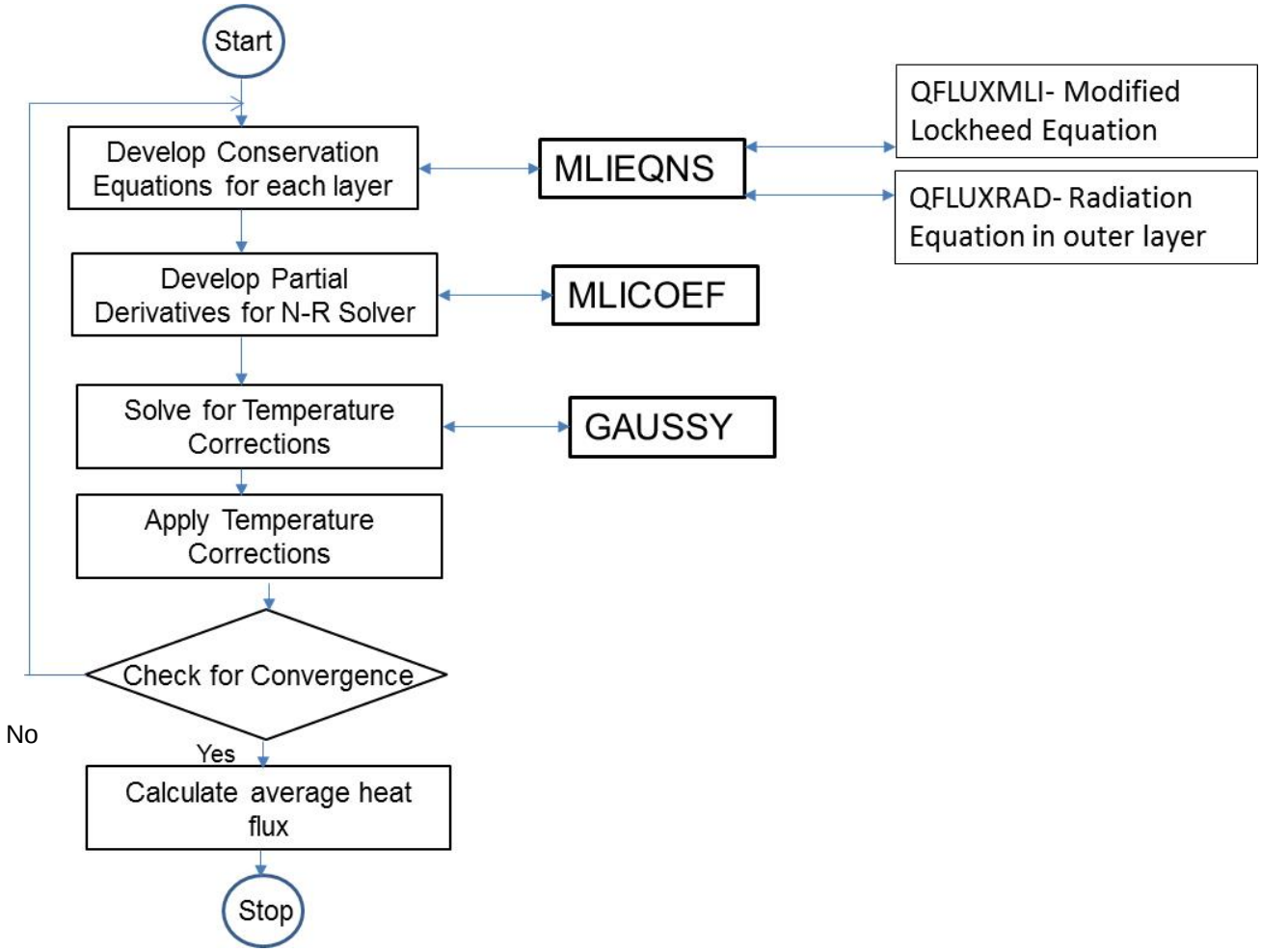


Figure 3.5.6 Flowchart of MLI_HEAT_RATE Subroutine

<u>Heat Transfer Coefficient Correlation</u>

The heat transfer coefficient between wall and ullage was computed from a natural convection correlation for a vertical plate [10]. The set of equations used for this correlation is given below:

$Nu = [(Nu_l)^m + (Nu_t)^m]^{1/m}$ m = 6 (3.5.12)

$Nu_t = C_t^V Ra^{1/3}/(1 + 1.4 \times 10^9 Pr/Ra)$ (3.5.13)

$Nu_l = \frac{2.0}{ln(1+2.0/Nu^T)}$ (3.5.14)

$Nu^T = \overline{C}_l Ra^{1/4}$ (3.5.15)

$C_t^V = \frac{0.13Pr^{0.22}}{(1+0.61Pr^{0.81})^{0.42}}$ (3.5.16)

where,

$$Gr = \frac{L^3 \rho^2 g \beta \Delta T}{\mu^2}\,;\; Pr = \frac{C_p \mu}{k}\,;\; Ra = GrPr$$

$$Nu = \frac{hL}{k}$$

Results

Figure 3.5.7 shows the comparison between GFSSP predictions (in green and blue) and the MHTB Test Data (in orange). GFSSP predictions of pressure are shown for a Degradation Factor of 1 and 2.8. The Degradation Factor is used to multiply Equation 3.5.7 to represent the degradation of performance of the MLI. It is observed that a Degradation Factor of 2.8 matches the test data well.

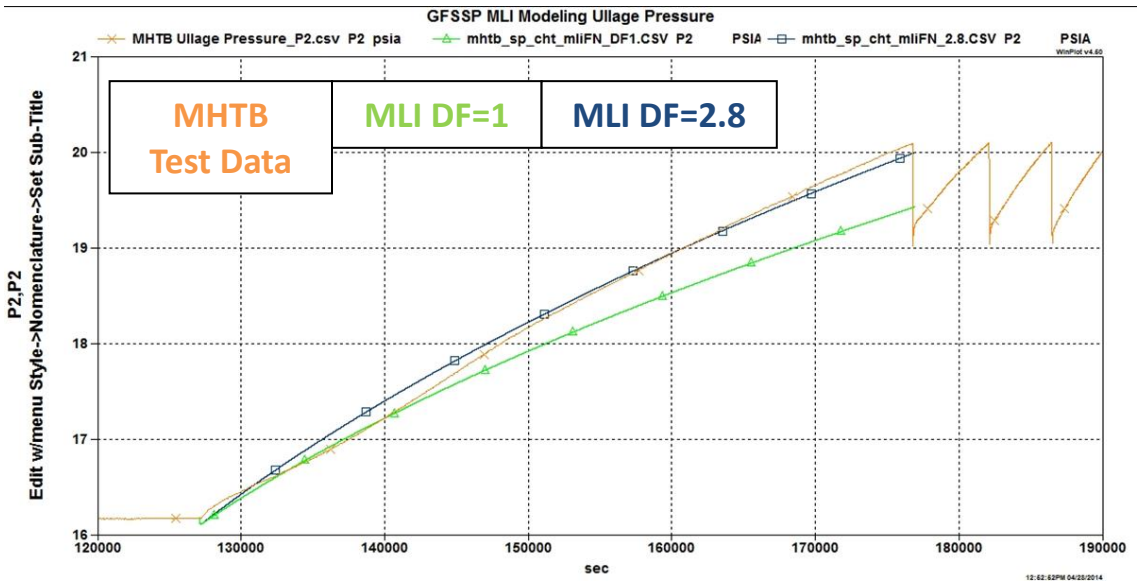


Figure 3.5.7 Application Results for MHTB Self Pressurization Model

3.6 Modeling Solid Propellant Ballistic with GFSSP (Example 30)

This example demonstrates the use of GFSSP to model Solid Propellant Ballistic, which includes burning of solid propellant and expansion of the gas through a converging-diverging nozzle. Figure 3.6.1 shows the schematic of the chamber, orifice, and nozzle. The GFSSP model of the schematic is shown in Figure 3.6.2.

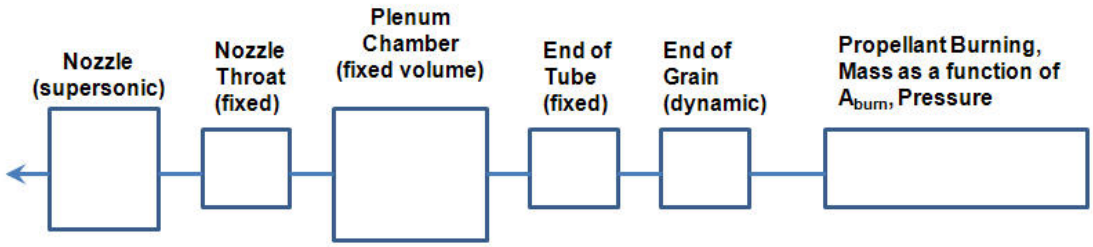


Figure 3.6.1 Schematic of a typical solid propellant rocket motor

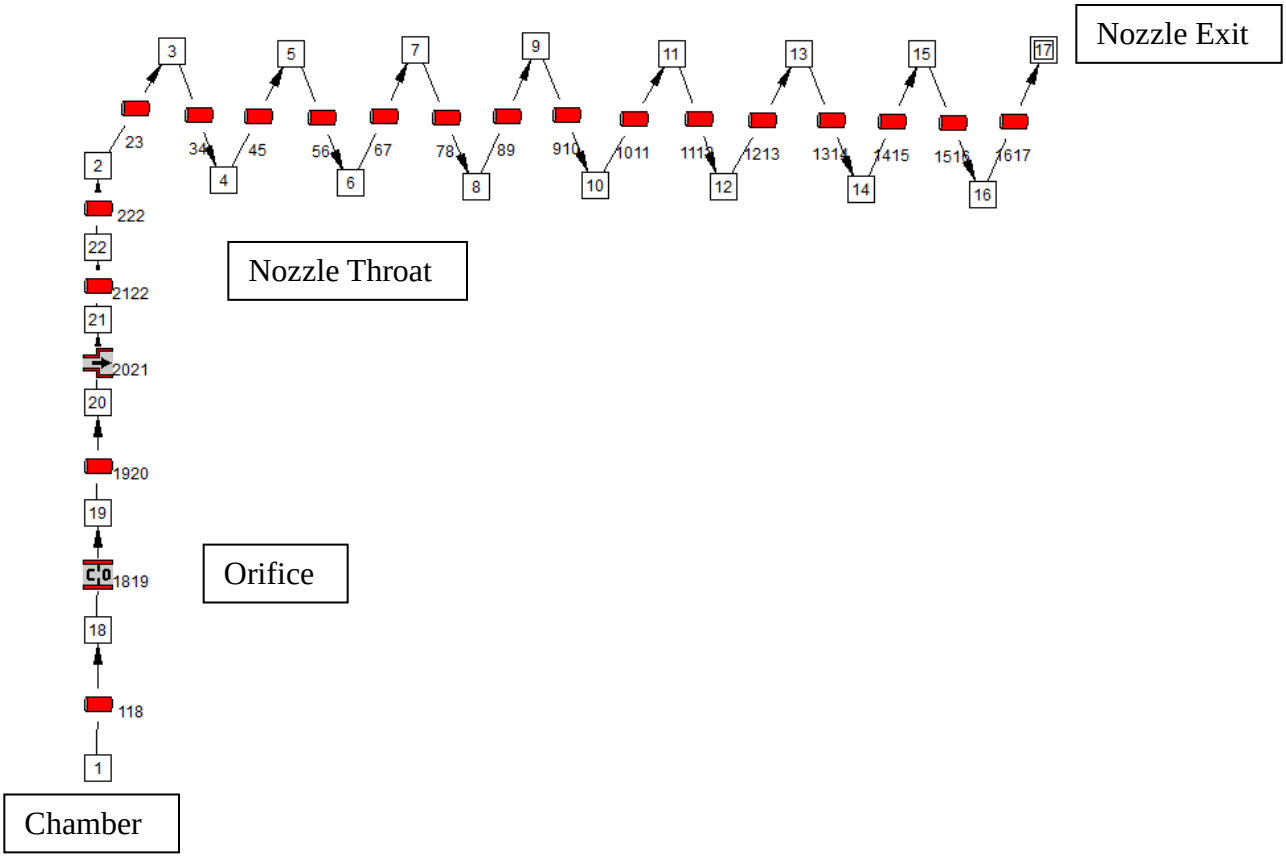


Figure 3.6.2 GFSSP model of Solid Propellant Rocket Motor

The model consists of a propellant grain, orifice, and transition chamber followed by a converging-diverging nozzle. The model requires development of a user subroutine that calculates the burn rate as a function of chamber pressure and estimates the change in the area as propellant burns.

The purpose of the model is to calculate the pressure distribution, thrust, and area change during the burning of solid propellant.

The propellant burning rate (in/sec) was expressed as:

$$\dot{r} = a_{BR} P_C^{n_{BR}}$$

Where a_{BR} = 0.0687 and n_{BR} = 0.3
Propellant Density, ρ_{prop} = 0.06 lbm/in^3

The mass source in the chamber node (Node 1) was calculated from the propellant burning rate and propellant density in user subroutine BNDUSER.

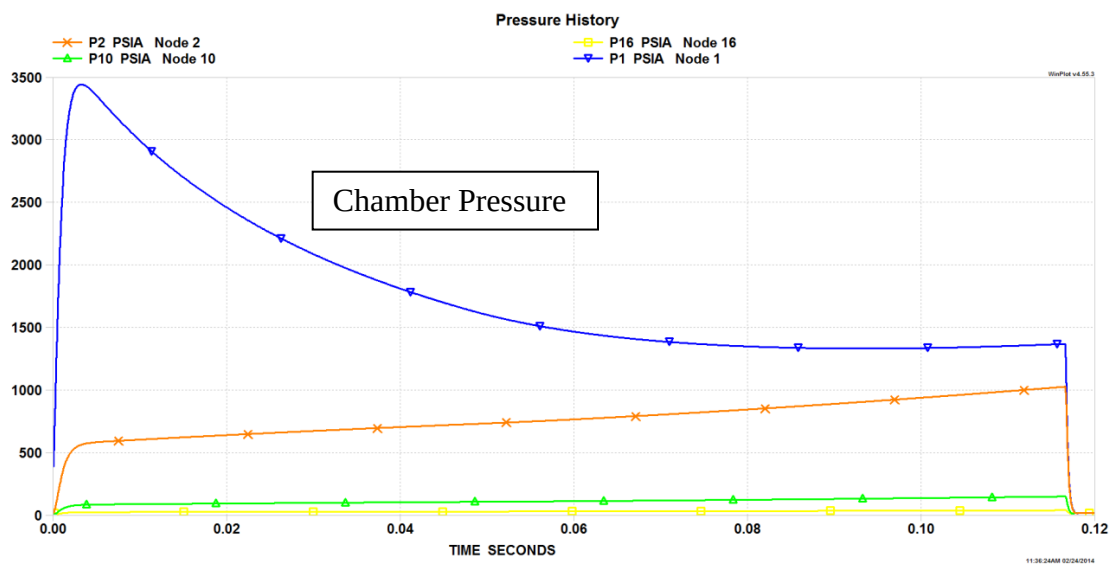


Figure 3.6.3 Pressure history during the Propellant Burn and Tail-off

Figure 3.6.3 shows the pressure history in the chamber as well as at sections downstream of the chamber. The rapid pressure rise during the start and tail-off after the completion of the burn is predicted. Figure 3.6.4 shows the history of burning rate (expressed as a mass source in lb/sec) and the flow rate at the nozzle exit. The cause of the pressure rise can be explained from this figure as the mass source exceeds the flow through nozzle exit at the start. Figure 3.6.5 shows the history of area change during propellant burn. Once the propellant grain radius reaches its maximum, there is no further burning and area remains constant during the tail-off. Figure 3.6.6 shows the history of thrust during propellant burn and tail-off.

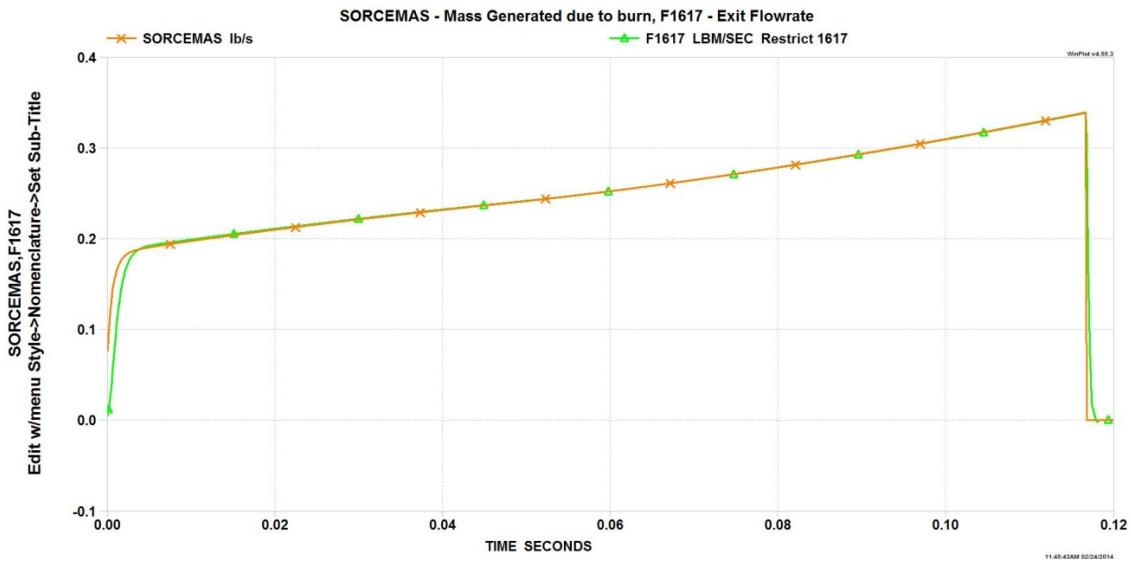


Figure 3.6.4 History of Burning Rate and Flow Rate at Nozzle Exit

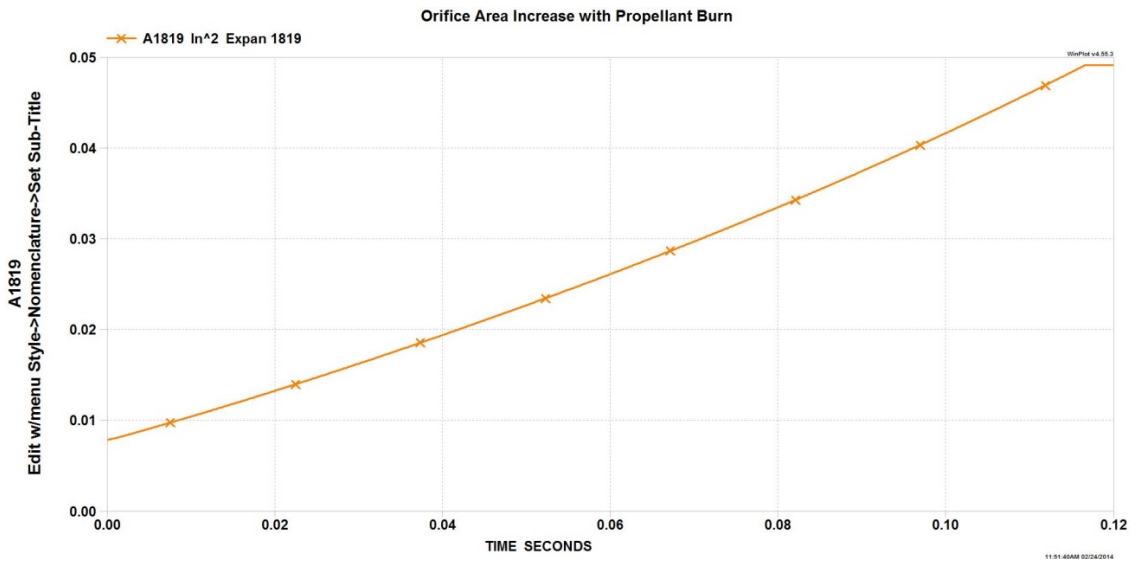


Figure 3.6.5 History of Area Change during Propellant Burn

Highlights of this model's control parameters that ensure stable numerical solution are:

Very Small Time Step (DTAU = 0.0001 sec)
Stringent Convergence Criterion (CC = 1e-07)
Heavy Under-Relaxation on Density (RELAXD = 0.05)
During Tail-Off, RELAXNR was set to 0.3 in User Subroutine

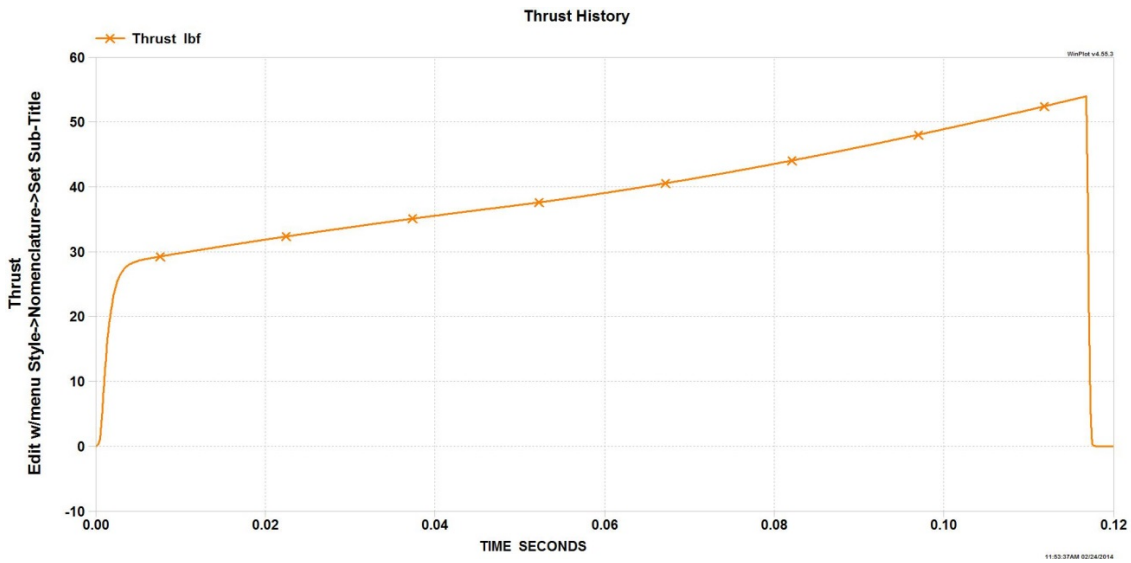


Figure 3.6.6 History of Thrust during Propellant Burn and Tail-off

4.0 References

1. A.K. Majumdar, A.C. LeClair, R. Moore, P.A. Schallhorn, "Generalized Fluid System Simulation Program, Version 6.0", NASA/TM—2013–217492, October 2013.
2. Eric W. Lemmon, Marcia L. Huber, Mark O. McLinden, "NIST Reference Fluid Thermodynamic and Transport Properties— REFPROP, Version 8.0, User's Guide, http://www.nist.gov/srd/upload/REFPROP8.PDF
3. Brennan, J. A., Brentari, E. G., Smith, R. V., and Steward, W. G., "Cooldown of Cryogenic Transfer Lines, An Experimental Report," National Bureau of Standards Report 9264, November 1966.
4. Cross, M.; Majumdar, A.; Bennett, J.; and Malla, R.: "Modeling of Chill Down in Cryogenic Transfer Lines," *J. Spacecraft Rockets*, Vol. 39, No. 2, pp. 284–289, 2002.
5. Lee, N.H. and Martin, C.S., "Experimental and Analytical Investigation of Entrapped Air in a Horizontal Pipe", Proc. 3rd ASME/JSME Joint Fluids Engg. Conference, ASME, NY, 1999, pp 1-8.
6. Bandyopadhyay, A. and Majumdar, A., "Network Flow Simulation of Fluid Transients in Rocket Propulsion System" submitted to *Journal of Propulsion and Power* for publication
7. David Chato & Rafael Sanabria, "Review and Test of Chilldown Methods for Space-Based Cryogenic Tanks", NASA Technical Memorandum 104458, AIAA-91-1843.
8. L.J. Hastings R.H. Flachbart, J.J. Martin, A. Hedayat, and M. Fazah, *Spray Bar Zero-Gravity Vent System for On-Orbit Liquid Hydrogen Storag*e, NASA/TM—2003–212926, October 2003.
9. Hastings, L. J., Hedayat, A., and Brown, T. M., *Analytical Modeling and Test Correlation of Variable Density Multilayer Insulation for Cryogenic Storage*, NASA/MSFC, NASA/TM-2004-213175, May 2004.
10. Rohsenow, W.M., Hartnett, J.P., and Cho, Y.I., *Handbook of Heat Transfer, 3rd Edition, pp.4.13*, McGraw-Hill, 1998.
11. Danielson, G. C. and Lanczos C., "Some improvements in practical Fourier analysis and their application to X-ray scattering from liquids", *Journal of Franklin Institute*, vol. 233, pp 365-380 and 435-452.

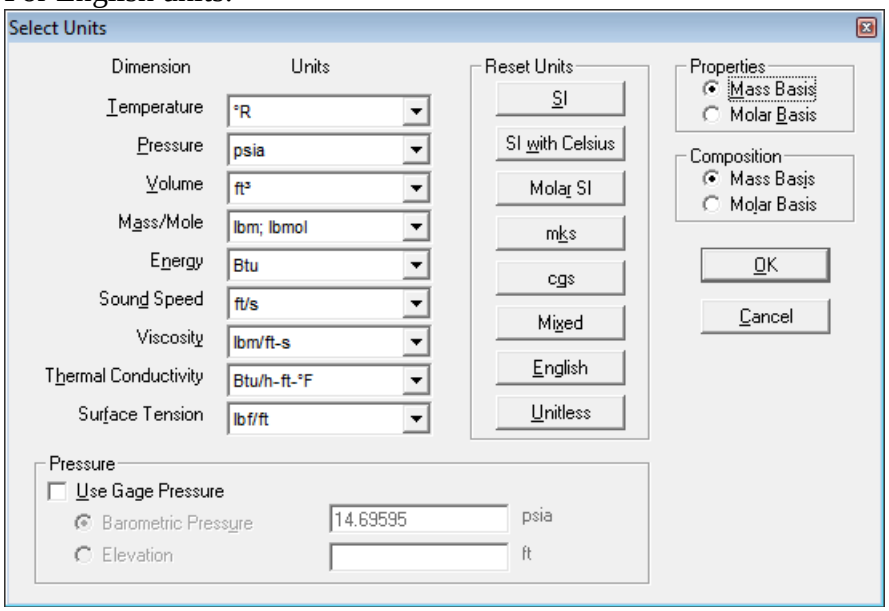
Appendix

A.1 Instructions to convert REFPROP data to GFSSP Property Table

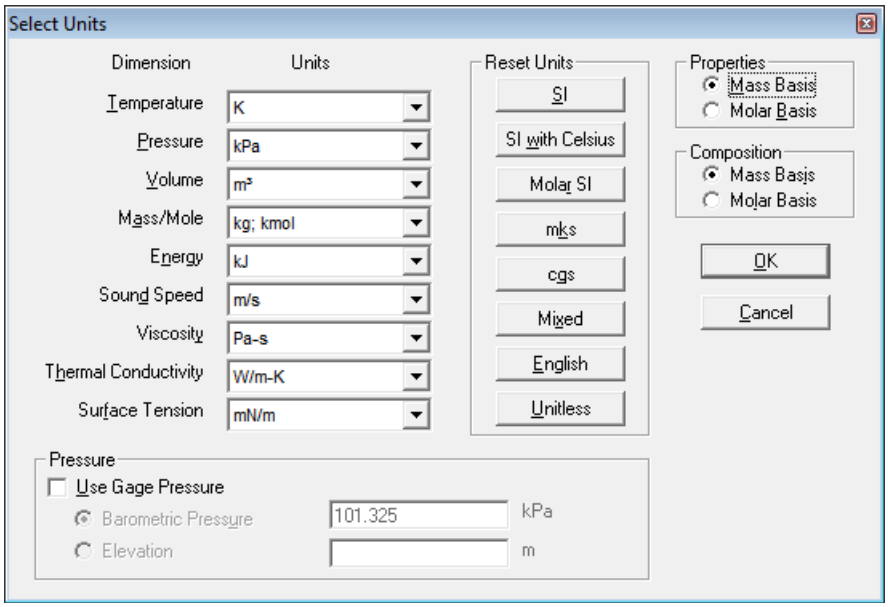
This document provides instructions for converting fluid property data from the REFPROP program into GFSSP's seven user-defined fluid files.

1.) In REFPROP, select OPTIONS/UNITS. (Note that REFPROP English units set Thermal Conductivity to BTU/h-ft-F; this will be converted to BTU/s-ft-F later.)

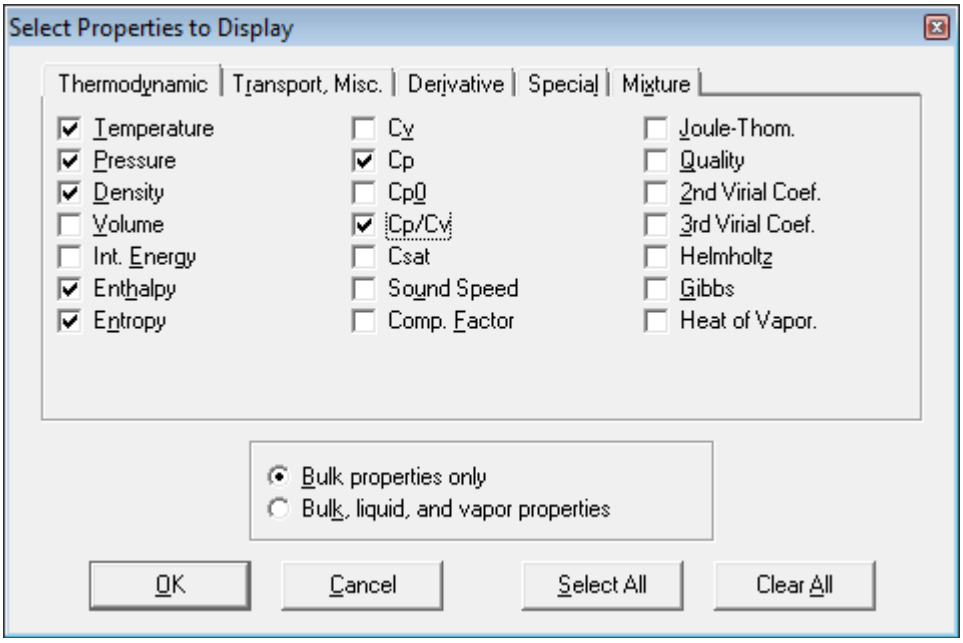
For English units:

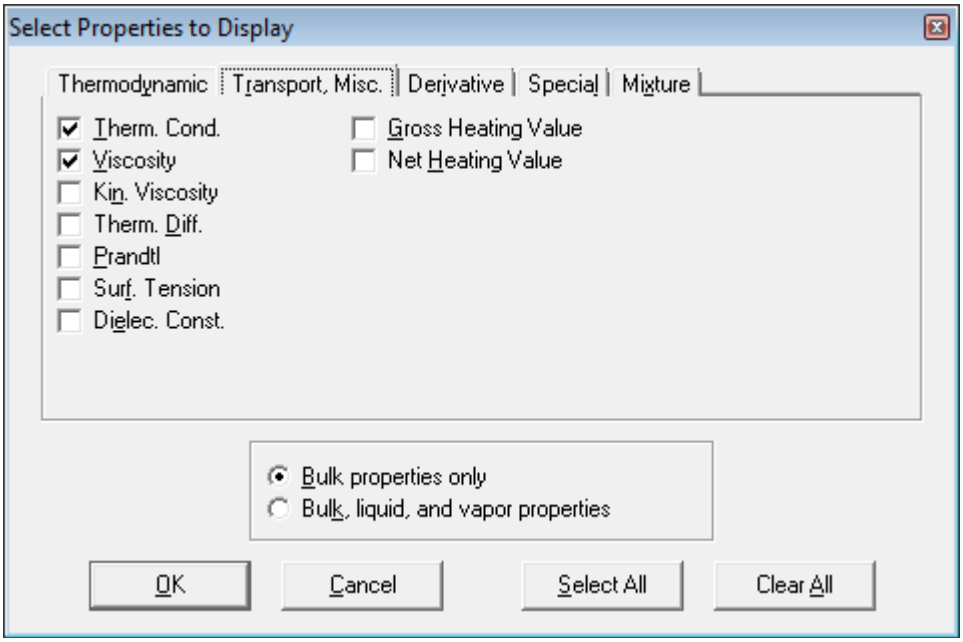


For SI units:

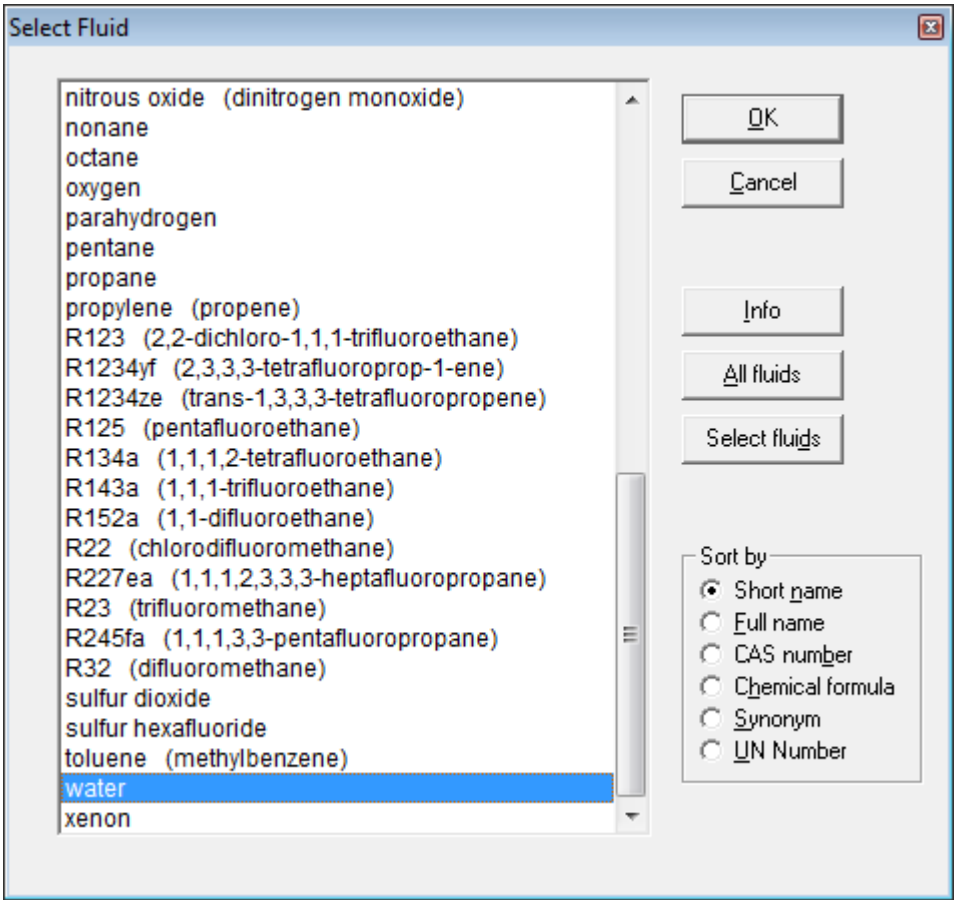


2.) Select OPTIONS/PROPERTIES and select Temperature, Pressure, Density, Enthalpy, Entropy, Cp, Cp/Cv, Thermal Conductivity, and Viscosity:

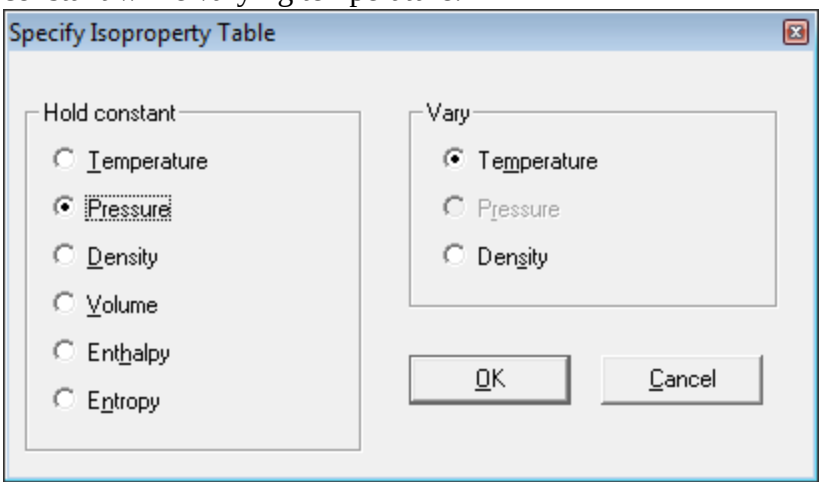




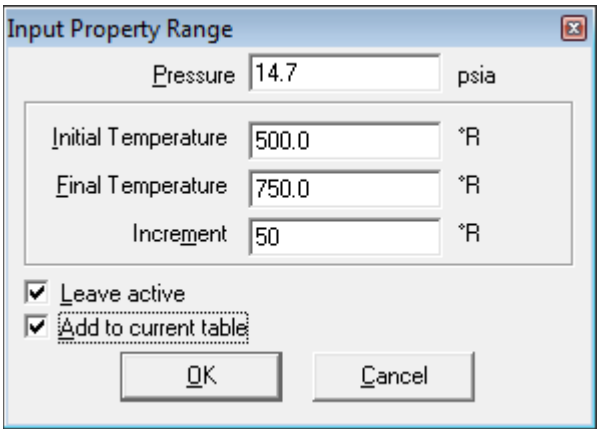
3.) Select SUBSTANCE/PURE FLUID and choose a fluid.



4.) Select CALCULATE/ISOPROPERTY TABLES, then choose to hold Pressure constant while varying temperature:



5.) Enter your first pressure and the temperature range and increment. Check the Leave Active and Add To Current Table boxes to make it easier to assemble a table at multiple pressures.



The example below shows water properties at three pressures and six temperatures (GFSSP's maximum is 301 pressures and 301 temperatures). Note carefully rows 5 and 6. At the given pressure of 14.7 psia, water is saturated at 671.64 °R (212 °F), and so REFPROP has added these extra rows. Examine your property data carefully for saturation properties.

1: water: p = 14.7 psia

	Temperature (°R)	Pressure (psia)	Density (lbm/ft³)	Enthalpy (Btu/lbm)	Entropy (Btu/lbm-°R)	Cp (Btu/lbm-°R)	Cp/Cv	Therm. Cond. (Btu/h-ft-°F)	Viscosity (lbm/ft-s)
1	500.00	14.700	62.426	8.4126	0.016879	1.0052	1.0000	0.32948	0.0010323
2	550.00	14.700	62.109	58.456	0.11228	0.99891	1.0176	0.35808	0.00050937
3	600.00	14.700	61.373	108.42	0.19922	1.0002	1.0527	0.37845	0.00031229
4	650.00	14.700	60.347	158.53	0.27944	1.0046	1.0972	0.38988	0.00021614
5	671.64	14.700	59.829	180.30	0.31238	1.0076	1.1187	0.39263	0.00018925
6	671.64	14.700	0.037320	1151.0	1.7577	0.49712	1.3369	0.014509	0.0000082192
7	700.00	14.700	0.035702	1165.0	1.7780	0.48507	1.3338	0.015144	0.0000086294
8	750.00	14.700	0.033204	1188.9	1.8111	0.47573	1.3281	0.016382	0.0000093633
1	500.00	114.70	62.447	8.7082	0.016877	1.0045	1.0000	0.32969	0.0010316
2	550.00	114.70	62.129	58.724	0.11222	0.99848	1.0177	0.35826	0.00050938
3	600.00	114.70	61.392	108.67	0.19914	0.99988	1.0527	0.37864	0.00031241
4	650.00	114.70	60.367	158.76	0.27932	1.0043	1.0971	0.39008	0.00021626
5	700.00	114.70	59.107	209.15	0.35401	1.0122	1.1488	0.39491	0.00016218
6	750.00	114.70	57.633	260.07	0.42426	1.0252	1.2075	0.39502	0.00012873
1	500.00	214.70	62.469	9.0036	0.016875	1.0038	1.0000	0.32989	0.0010309
2	550.00	214.70	62.148	58.993	0.11217	0.99804	1.0178	0.35844	0.00050939
3	600.00	214.70	61.411	108.92	0.19905	0.99952	1.0528	0.37883	0.00031252
4	650.00	214.70	60.386	158.99	0.27921	1.0039	1.0971	0.39029	0.00021639
5	700.00	214.70	59.128	209.37	0.35387	1.0118	1.1486	0.39515	0.00016230
6	750.00	214.70	57.657	260.26	0.42408	1.0247	1.2071	0.39528	0.00012885

Highlight any lines of saturated data. Then select EDIT/DELETE ROWS.

1: water: p = 14.7 psia

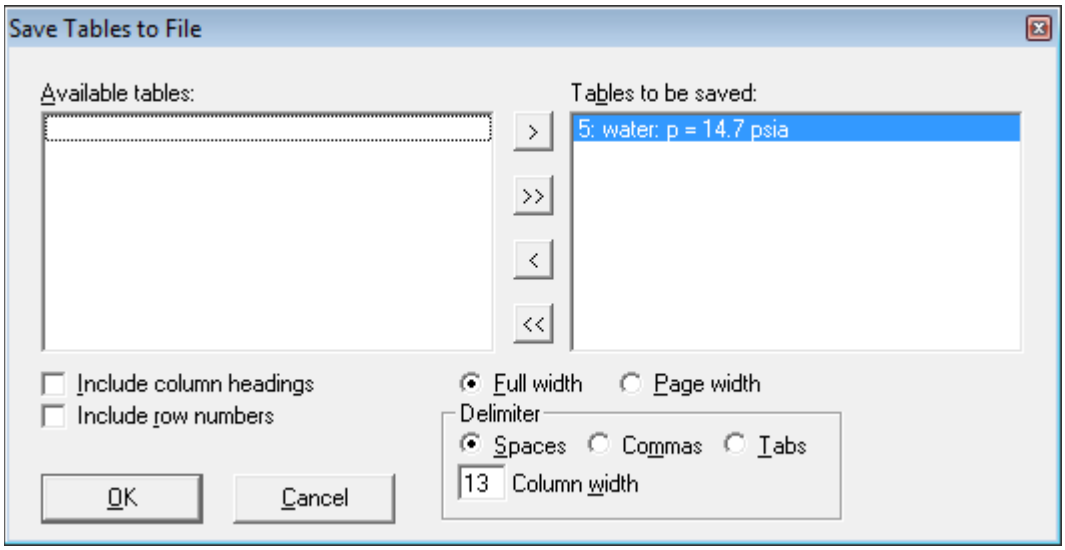
	Temperature (°R)	Pressure (psia)	Density (lbm/ft³)	Enthalpy (Btu/lbm)	Entropy (Btu/lbm-°R)	Cp (Btu/lbm-°R)	Cp/Cv	Therm. Cond. (Btu/h-ft-°F)	Viscosity (lbm/ft-s)
1	500.00	14.700	62.426	8.4126	0.016879	1.0052	1.0000	0.32948	0.0010323
2	550.00	14.700	62.109	58.456	0.11228	0.99891	1.0176	0.35808	0.00050937
3	600.00	14.700	61.373	108.42	0.19922	1.0002	1.0527	0.37845	0.00031229
4	650.00	14.700	60.347	158.53	0.27944	1.0046	1.0972	0.38988	0.00021614
5	671.64	14.700	59.829	180.30	0.31238	1.0076	1.1187	0.39263	0.00018925
6	671.64	14.700	0.037320	1151.0	1.7577	0.49712	1.3369	0.014509	0.0000082192
7	700.00	14.700	0.035702	1165.0	1.7780	0.48507	1.3338	0.015144	0.0000086294
8	750.00	14.700	0.033204	1188.9	1.8111	0.47573	1.3281	0.016382	0.0000093633
1	500.00	114.70	62.447	8.7082	0.016877	1.0045	1.0000	0.32969	0.0010316
2	550.00	114.70	62.129	58.724	0.11222	0.99848	1.0177	0.35826	0.00050938
3	600.00	114.70	61.392	108.67	0.19914	0.99988	1.0527	0.37864	0.00031241
4	650.00	114.70	60.367	158.76	0.27932	1.0043	1.0971	0.39008	0.00021626
5	700.00	114.70	59.107	209.15	0.35401	1.0122	1.1488	0.39491	0.00016218
6	750.00	114.70	57.633	260.07	0.42426	1.0252	1.2075	0.39502	0.00012873
1	500.00	214.70	62.469	9.0036	0.016875	1.0038	1.0000	0.32989	0.0010309
2	550.00	214.70	62.148	58.993	0.11217	0.99804	1.0178	0.35844	0.00050939
3	600.00	214.70	61.411	108.92	0.19905	0.99952	1.0528	0.37883	0.00031252
4	650.00	214.70	60.386	158.99	0.27921	1.0039	1.0971	0.39029	0.00021639
5	700.00	214.70	59.128	209.37	0.35387	1.0118	1.1486	0.39515	0.00016230
6	750.00	214.70	57.657	260.26	0.42408	1.0247	1.2071	0.39528	0.00012885

Your data should now look as below, with the same temperatures at each pressure.

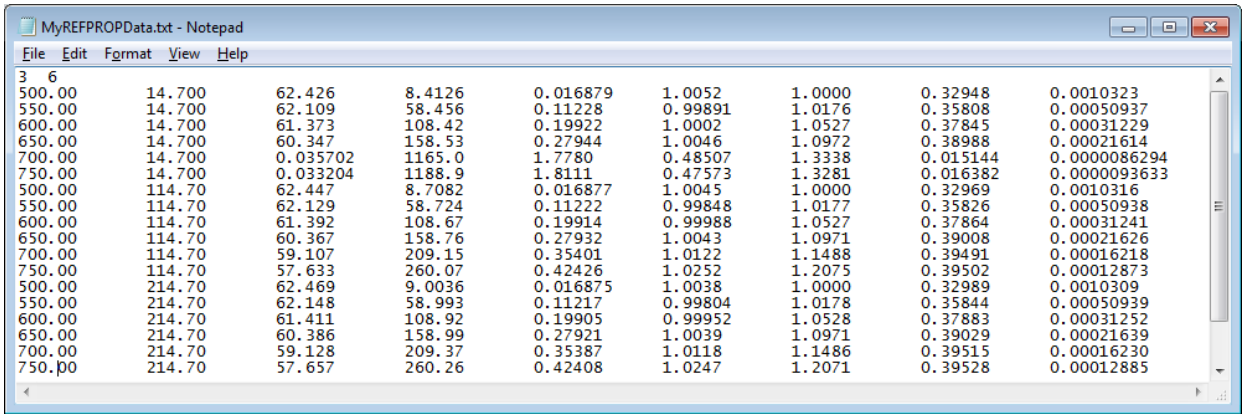
1: water: p = 14.7 psia

	Temperature (°R)	Pressure (psia)	Density (lbm/ft³)	Enthalpy (Btu/lbm)	Entropy (Btu/lbm-°R)	Cp (Btu/lbm-°R)	Cp/Cv	Therm. Cond. (Btu/h-ft-°F)	Viscosity (lbm/ft-s)
1	500.00	14.700	62.426	8.4126	0.016879	1.0052	1.0000	0.32948	0.0010323
2	550.00	14.700	62.109	58.456	0.11228	0.99891	1.0176	0.35808	0.00050937
3	600.00	14.700	61.373	108.42	0.19922	1.0002	1.0527	0.37845	0.00031229
4	650.00	14.700	60.347	158.53	0.27944	1.0046	1.0972	0.38988	0.00021614
7	700.00	14.700	0.035702	1165.0	1.7780	0.48507	1.3338	0.015144	0.0000086294
8	750.00	14.700	0.033204	1188.9	1.8111	0.47573	1.3281	0.016382	0.0000093633
1	500.00	114.70	62.447	8.7082	0.016877	1.0045	1.0000	0.32969	0.0010316
2	550.00	114.70	62.129	58.724	0.11222	0.99848	1.0177	0.35826	0.00050938
3	600.00	114.70	61.392	108.67	0.19914	0.99988	1.0527	0.37864	0.00031241
4	650.00	114.70	60.367	158.76	0.27932	1.0043	1.0971	0.39008	0.00021626
5	700.00	114.70	59.107	209.15	0.35401	1.0122	1.1488	0.39491	0.00016218
6	750.00	114.70	57.633	260.07	0.42426	1.0252	1.2075	0.39502	0.00012873
1	500.00	214.70	62.469	9.0036	0.016875	1.0038	1.0000	0.32989	0.0010309
2	550.00	214.70	62.148	58.993	0.11217	0.99804	1.0178	0.35844	0.00050939
3	600.00	214.70	61.411	108.92	0.19905	0.99952	1.0528	0.37883	0.00031252
4	650.00	214.70	60.386	158.99	0.27921	1.0039	1.0971	0.39029	0.00021639
5	700.00	214.70	59.128	209.37	0.35387	1.0118	1.1486	0.39515	0.00016230
6	750.00	214.70	57.657	260.26	0.42408	1.0247	1.2071	0.39528	0.00012885

6.) Select FILE/SAVE TABLES. Uncheck the Include Column Headings and Include Row Numbers boxes, and set the Full Width radio button.

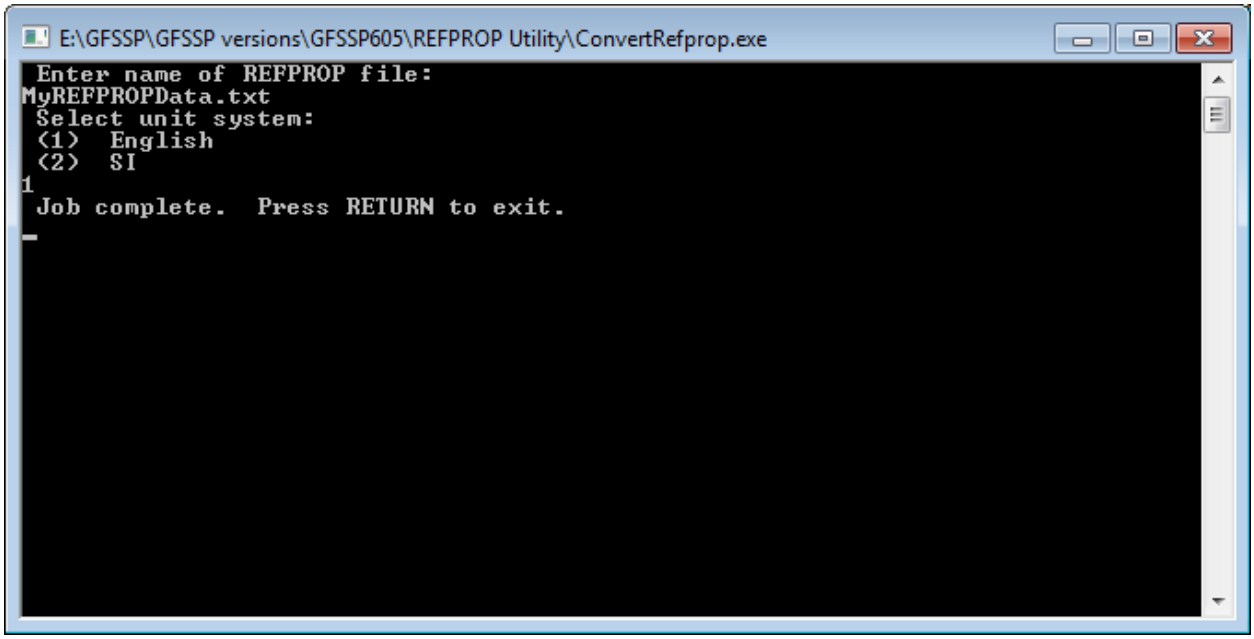


7.) Open the file you just saved with a text editor. Edit the file so that the first line gives the number of pressures and number of temperatures. In this example there are three pressures and six temperatures.




```
3   6
500.00      14.700      62.426      8.4126      0.016879    1.0052      1.0000      0.32948     0.0010323
550.00      14.700      62.109      58.456      0.11228     0.99891     1.0176      0.35808     0.00050937
600.00      14.700      61.373      108.42      0.19922     1.0002      1.0527      0.37845     0.00031229
650.00      14.700      60.347      158.53      0.27944     1.0046      1.0972      0.38988     0.00021614
700.00      14.700      0.035702    1165.0      1.7780      0.48507     1.3338      0.015144    0.0000086294
750.00      14.700      0.033204    1188.9      1.8111      0.47573     1.3281      0.016382    0.0000093633
500.00      114.70      62.447      8.7082      0.016877    1.0045      1.0000      0.32969     0.0010316
550.00      114.70      62.129      58.724      0.11222     0.99848     1.0177      0.35826     0.00050938
600.00      114.70      61.392      108.67      0.19914     0.99988     1.0527      0.37864     0.00031241
650.00      114.70      60.367      158.76      0.27932     1.0043      1.0971      0.39008     0.00021626
700.00      114.70      59.107      209.15      0.35401     1.0122      1.1488      0.39491     0.00016218
750.00      114.70      57.633      260.07      0.42426     1.0252      1.2075      0.39502     0.00012873
500.00      214.70      62.469      9.0036      0.016875    1.0038      1.0000      0.32989     0.0010309
550.00      214.70      62.148      58.993      0.11217     0.99804     1.0178      0.35844     0.00050939
600.00      214.70      61.411      108.92      0.19905     0.99952     1.0528      0.37883     0.00031252
650.00      214.70      60.386      158.99      0.27921     1.0039      1.0971      0.39029     0.00021639
700.00      214.70      59.128      209.37      0.35387     1.0118      1.1486      0.39515     0.00016230
750.00      214.70      57.657      260.26      0.42408     1.0247      1.2071      0.39528     0.00012885
```

8.) Copy the program ConvertRefprop.exe to your working directory (Windows will not allow it to write output to the GFSSP installation directory.). Run the program and enter the REFPROP data filename. Choose English or SI units (choosing English units converts the units of thermal conductivity from BTU/h-ft-F to BTU/s-ft-F). The program will produce the seven user-defined fluid data files in your working directory.

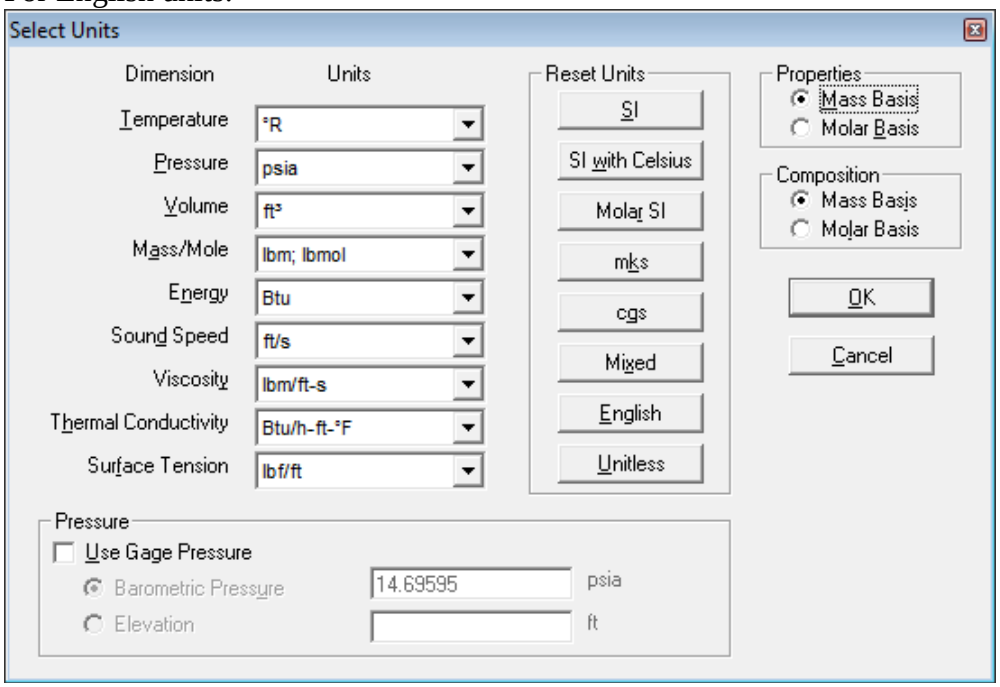


9.) Did you have to remove any lines of saturated data from the pressure/temperature range of your data? Do you expect your fluid will change phase in your GFSSP model? If so, you must also construct an eighth file of saturated properties. Instructions are given below, or consult the file *InstructionsToConvertREFPROP-DataToSaturatedFluidFile.pdf*, also in the utilities folder.

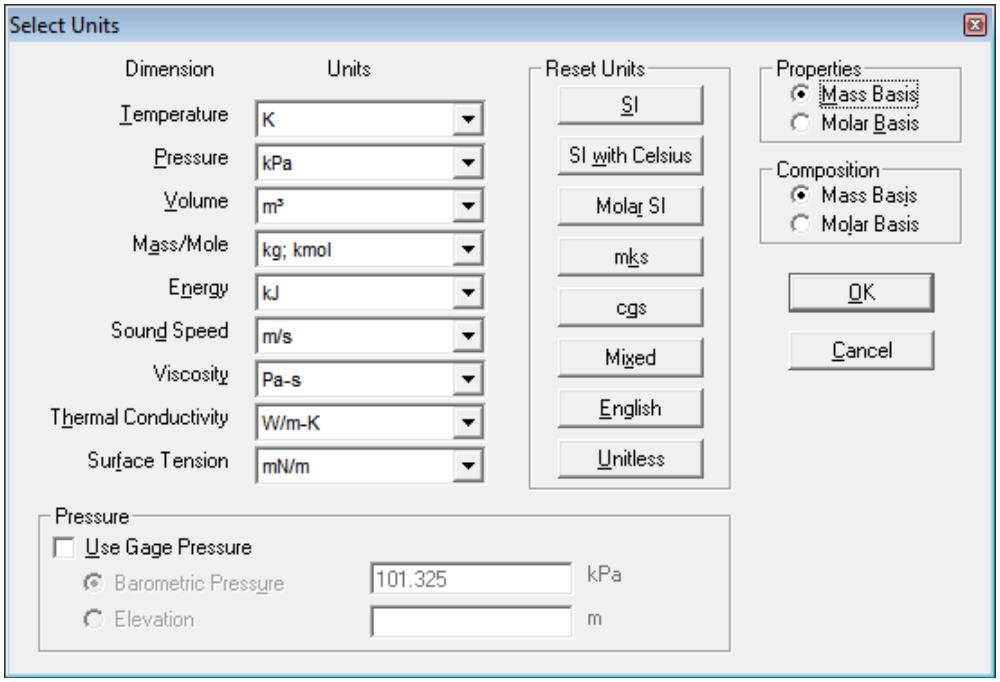
This document provides instructions for converting SATURATED fluid property data from the REFPROP program into GFSSP's saturated user-defined fluid input file.

1.) In REFPROP, select OPTIONS/UNITS. (Note that REFPROP English units set Thermal Conductivity to BTU/h-ft-F; this will be converted to BTU/s-ft-F later.)

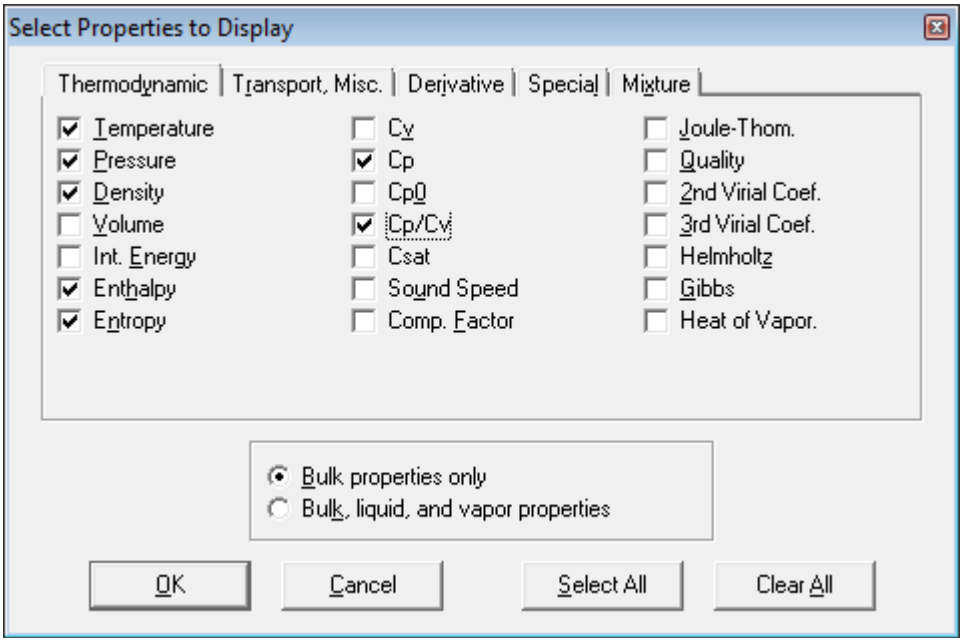
For English units:

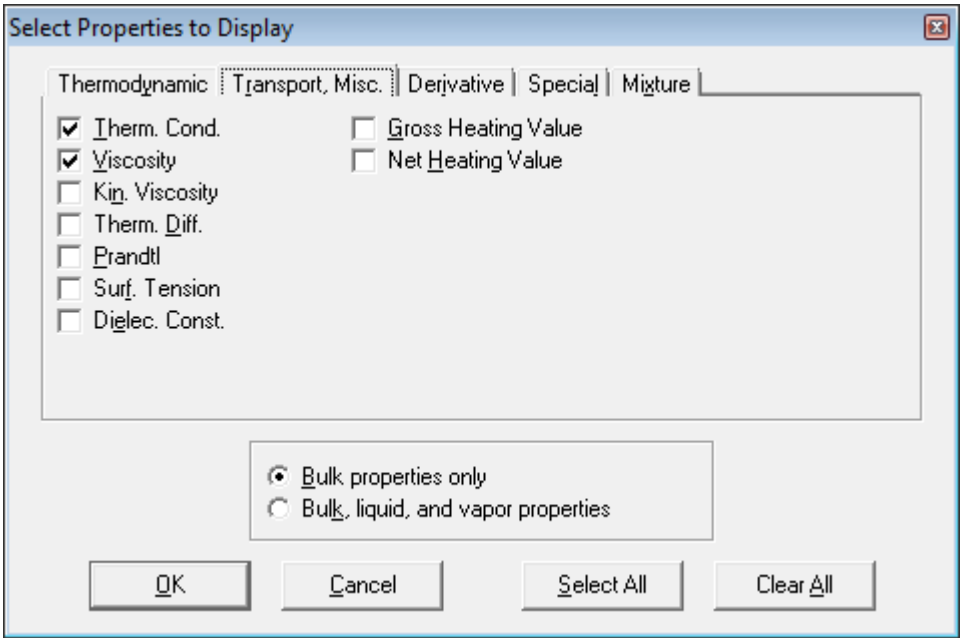


For SI units:

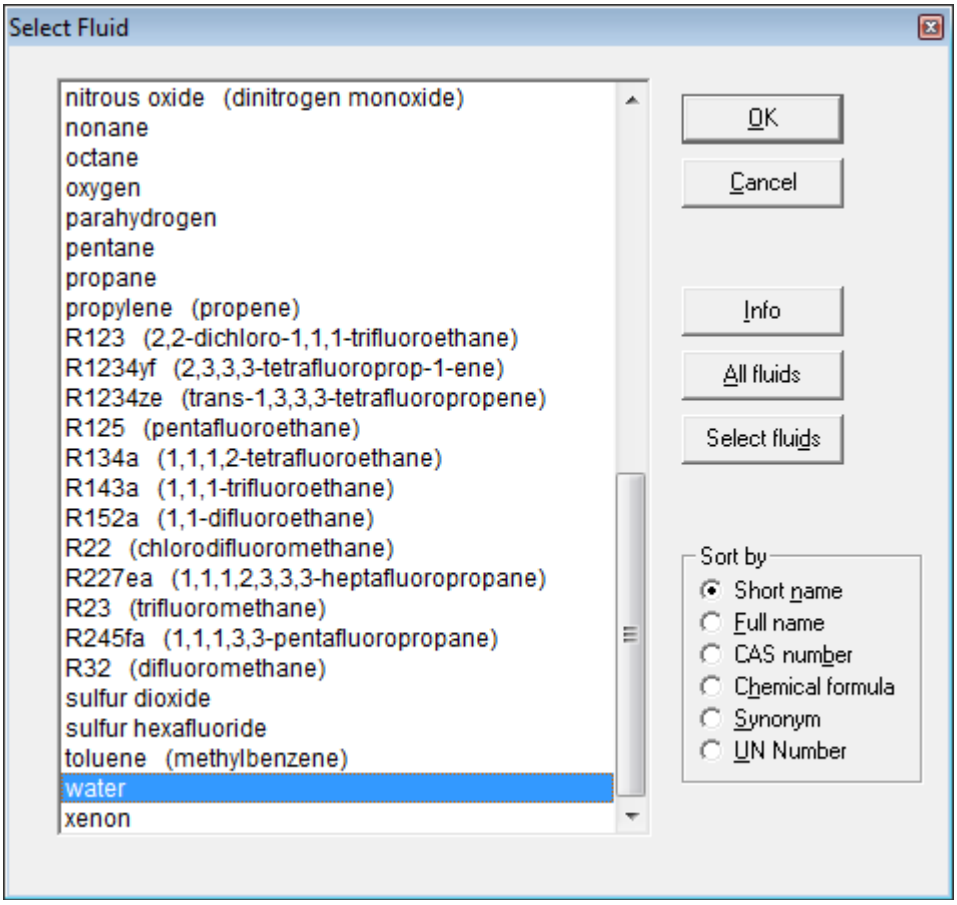


2.) Select OPTIONS/PROPERTIES and select Temperature, Pressure, Density, Enthalpy, Entropy, Cp, Cp/Cv, Thermal Conductivity, and Viscosity:

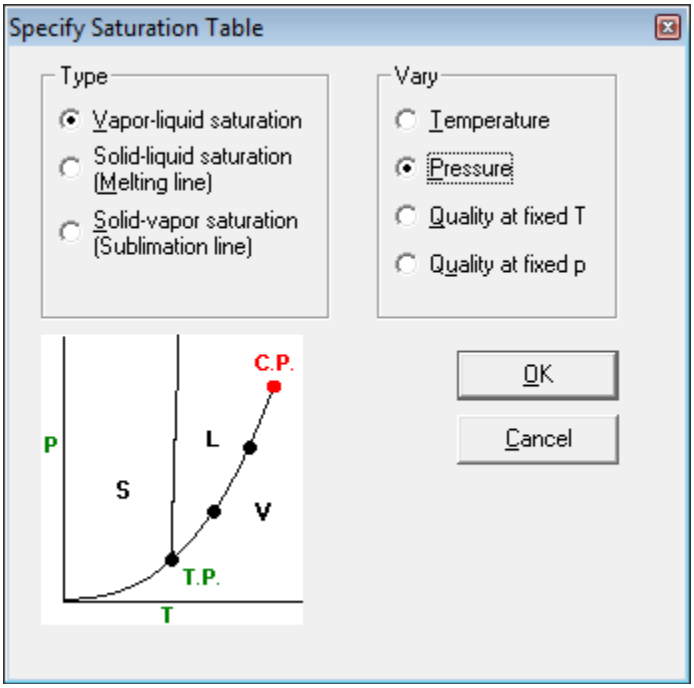




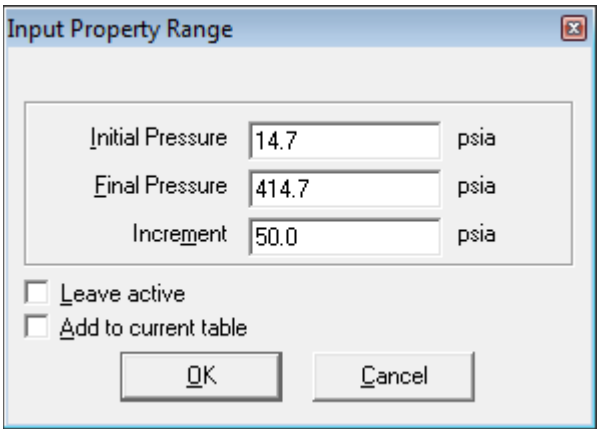
3.) Select SUBSTANCE/PURE FLUID and choose a fluid.



4.) Select CALCULATE/SATURATION TABLES, then choose to vary Pressure:



5.) Enter a pressure range and increment.

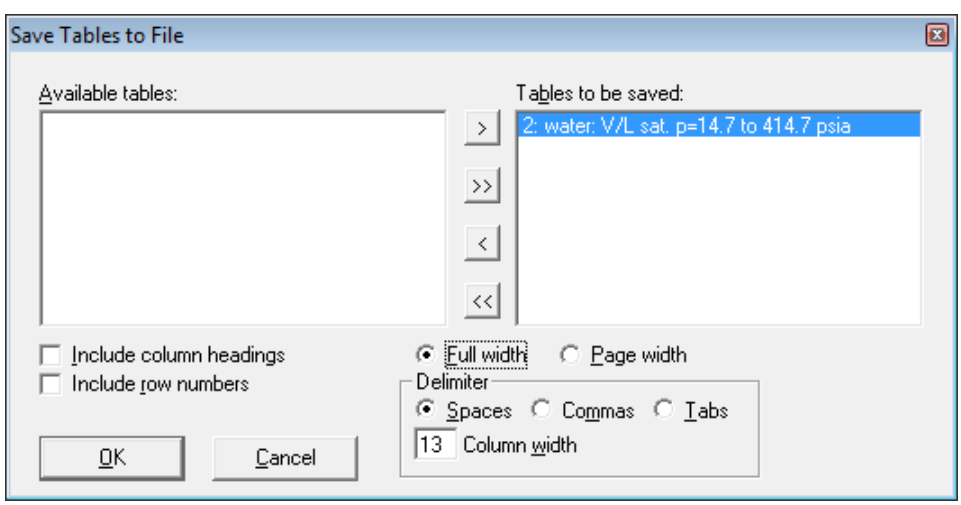


The example below shows saturated water properties at nine pressures.

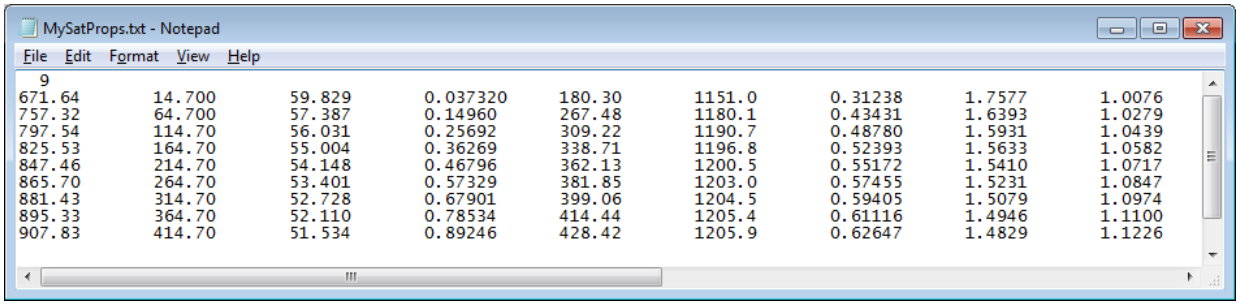
1: water: V/L sat. p=14.7 to 414.7 psia

	Temperature (°R)	Pressure (psia)	Liquid Density (lbm/ft³)	Vapor Density (lbm/ft³)	Liquid Enthalpy (Btu/lbm)	Vapor Enthalpy (Btu/lbm)	Liquid Entropy (Btu/lbm-°R)	Vapor Entropy (Btu/lbm-°R)	Liquid Cp (Btu/lbm-°R)	Vapor Cp (Btu/lbm-°R)	Liquid Cp/Cv	Vapor Cp/Cv	Liquid Therm. Cond. (Btu/h-ft-°F)	Vapor Therm. Cond. (Btu/h-ft-°F)	Liquid Viscosity (lbm/ft-s)	Vapor Viscosity (lbm/ft-s)
1	671.64	14.700	59.829	0.037320	180.30	1151.0	0.31238	1.7577	1.0076	0.49712	1.1187	1.3369	0.39263	0.014509	0.00018925	0.0000082192
2	757.32	64.700	57.387	0.14960	267.48	1180.1	0.43431	1.6393	1.0279	0.56711	1.2169	1.3658	0.39456	0.018060	0.00012487	0.0000093258
3	797.54	114.70	56.031	0.25692	309.22	1190.7	0.48780	1.5931	1.0439	0.61990	1.2710	1.3917	0.39147	0.020084	0.00010741	0.0000098392
4	825.53	164.70	55.004	0.36269	338.71	1196.8	0.52393	1.5633	1.0582	0.66520	1.3126	1.4167	0.38791	0.021633	0.000097902	0.000010195
5	847.46	214.70	54.148	0.46796	362.13	1200.5	0.55172	1.5410	1.0717	0.70608	1.3479	1.4413	0.38429	0.022934	0.000091561	0.000010474
6	865.70	264.70	53.401	0.57329	381.85	1203.0	0.57455	1.5231	1.0847	0.74413	1.3795	1.4659	0.38070	0.024077	0.000086879	0.000010707
7	881.43	314.70	52.728	0.67901	399.06	1204.5	0.59405	1.5079	1.0974	0.78031	1.4087	1.4906	0.37714	0.025114	0.000083201	0.000010909
8	895.33	364.70	52.110	0.78534	414.44	1205.4	0.61116	1.4946	1.1100	0.81526	1.4362	1.5155	0.37363	0.026072	0.000080190	0.000011089
9	907.83	414.70	51.534	0.89246	428.42	1205.9	0.62647	1.4829	1.1226	0.84945	1.4625	1.5409	0.37016	0.026972	0.000077648	0.000011254

6.) Select FILE/SAVE TABLES. Uncheck the Include Column Headings and Include Row Numbers boxes, and set the Full Width radio button.



7.) Open the file you just saved with a text editor. Edit the file so that the first line gives the number of saturation pressures. In this example there are nine pressures.

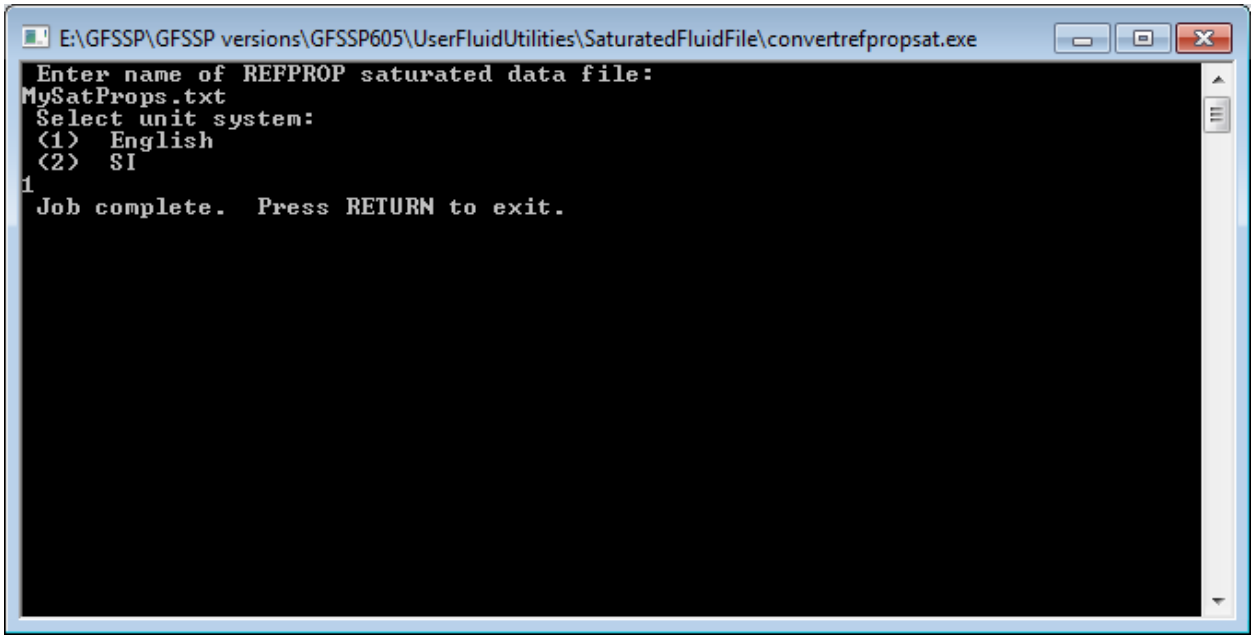


MySatProps.txt - Notepad

File Edit Format View Help

```
  9
671.64      14.700      59.829      0.037320    180.30      1151.0      0.31238     1.7577      1.0076
757.32      64.700      57.387      0.14960     267.48      1180.1      0.43431     1.6393      1.0279
797.54      114.70      56.031      0.25692     309.22      1190.7      0.48780     1.5931      1.0439
825.53      164.70      55.004      0.36269     338.71      1196.8      0.52393     1.5633      1.0582
847.46      214.70      54.148      0.46796     362.13      1200.5      0.55172     1.5410      1.0717
865.70      264.70      53.401      0.57329     381.85      1203.0      0.57455     1.5231      1.0847
881.43      314.70      52.728      0.67901     399.06      1204.5      0.59405     1.5079      1.0974
895.33      364.70      52.110      0.78534     414.44      1205.4      0.61116     1.4946      1.1100
907.83      414.70      51.534      0.89246     428.42      1205.9      0.62647     1.4829      1.1226
```

8.) Copy the program ConvertRefpropSat.exe to your working directory (Windows will not allow it to write output to the GFSSP installation directory.). Run the program and enter the REFPROP data filename. Choose English or SI units (choosing English units converts the units of thermal conductivity from BTU/h-ft-F to BTU/s-ft-F). The program will produce the saturated properties data file in your working directory.



E:\GFSSP\GFSSP versions\GFSSP605\UserFluidUtilities\SaturatedFluidFile\convertrefpropsat.exe

```
 Enter name of REFPROP saturated data file:
MySatProps.txt
 Select unit system:
 (1)  English
 (2)  SI
1
 Job complete.  Press RETURN to exit.
```

A.2 User Subroutine for Example 26

```
C*********************************************************************
      SUBROUTINE SORCEM(IPN,TERMU)
C     PURPOSE: ADD MASS SOURCES
C     IPN - GFSSP INDEX NUMBER FOR NODE
C     TERMU - UNSTEADY TERM IN MASS CONSERVATION EQUATION
C*********************************************************************
       INCLUDE 'comblk.for'
C*******************************************************************
C     ADD CODE HERE
                      EQUIVALENCE (VOLAIRM, USRVAR1(1) )
                      EQUIVALENCE (VOLAIR, USRVAR1(2) )

                      EQUIVALENCE (PGASM, USRVAR1(3))
                      EQUIVALENCE (PGAS, USRVAR1(4) )
                      EQUIVALENCE (DVOLUME, USRVAR1(7))


                      DATA alpha_g/0.448/

                      DATA RAIR, GAMAAIR/53.3, 1.4/

                      DATA RELAXVOL,RELAXPGAS/1,1/
                      DATA RELAXV/0.8/
                      DATA p0air, T0air/14.7, 60/
                      DATA xWater/20.0/



c Initial property calculations
c Temperature in Rankine

                      t0airR = t0air + 459.67  ! converting into R
c Density: (using ideal gas law)
                      rhoair0 = 144.0 * p0air /(RAIR * t0airR) ! in lbmass/cuft
c Air col length:
                      xair = alpha_g * xWater/(1.0 - alpha_g)


c Initial volume and mass of air entrapped in the pipe:

                      DPipe = 1.025/12.0     ! in ft
                      volAir0 = 3.1415927 * DPipe*DPipe*xair/4.0

                      AirMass0 = volAir0 * rhoAir0         ! initial air mass

                      EMGAS = AirMass0



                      NUMBER = 12                   ! Last Node

                      CALL INDEXI(NUMBER,NODE,NNODES,IPN)


                      IF(ISTEP.EQ.1) then
                      PGAS = P(IPN)
                      PGASM = PM(IPN)
                      vwat0 = volume(ipn)
                      vtot0 = vwat0 + volair0
                      endif

                      VOLOLD = VOLUME(IPN)

C        Using real gas law (for Water)
```

```
c                     Z = compressibility factor
c                     EM = mass
c                     RNODE = gas constant for the gas
c                     TF = fluid temp, is it automatically in Rankin?
c                     PGAS = air pressure
c                     VOLNEW = volume at current time step, computed using the thermo
c                     dynamic relations
c                     DVOLUME = change in volume of node 12 from the previous time step

c                     VOLNEW = Z(IPN) * EM(IPN) * RNODE(IPN)*TF(IPN)/PGAS

                      FACTVOL = AirMass0*RAIR/(Z(IPN)*EM(IPN)*RNODE(IPN))

c
                      VOLNEW = vtot0/(1.0+FACTVOL)

                      VOLUME(IPN) = (1.-RELAXV)*VOLOLD + RELAXV*VOLNEW

                      DVOLUME = VOLUME(IPN) - VOLUMEM(IPN)

c     air volume change (DVOLAIR) is same in mag but opposite sign than
c                     water volume.

                      DVOLAIR = - DVOLUME


                      IF(ISTEP.EQ.1) then


                      VOLAIR = volair0
                      ELSE
                      VOLAIROLD = VOLAIR
c                     volairm is being updated in BNDUSER

                      VOLAIRNEW = VOLAIRM + DVOLAIR

                      VOLAIR = (1-RELAXVOL)*VOLAIROLD + RELAXVOL * VOLAIRNEW

                      ENDIF

                      IF (VOLAIR .GT. 0.0) THEN
c                      Pressure and Temperature Calculation
                        PGAS = EMGAS*RAIR*TF(IPN)/(Vtot0-VOLUME(IPN))
                        TGAS = PGAS * (Vtot0-VOLUME(IPN))/(EMGAS * RAIR)
                      ELSE
                        PRINT *, " ***AIR VOLUME IS 0 or NEGATIVE, and = ", VOLAIR
                      ENDIF

                      PGASPSI = PGAS/144.0
                      TGASF = TGAS - 460
                      USRPVAR(2) = TGASF
                      USRPVARNAME(2) = 'TGAS'
                      USRPVARUNIT(2) = 'F'
                      USRPVAR(1) = PGASPSI
                      USRPVARNAME(1) = 'PGAS'
                      USRPVARUNIT(1) = 'psi'
      RETURN
      END
C***********************************************************************
      SUBROUTINE SORCEF(I,TERM0,TERM1,TERM2,TERM3,TERM4,TERM5,TERM6,
     &                  TERM7,TERM8,TERM9,TERM10,TERM100)
C     PURPOSE: ADD MOMENTUM SOURCES (LBF)
C     I - GFSSP INDEX NUMBER FOR BRANCH
C     TERM0 - UNSTEADY TERM IN MOMENTUM CONSERVATION EQUATION
C     TERM1 - LONGITUDINAL INERTIA
C     TERM2 - PRESSURE GRADIENT
C     TERM3 - GRAVITY FORCE
```

```
C     TERM4 - FRICTION FORCE
C     TERM5 - CENTRIFUGAL FORCE
C     TERM6 - EXTERNAL MOMETUM SOURCE DUE TO PUMP
C     TERM7 - MOMENTUM SOURCE DUE TO TRANSVERSE FLOW(MULTI-DIMENSIONAL MODEL)
C     TERM8 - MOMENTUM SOURCE DUE TO SHEAR(MULTI-DIMENSIONAL MODEL)
C     TERM9 - VARIABLE GEOMETRY UNSTEADY TERM
C     TERM10 - NORMAL STRESS
C     TERM100 - USER SUPPLIED MOMENTUM SOURCE
C*********************************************************************
       INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE

      IF(IBRANCH(I).EQ.1112)then
        NUMBER = 12
        CALL INDEXI(NUMBER,NODE,NNODES,IPN)
c Momentum Source term for the Last Branch
        TERM100 = -RHO(IPN)*(VOLUME(IPN)-VOLUMEM(IPN))*VEL(I)/(GC*DTAU)

          ELSE                ! For all other branches

           TERM100 = 0.0
      ENDIF

      RETURN
      END
C*********************************************************************
      SUBROUTINE BNDUSER
C     PURPOSE: MODIFY BOUNDARY CONDITIONS
C*********************************************************************
      INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE
                       EQUIVALENCE (VOLAIRM, USRVAR1(1) )
                       EQUIVALENCE (VOLAIR,  USRVAR1(2) )

                       EQUIVALENCE (PGASM,    USRVAR1(3))
                       EQUIVALENCE (PGAS, USRVAR1(4) )
                       EQUIVALENCE (DVOLUME, USRVAR1(7))
c                      print *, "TAU = " , tau
c                      print *, PGAsM, PGAS, VOLAIRM, VOLAIR, DVOLUME

                       VOLAIRM = VOLAIR
                       PGASM = PGAS
c        print *, 'pasm = ', pgasm, 'pgas = ', pgas

      RETURN
      END
```

A.3 User Subroutine for Example 28

```
C*********************************************************************
      SUBROUTINE USRHCF(NUMBER,HCF)
C     PURPOSE: PROVIDE HEAT TRANSFER COEFFICIENT
C*********************************************************************
      INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE
      DATA TANKDIA/7.25/
      DATA CON1,FN1/0.27,0.50/
      DATA FACTHC/1.0/
C     OBTAIN INDICES OF THE FLUID NODE
      NUM2=ICF(NUMBER)
      CALL INDEXI(NUM2,NODE,NNODES,IPN)
      NUM1=ICS(NUMBER)
```

```
      CALL INDEXS(NUM1,NODESL,NSOLIDX,IPSN)

C     COMPUTE HEAT TRANSFER BETWEEN WALL AND PROPELLANT
       BETA=1./TF(IPN)
       DELTAT=ABS(TS(IPSN)-TF(IPN))
       IF(DELTAT .NE. 0.0) THEN
        GR=TANKDIA**3*RHO(IPN)**2*G*BETA*DELTAT/(EMU(IPN)**2)
        PRNDTL=CPNODE(IPN)*EMU(IPN)/CONDF(IPN)
        HCF=FACTHC*CON1*CONDF(IPN)*(GR*PRNDTL)**FN1/TANKDIA
       ENDIF
      RETURN
      END
```

A.4 User Subroutine for Example 29

```
C*********************************************************************
      SUBROUTINE SORCEM(IPN,TERMU)
C     PURPOSE: ADD MASS SOURCES
C     IPN - GFSSP INDEX NUMBER FOR NODE
C     TERMU - UNSTEADY TERM IN MASS CONSERVATION EQUATION
C*********************************************************************
       INCLUDE 'comblk.for'
C*******************************************************************
C     ADD CODE HERE
      DATA TIL,HFGLH2/38.07,191.30/
      DATA HAREA,HL/ 78.5,5.0/
      DATA C1, C2 /0.1, 0.25/
      NUMUL=2
      NUMPRP = 4
      CALL INDEXI(NUMUL,NODE,NNODES,IPUL)
      CALL INDEXI(NUMPRP,NODE,NNODES,IPPRP)
C     ESTIMATE MASS TRANSFER FROM PROPELLANT TO ULLAGE

C     CALCULATE ULLAGE TO INTERFACE HEAT TRANSFER COEFFICIENT
c      BETA = 1.0 / TF(IPUL)
c      DELTAT = ABS(TF(IPUL) - TIL)
c      GR = HL**3 * RHO(IPUL)**2 * G * BETA * DELTAT / (EMU(IPUL)**2)
c      PRNDTL = CPNODE(IPUL) * EMU(IPUL) / CONDF(IPUL)
c      XNU = C1 * (GR * PRNDTL)**C2
      HUL =  CONDF(IPUL) / HL
      HLP = CONDF(IPPRP) / HL
      QDOTUL = HUL*HAREA*(TF(IPUL)-TIL)
      QDOTLP = HLP*HAREA*(TIL-TF(IPPRP))
      EMDOTGH2 = (QDOTUL-QDOTLP)/HFGLH2
      IF (NODE(IPN).EQ. 2) EMS(IPN) = EMDOTGH2
      IF (NODE(IPN).EQ. 4) EMS(IPN) = -EMDOTGH2

C     EXTRACT MASS FROM LIQUID NODE

      CALL INDEXI(164,IBRANCH,NBR,IB164)
      CALL INDEXI(4,NODE,NNODES,IP2)
      SORCEMAS(IP2) = -FLOWR(IB164)


c      WRITE(*,100) TAU,TF(IPUL),TF(IPPRP),QDOTUL,QDOTLP,EMDOTGH2

c100   FORMAT (6E12.3)


      RETURN
      END
C*********************************************************************
      SUBROUTINE SORCETS(IPSN,TERMD)
C     PURPOSE: ADD SOURCE TERM IN SOLID TEMPERATURE EQUATION
C*********************************************************************
```

```
      INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE
C***  COMMON BLOCK FOR MLI SUBROUTINE
      COMMON/CMLI/FNSTAR(10),FNLAYER(10),QFLUX(10),SAREA(10),
     &       TC,TH,PTORR,FMLIEMISS,SHRDEMISS,CR,CS,CG,DF

C    MLI Layer Density per section (layer/cm)
      DATA FNSTAR(1),FNSTAR(2),FNSTAR(3)/8,12,16/

C    Number of layers per section
      DATA FNLAYER(1),FNLAYER(2),FNLAYER(3)/10,15,20/

      DATA SAREA(1),SAREA(2),SAREA(3),SAREA(4)/4*1.0/
      DATA PTORR/5.0E-6/
      DATA FMLIEMISS/0.031/
      DATA SHRDEMISS/0.04/

      DATA THR/522./
C*****Determine with nodes are to be MLI nodes
      DIMENSION MLINODE(6)
      DATA (MLINODE(I),I = 1,6)/7,6,12,13,14,15/

      DF=2.9
      CS=2.4E-4
      CR=4.944E-10
      CG=14600.0

      NLAYER = 3

C     DEFINE SOLID NODE NUMBERS CONNECTED TO MLI


      TH = THR/1.8

      DO I = 1,6
        IF(NODESL(IPSN).EQ.MLINODE(I)) THEN
          TC = TS(IPSN)/1.8
          CALL MLI_HEAT_RATE(NLAYER,QAVG)
C     CONVERT THE HEAT RATE FROM WATT/MT**2 TO BTU/SEC-FT**2
          QAVGBTUSEC = QAVG*0.0009486608/10.7631
C     GET THE CONDUCTOR AREA
          NUMBER = NAMESF(IPSN,1)
          CALL INDEXSFC(NUMBER,ICONSF,NSFC,ICSF)
          SHSORC(IPSN) = QAVGBTUSEC*ARSF(ICSF)
        ENDIF
      ENDDO
      RETURN
      END
C**********************************************************************
      SUBROUTINE BNDUSER
C     PURPOSE: MODIFY BOUNDARY CONDITIONS
C**********************************************************************
      INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE
C     PLOT MLI HEAT LEAK
      DIMENSION MLINODE(6)
      DATA (MLINODE(I),I = 1,6)/7,6,12,13,14,15/

C     UPDATE PRESSURE OF THE PSEUDO-BOUNDARY NODE

      DATA TIL,HFGLH2/38.07,191.30/
      DATA HAREA/ 27.882/
      PULMAX = 20*144
      PULMIN = 19*144
      OPENARU = 0.003526
```

```
      OPENARL = 0.003526
      CLAREA = 1.E-16
      NUMUL=2
      NUMPSN = 3
      NUMPRP = 4
      USRVAR = .TRUE.
      USRVARSNUM=7
      USRPVARNAME(1)='QMLI7'
      USRPVARNAME(2)='QMLI6'
      USRPVARNAME(3)='QMLI12'
      USRPVARNAME(4)='QMLI13'
      USRPVARNAME(5)='QMLI14'
      USRPVARNAME(6)='QMLI15'
      USRPVARNAME(7)='SQMLI'

      USRPVARUNIT(1)='BTU/SEC'
      USRPVARUNIT(2)='BTU/SEC'
      USRPVARUNIT(3)='BTU/SEC'
      USRPVARUNIT(4)='BTU/SEC'
      USRPVARUNIT(5)='BTU/SEC'
      USRPVARUNIT(6)='BTU/SEC'
      USRPVARUNIT(7)='BTU/SEC'

      SUMQMLI = 0.0
      DO I = 1,6
        NUMBER = MLINODE(I)
        CALL INDEXS(NUMBER,NODESL,NSOLIDX,IPSN)
        USRPVAR(I) = SHSORC(IPSN)
        SUMQMLI = SUMQMLI + SHSORC(IPSN)
      ENDDO

      USRPVAR(7) = SUMQMLI

      CALL INDEXI(NUMUL,NODE,NNODES,IPUL)
      CALL INDEXI(NUMPSN,NODE,NNODES,IPPSN)
      CALL INDEXI(NUMPRP,NODE,NNODES,IPPRP)
      P(IPPSN) = P(IPUL)
      IF (ISTEP.EQ.1) TANKVOL = VOLUME(IPUL)+VOLUME(IPPRP)
C     CALCULATE PROPELLANT AND ULLAGE VOLUME
      VOLUME(IPPRP) = EM(IPPRP)*Z(IPPRP)*RNODE(IPPRP)*TF(IPPRP)/P(IPPRP)
      VOLUME(IPUL) = TANKVOL-VOLUME(IPPRP)

C     TVS VALVE OPENING & CLOSING SEQUENCE
C     ULLAGE BRANCHES (162,168,169,1610,1611)

      CALL INDEXI(162,IBRANCH,NBR,IB162)
      CALL INDEXI(168,IBRANCH,NBR,IB168)
      CALL INDEXI(169,IBRANCH,NBR,IB169)
      CALL INDEXI(1610,IBRANCH,NBR,IB1610)
      CALL INDEXI(1611,IBRANCH,NBR,IB1611)

C     NODE 2 IS PRESSURE MONITORING NODE
      CALL INDEXI(2,NODE,NNODES,IP2)
      IF (P(IP2).GE.PULMAX) THEN
        AREA(IB162) = OPENARU
        AREA(IB168) = OPENARU
        AREA(IB169) = OPENARU
        AREA(IB1610) = OPENARU
        AREA(IB1611) = OPENARU
      ENDIF
      IF (P(IP2).LE.PULMIN) THEN
        AREA(IB162) = CLAREA
        AREA(IB168) = CLAREA
        AREA(IB169) = CLAREA
        AREA(IB1610) = CLAREA
        AREA(IB1611) = CLAREA
      ENDIF
```

```
C     LIQUID
      NUMBER = 164
      CALL INDEXI(NUMBER,IBRANCH,NBR,IB164)
C     NODE 2 IS PRESSURE MONITORING NODE
      IF (P(IP2).GE.PULMAX) AREA(IB164) = OPENARL
      IF (P(IP2).LE.PULMIN) AREA(IB164) = CLAREA

C     PRESSURE AT BOUNDARY NODE 16 IS FIXED TO BE 0.1 PSI
C     HIGHER THAN PRESSURE AT INTERNAL NODE 2.

      CALL INDEXI(2, NODE, NNODES, IPN2)
      CALL INDEXI(16, NODE, NNODES, IPN16)
      P(IPN16) = (P(IPN2) / 144.0 + 0.268) * 144


      RETURN
      END
C*********************************************************************
      SUBROUTINE USRHCF(NUMBER,HCF)
C     PURPOSE: PROVIDE HEAT TRANSFER COEFFICIENT
C**********************************************************************
      INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE
      DATA HL /5.0/


c      return ! heat transfer coefficient specified in VTASC

C     IF(ICONSF(NUMBER).NE.62) RETURN

      NUMF = ICF(NUMBER)
      CALL INDEXI(NUMF, NODE, NNODES, IPN)
      NUMS = ICS(NUMBER)
      CALL INDEXS(NUMS, NODESL, NSOLIDX, IPSN)

      BETA = 1.0 / TF(IPN)
      DELTAT = ABS(TF(IPN) - TS(IPSN))
      GR = HL**3 * RHO(IPN)**2 * G * BETA * DELTAT / (EMU(IPN)**2)
      PRNDTL = CPNODE(IPN) * EMU(IPN) / CONDF(IPN)
      RA = GR*PRNDTL
      CVT = (0.13*PRNDTL**0.22)/(1.0+0.61*PRNDTL**0.81)**0.42
      CLBAR = 0.671/(1+(0.492/PRNDTL)**(9.0/16.0))**(4.0/9.0)
      ANUUT = CLBAR*RA**0.25
      ANUL = 2.0/LOG(1.0+2.0/ANUUT)
      ANUT = CVT*RA**0.33/(1.0+1.4E09*PRNDTL/RA)
      ANU = (ANUL**6+ANUT**6)**(1.0/6.0)
      HCF = 10 * ANU * CONDF(IPN) / HL

      RETURN
      END
!*************************************************************************
!
      SUBROUTINE MLI_HEAT_RATE(NLAYER,QAVG)
!
!  PURPOSE:  Determine MLI heat leak of variable densitty MLI with the Modified Lockeed
Equation
!
!*************************************************************************
        DIMENSION RMLI(10), MLICORR(10), PDMLI(10,10),TMLI(10),
     &            TMLICORR(10), FTMLICORR(10)
!=================================================================================
      COMMON/CMLI/FNSTAR(10),FNLAYER(10),QFLUX(10),SAREA(10),
     &       TC,TH,PTORR,FMLIEMISS,SHRDEMISS,CR,CS,CG,DF
        LOGICAL PRINTI
        OPEN (14,FILE = 'MLI_OUT',STATUS = 'UNKNOWN')
```

```
        DATA CCMULT/0.1/
        DATA RELAXMLI/0.5/
        DATA CCMLI/1.0E-6/
        DATA ITMAX/500/
        DATA PRINTI/.TRUE./
!       DATA DF/1.0/
       F=(TH-TC)/4
      TMLI(1)=TC+F
      TMLI(2)=TMLI(1)+F
      TMLI(3)=TH-F




! Define number of equations
      NVARMLI= NLAYER

!Start interation counter
      ITERMLI= 0

!Call subroutines to calculate values of residuals, partial derivatives and changes in
values of varialbes

30    CAll MLIEQNS(RMLI,TMLI,NVARMLI)

      ITERMLI= ITERMLI + 1

      CALL MLICOEF(RMLI, PDMLI,TMLI,NVARMLI)

      CALL GAUSSY(PDMLI, RMLI, TMLICORR, NVARMLI, MESSAGE)

      IF (MESSAGE .EQ. 1) THEN
        WRITE (14,*) ITERMLI
      END IF

C   Correcting values of the variables

      DO L=1, NVARMLI
        TMLI(L)= TMLI(L)- TMLICORR(L)* RELAXMLI
      END DO

C    CAlculate Fractional Change

      DIFMAX=0
      IF (ITERMLI .GT. 20)  RELAXMLI = 0.9

      DO I= 1, NVARMLI

       IF (TMLI(I) .GT. 1.E-6) THEN
        FTMLICORR(I)= ABS(TMLICORR(I)/ TMLI(I))
        DIFMAX= MAX(FTMLICORR(I), DIFMAX)!sdd
       END IF
      END DO

       IF (DIFMAX .GT. CCMLI .AND. ITERMLI .LT. ITMAX) GO TO 30
       IF (DIFMAX .GT. CCMLI .AND. ITERMLI .GT. ITMAX) THEN
       WRITE (*,*) 'MLI equation did not converge -- DIFMAX=', DIFMAX

      END IF

      SUMQFLX = 0.0
      NLAYERP1 = NLAYER+1
      DO I = 1, NLAYER+1
        IF (I.EQ.1) CALL QFLUXMLI(TMLI(1),TC,CR,CS,CG,FNSTAR(1),
     &                          FNLAYER(1),FMLIEMISS,PTORR,DF,QFLUX(1))
       IF (I.EQ.NLAYERP1) CALL QFLUXRAD(TH,TMLI(NVARMLI),FMLIEMISS,
     &                                  SHRDEMISS,QFLUX(NLAYERP1))
```

```
       IF (I.NE.1.AND.I.NE.NLAYERP1) CALL QFLUXMLI(TMLI(I),TMLI(I-1),
     &  CR,CS,CG,FNSTAR(I),FNLAYER(I),FMLIEMISS,PTORR,DF,QFLUX(I))
        SUMQFLX = SUMQFLX + QFLUX(I)
      ENDDO
      QAVG = SUMQFLX/FLOAT(NLAYERP1)

      RETURN
      END

!=================================================================================

      SUBROUTINE MLICOEF(RMLI, PDMLI,TMLI,NVARMLI)

! Calculates numerical differentiation of Modified Lockheed equation
      DIMENSION  TMLI(10),TMLID(10), RMLI(10), RMLID(10), PDMLI(10,10)

      DELTA = 0.001

      DO I= 1, NVARMLI
        TMLID(I)= TMLI(I)
      END DO

! CAlculate Partial Derivatives

      DO J=1, NVARMLI
        IF (ABS(TMLI(J)) .LT. 1.E-10) THEN
            TMLI(J)= TMLI(J)+ DELTA
            ELSE
                TMLI(J)= (1.+ DELTA)* TMLI(J)
        END IF

        CALL MLIEQNS(RMLID,TMLI,NVARMLI)

! Calculate Partial Derivatives
      DO I=1, NVARMLI
        ANUM= RMLID(I)- RMLI(I)
        IF (ABS(TMLI(J)) .LT. 1.E-10) THEN
            PDMLI(I,J)= ANUM/DELTA
            ELSE
            PDMLI(I,J)= ANUM/(TMLI(J) *DELTA)
        END IF
      END DO
! Restore variables to original values
      TMLI(J)= TMLID(J)
      END DO
      RETURN
      END

!==================================================================================

      SUBROUTINE MLIEQNS(RMLI,TMLI,NVARMLI)

! Calculates residuals of the MLI Modified Lockheed Equation

      DIMENSION TMLI(10),RMLI(10)

      COMMON/CMLI/FNSTAR(10),FNLAYER(10),QFLUX(10),SAREA(10),
     &       TC,TH,PTORR,FMLIEMISS,SHRDEMISS,CR,CS,CG,DF

      NLAYERP1 = NVARMLI + 1

C     CALCULATE HEAT FLUX THROUGH EACH LAYER AND RADIATIVE HEAT FLUX FROM AMBIENT
      DO I = 1, NLAYERP1
       IF (I.EQ.1) CALL QFLUXMLI(TMLI(1),TC,CR,CS,CG,FNSTAR(1),
     &                         FNLAYER(1),FMLIEMISS,PTORR,DF,QFLUX(1))
       IF (I.EQ.NLAYERP1) CALL QFLUXRAD(TH,TMLI(NVARMLI),FMLIEMISS,
     &                                 SHRDEMISS,QFLUX(NLAYERP1))
```

```
       IF (I.NE.1.AND.I.NE.NLAYERP1)CALL QFLUXMLI(TMLI(I),TMLI(I-1),CR,
     &  CS,CG,FNSTAR(I),FNLAYER(I),FMLIEMISS,PTORR,DF,QFLUX(I))
      ENDDO
C     CALCULATE RESIDUAL
      DO I=1, NVARMLI
        RMLI(I)= QFLUX(I+1)*SAREA(I+1)- QFLUX(I)*SAREA(I)
      END DO
      RETURN
      END

!==============================================================================

      SUBROUTINE QFLUXMLI(T2,T1,CR,CS,CG,FNST,FNLR,FMLIEMISS,PTORR,
     &                    DF,QFLX)


! Constant for Modified Lockheed equation for 0.25mil Myler with Dacron Spacer layer

      TAVG=(T2+T1)/2
! Modified Lockheed Equation

       QFLX = DF*((((CS * (0.017+7.0E-6 * (800-TAVG) + 2.28E-2 *
     & log(TAVG)) * (FNST**2.68) * (T2-T1)))/FNLR)+((CR*FMLIEMISS*
     & ((T2**4.67)-(T1**4.67)))/FNLR)+(((CG*PTORR*
     & ((T2**0.52)-(T1**0.52))))/FNLR))

      RETURN
      END

C*********************************************************************
      SUBROUTINE QFLUXRAD(T2,T1,FMLIEMISS,SHRDEMISS,QFLX)
C     PURPOSE: CALCULATE RADIATIVE HEAT FLUX FROM AMBIENT
C*********************************************************************
      DATA SIGMA/5.67037E-08/

      ANUM = SIGMA*(T2**4 - T1**4)
      DENOM = 1./FMLIEMISS + 1./SHRDEMISS - 1.0
      QFLX = ANUM/DENOM

      RETURN
      END
```

A.5 User Subroutine for Example 30

```
C**********************************************************************
      SUBROUTINE SORCEQ(IPN,TERMD)
C     PURPOSE: ADD HEAT SOURCES
C     IPN - GFSSP INDEX NUMBER FOR NODE
C     TERMD - COMPONENT OF LINEARIZED SOURCE TERM APPEARING IN THE
C             DENOMINATOR OF THE ENTHALPY OR ENTROPY EQUATION
C**********************************************************************
       INCLUDE 'comblk.for'
C*********************************************************************
C*********************************************************************
C     ADD CODE HERE
      DATA TPROP/4460./

C        SET TEMPERATURE OF NODE TO PROPELLANT TEMPERATURE
         IF (NODE(IPN).EQ.1) THEN
            SORCEH(IPN) = SORCEMAS(IPN)*CPNODE(IPN)*TPROP
            TERMD = SORCEMAS(IPN)*CPNODE(IPN)
         ENDIF

      RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE BNDUSER
C     PURPOSE: MODIFY BOUNDARY CONDITIONS
C*********************************************************************
      INCLUDE 'comblk.for'
C*******************************************************************
C     ADD CODE HERE
C     DATA OF SOLID PROPELLANT BURN RATE
      DATA ABR,CNBR/0.0687,0.30/
      DATA PROPDEN/0.06/ ! LBM/IN**3
      DATA PROPOD/0.25/ ! INCH
      LOGICAL LBURN
      DATA LBURN/.TRUE./


C     OBTAIN INDICES FOR NODE AND BRANCHES
      CALL INDEXI(1,NODE,NNODES,IP1)
      CALL INDEXI(118,IBRANCH,NBR,IB118)
      CALL INDEXI(1819,IBRANCH,NBR,IB1819)

cmjo  Adding test on new logical
      if( lburn ) then

C        CALCULATE BURNING RATE, based on pressure at node 1
         PC = P(IP1)/144.
         RINCH = ABR*(PC**CNBR)

C        ESTIMATE NEW DIAMETER, Original Diameter set in VTASC
         ELPROP = BRPR1(IB118)*12. ! PROPELLANT LENGTH IN INCH
         RPROPM = BRPR2M(IB118)*12./2. ! PROPELLANT RADIUS IN PREVIOUS TIME STEP
         RPROP = RPROPM + RINCH*DTAU ! PROPELLANT RADIUS IN CURRENT TIME STEP
         BRPR2(IB118) = 2.*RPROP/12. ! PROPELLANT DIAMETER IN CURRENT TIME STEP

C        ESTIMATE MASS SOURCE FROM BURNING RATE
         VDOT = 2.*PI*RPROPM*ELPROP*RINCH
         SORCEMAS(IP1) = PROPDEN*VDOT

C        ADJUST AREA OF COMPRESSIBLE ORIFICE
         AREA(IB1819) = PI*BRPR2(IB118)**2/4.




      else

cmjo     When I added this statement, it pretty much zero'ed out the pressure in the
grain&plenum
         SORCEMAS(IP1) = 0.

      end if


      IF (BRPR2(IB118) .GE. PROPOD/12. ) then

cmjo     Turn off burning propellant??
         lburn = .false.
         RELAXNR = 0.3

c         write(nprnt,*) ' '
c         write(nprnt,*) ' Stopping in BNDUSER'
c         write(nprnt,*) ' Expanding burning surface reached OD'
c         write(nprnt,*) ' Shouldnt really be a stop condition'
c         STOP
      end if

C        PLOT NEW VARIABLES
C        CALCULATE THRUST IN LBF
```

```
   CALL INDEXI(1617,IBRANCH,NBR,IB1617)
   FORCE = FLOWR(IB1617)*VEL(IB1617)/GC
   USRVAR = .TRUE.
   USRVARSNUM = 3
   USRPVARNAME(1) = 'RINCH'
   USRPVARUNIT(1) = 'in/s'
   USRPVARNAME(2) = 'SORCEMAS'
   USRPVARUNIT(2) = 'lb/s'
   USRPVARNAME(3) = 'Thrust'
   USRPVARUNIT(3) = 'lbf'
   USRPVAR(1) = RINCH
   USRPVAR(2) = SORCEMAS(IP1)
   USRPVAR(3) = FORCE

RETURN
END
```