Generalized Fluid System Simulation Program (GFSSP)
Supplementary Documentation for Version 701

Alok Majumdar, Andre Leclair, and Ric Moore
ER43/Thermal Analysis Branch
&
Suzanne Dorney
ER42/Fluid Dynamics Branch

NASA/Marshall Space Flight Center
Huntsville, Alabama – 35812

October 2015

Summary

Since the release of GFSSP Version 605 and the User's Manual [1], the development of Version 7 started. This document describes the additional features of Version 701 and can be used as a supplement to the Version 6 User's Manual. The additional capabilities added to Version 701 are: 1) Psychrometric Property Package that allows calculation of environmental property parameters such as relative humidity, wet-bulb temperature, humidity ratio and dew point temperature, 2) Provision of Multi-Layer Insulation (MLI) on a Cryogenic Tank that permits calculation of heat leak from the ambient, 3) Incorporation of a plotting program, Multiplot, that allows animation of predicted parameters during transient simulation, 4) Revised property package for hydrogen-peroxide, and 5) Instructions to use REFPROP from User Subroutine. Several example problems were modified or added in Version 701. They are: *i*) Example 4 - Simulation of the Mixing of Combustion Gases and a Cold Gas Stream, *ii*) Example 10 - Pressurization of a Propellant Tank, *iii*) Example 12 – Helium Pressurization of LOX and RP-1 Propellant Tanks, *iv*) Example 29 - Self-Pressurization of a Cryogenic Propellant Tank Due to Boil-off, *v*) Example 31 - Modeling Psychrometrics of Air-Water Vapor mixture with GFSSP, *vi*) Example 32 – Flow Distribution in Manifold.

Topics

Acknowledgement

The authors would like to acknowledge the support of MSFC's NIRPS (National Institute of Rocket Propulsion System), KSC's LSP (Launch Service Program), and GRC's e-CRYO Project. The encouragement and support of Arthur Werkheiser(e-CRYO), Wendy Cruit/Tony Lavoie (SLS-Stages), Danny Garcia (Tech Transfer), and Alicia Turpin of MSFC/ER43, Paul Schallhorn of KSC, and Jeff Moder of GRC are gratefully acknowledged.

1. New Features

Several new features are added in Version 701. They are: a) Psychrometric Property Package that allows calculation of environmental control parameters such as relative humidity, humidity ratio and dew point temperature; b) Heat leak calculation through MLI blankets that permits estimation of boil-off in a cryogenic tank; c) Provision of animation of transient simulation using Multiplot program; d) Revised property package for hydrogen-peroxide, and e) Instructions to use REFPROP from User Subroutine.

1.1 Psychrometric Property Package

The purpose of including psychrometric property package in GFSSP is to perform analysis to assess the potential of condensation during thermal transient for Environmental Control. GFSSP's capability has been extended to compute psychrometric properties such as Dew Point Temperature, Relative Humidity, Wet Bulb Temperature and Humidity Ratio.

Definition of Psychrometric Properties

In this section the calculation of psychrometric properties are explained[2]. Figure 1.1 shows the thermodynamic state of water in moist air.

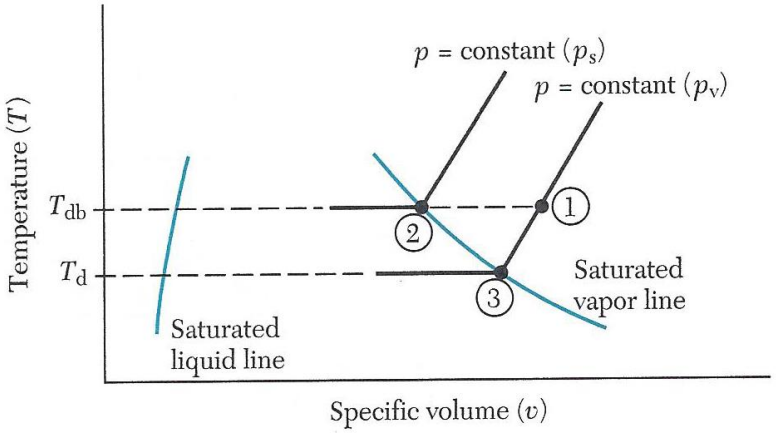


Figure 1,1 Thermodynamic state of moist air

State 1 represents a typical state of water vapor in moist air when relative humidity is less than 100%. The temperature at state 1 is the dry bulb temperature (T_{db}). State 2 and 3 are located in the saturation line and therefore, relative humidity is 100%. Saturation temperature at vapor pressure, p_v is the dew point temperature (T_d). p_s is the vapor pressure at T_{db} .

In psychrometric calculation Dalton's law of partial pressure is assumed.

$$p = p_{a+}p_v \qquad (1.1)$$

The relative humidity is defined as

$$\varphi = \frac{p_v}{p_s} \qquad (1.2)$$

The humidity ratio is defined as

$$\omega = \frac{m_v}{m_a} = \frac{0.622p_v}{p_a} \qquad (1.3)$$

The Carrier Equation is an empirical equation for calculating wet bulb temperature expressed as:

$$\frac{(p-p_{wb})(T_{db}-T_{wb})}{2831-1.43T_{wb}} = p_{wb} - p_v \qquad (1.4)$$

The vapor pressure relation for water can be expressed as:

$$\ln(p_{sat}) = A + \frac{B}{T_{sat}} + Cln(T_{sat}) + DT_{sat} \qquad (1.5)$$

where A = 99.4824; B=-7894.6011; C=-11.9783; D = 0.01101

All psychrometric calculations are done in SI units (pressure in Pascal and temperature in degree Celsius). However the Carrier equation is an empirical equation in Engineering Unit (pressure in psi and temperature in degree Fahrenheit). In equation (1.5), pressure is in Pascal and temperature is in degree Kelvin.

Psychrometric Property Calculation in GFSSP

Four subroutines are added in GFSSP to perform psychrometric calculation:

SUBROUTINE PSAT(T,P)

This subroutine calculates saturation pressure of water at a given temperature using equation 1.5.

SUBROUTINE TSATT(P,T,TGUESS)

This subroutine calculates saturation temperature of water at a given pressure by iteratively solving vapor pressure relation (equation 1.5) by Newton-Raphson method.

SUBROUTINE TWBCAR(TDB,PDP,PAMB,TWB)

This subroutine calculates wet bulb temperature by iteratively solving Carrier equation (equation 1.4) by Newton-Raphson method.

SUBROUTINE CARIER(TWB,TDB,PWB,PAMB,PDP)

This subroutine calculates pressure at the dew point temperature (p_v) from Carrier equation (equation 1.4).

A control parameter, IOPTPSY has been introduced. Depending upon the value of IOPTPSY, GFSSP will activate psychrometric calculation and decide which input parameter to use. Here are the possible options for IOPTPSY.

IOPTPSY = 0: PSYCHROMETRIC PROPERTY INACTIVE
IOPTPSY = 1: INPUT RELATIVE HUMIDITY (PHI)
IOPTPSY = 2: INPUT WETBULB TEMPERATURE (TWB)
IOPTPSY = 3: INPUT HUMIDITY RATIO (OMEGA)

When Psychrometric Option is active, GFSSP reads either PHI, TWB or OMEGA for both steady-state and transient models. Boundary History File requires one of the three property in addition to pressure, temperature and concentration. GFSSP uses water from the GASP/WASP property package and air from the GASPAK package.

Example 31 illustrates the use of psychrometric option.

1.2 Heat Leak Calculation across Multi-Layer Insulation (MLI)

Heat transfer through the MLI can be expressed by the Modified Lockheed equation [3]:

$$q = \left[\frac{C_s\left(0.017 + 7E-6*\left(800 - T_{\text{avg}}\right) + 2.28E-2*\ln\left(T_{\text{avg}}\right)\right)(N^*)^{2.63}\left(T_h - T_c\right)}{N_s} \right.$$

$$\left. + \frac{C_r \varepsilon\left(T_h^{4.67} - T_c^{4.67}\right)}{N_s} + \frac{C_g P\left(T_h^{0.52} - T_c^{0.52}\right)}{N_s} \right]. \qquad (2.1)$$

The actual heat transfer, however, was calculated by introducing a degradation factor, D_f. The heat transfer rate through the MLI was expressed as:

$$q_{MLI} = D_f\, q \qquad (2.2)$$

where constants

C_s = 2.4×10^{-4}
C_r = 4.944×10^{-10}
C_g = 14,600

and variables and units

q = heat flux through MLI (W/m^2)
T_{avg} = average of hot and cold boundary temperatures (K)
N^* = MLI layer density (layers/cm)
T_h = hot boundary temperature (K)
T_c = cold boundary temperature (K)
N_s = number of MLI layers
ε = MLI layer emissivity ($\varepsilon = 0.031$)
P = interstitial gas pressure (torr).

Typically, several MLI blankets constitute the MLI. The mathematical modeling methodology is shown in Figure 2.1. According to the law of energy conservation:

$$Q_{rad} = Q_1 = Q_2 = Q_3, \tag{2.3}$$

where radiative heat transfer from the shroud to the outer layer of MLI is given as:

$$q_{rad} = \frac{\sigma\left(T_{amb}^4 - T_{outer}^4\right)}{\frac{1}{\varepsilon_{MLI}} + \frac{1}{\varepsilon_{shrd}} - 1}. \tag{2.4}$$

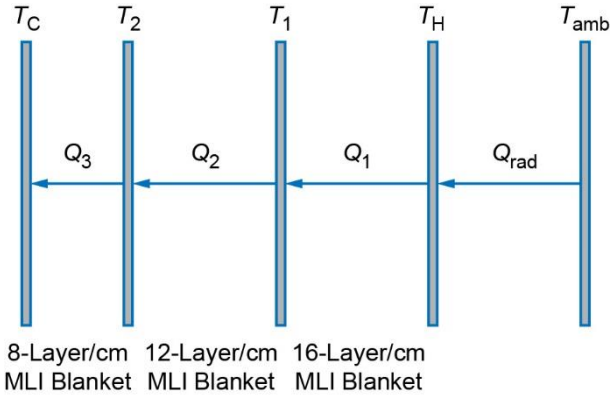


Figure 2.1. MLI modeling methodology.

The law of energy conservation can also be expressed as:

$$Q_2(T_1, T_2) - Q_3(T_2, T_c) = 0, \tag{2.5}$$

$$Q_1(T_h, T_1) - Q_2(T_1, T_2) = 0, \tag{2.6}$$

and

$$Q_{rad}(T_{amb}, T_H) - Q_1(T_H, T_1) = 0 \tag{2.7}$$

Equations (2.5)–(2.7) are the governing equations to calculate temperature at the outer boundary (T_H) and two intermediate temperatures (T_1 and T_2) by the Newton-Raphson method. Subroutine MLI_HEAT_RATE was developed to solve these equations. Figure

2.2 shows the flowchart of the MLI_HEAT_RATE Subroutine that was called from Subroutine TSOLID or TSEQNS of GFSSP Source code.

MLI_HEAT_RATE Subroutine calls MLIEQNS, MLICOEF, and GAUSSY to perform main computational tasks. MLIEQNS calculates residuals of the governing equations (Eqs. (2.5)–(2.7)). Equations (2.1) and (2.2) are computed in QFLUXMLI and Eq. (2.4) is computed in QFLUXRAD. The coefficients of the correction equation are computed in MLICOEF. The correction equations are solved in GAUSSYMLI.

Example 29 demonstrates the use of MLI in a cryogenic tank.

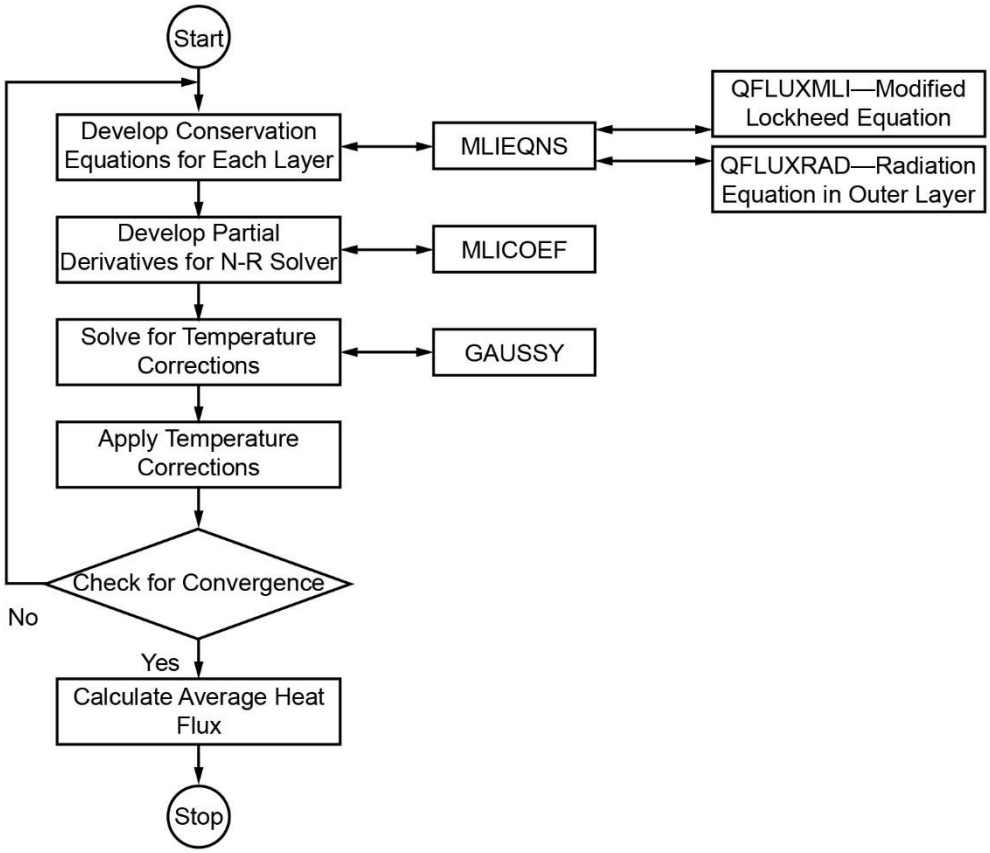


Figure 2.2. Flowchart of MLI_HEAT_RATE Subroutine.

1.3 Multiplot Program for Animation

MultiPlot is an interactive plotting package specifically designed to visualize the output of time-dependent simulations run using the Generalized Fluid System Simulation Program (GFSSP) [1]. MultiPlot adds to the existing visualization capabilities incorporated into the graphical user interface to GFSSP, the Visual Thermofluid Dynamic Analyzer for Systems and Components (VTASC[1]). VTASC currently has an option to use the utilities WinPlot (https://software.nasa.gov/software/MFS-31664-1) or TecPlot (http://www.tecplot.com), two standalone plotting packages. The new option to plot

results using MultiPlot will add the ability to animate line plots. Specifically, the results for each node can be displayed on a single plot that is animated over time.

1.3.1 Format of Input File

The input file used by MultiPlot is written specifically by GFSSP. The format of the file and how it is read into MultiPlot is discussed below.

The input file for MultiPlot is an ASCII file that follows a specific format. The first four lines contain header information. Each of these lines starts with a key word followed by the data that is specific to the particular case. Table 1.3.1, shown below, gives details on this header information.

Keyword	Meaning	Details
CaseName:	The name of the case	This value is used as the title on the constructed plot/animation. The title can contain blanks. Code will read to the end of the line. Ex: CaseName: Valve Opening
TimeSteps:	Number of time steps in the simulation	For each timestep in the simulation a plot of value vs nodes will be constructed. The animation will then run through each of these plots Ex: TimeSteps: 100
Nodes:	The number of nodes in the simulation followed by the name of each node	The name of each node is a numeric value but the order of the nodes does not need to be in numeric order. Therefore, the list of node names is given for each node. Ex: Nodes: 12 1 2 3 4 5 6 7 8 9 10 11 12
Values:	The number of values contained in the file followed by the name and units of each value	The number of separate function values contained in the file is specified on this line. For each value its name and units are then specified. Ex: Values 2 Press PSIA Temp DEG_F

Table 1.3.1. Explanation of Header Lines

After the four header lines follows the data for each time step. Each line starts with the time value for that time step. After the time value comes each of the function values for each node in the simulation.

Figure 1.3.1 shows part of an example data file. The four header lines are given first. Then the first three data lines are shown. The values highlighted in blue are time values, the red color block highlights the pressure values for each of the 12 nodes in the simulation. The green color block highlights the temperature values for the 12 nodes in the simulation.

```
CaseName: Valve Opening
TimeSteps: 100
Nodes: 12 1 2 3 4 5 6 7 8 9 10 11 12
Values: 2 Press PSIA Temp DEG_F
0.050000 102.900002 14.700000 14.700000 14.700000 14.700000
14.700000 14.700000 14.700000 14.700000 14.700000 14.700000
14.699100 60.000000 59.999901 59.999901 59.999901 59.999901
59.999901 59.999901 59.999901 59.999901 59.999901 59.999901
59.999901
0.100000 102.900002 14.700000 14.700000 14.700000 14.700000
14.700000 14.700000 14.700000 14.700000 14.700000 14.700000
14.699100 60.000000 59.999901 59.999901 59.999901 59.999901
59.999901 59.999901 59.999901 59.999901 59.999901 59.999901
59.999901
0.150000 102.900002 14.700500 14.700000 14.700000 14.700000
14.700000 14.700000 14.700000 14.700000 14.700000 14.700000
14.699100 60.000000 59.999901 59.999901 59.999901 59.999901
59.999901 59.999901 59.999901 59.999901 59.999901 59.999901
59.999901
```

Figure 1.3.1. Example Data File

1.3.2 Reading Input File into MultiPlot

On the main menu bar of MultiPlot is the File option. Currently this option contains the two choices *Load MultiPlot File* and *Exit*. This process is shown in Figure 1.3.2.

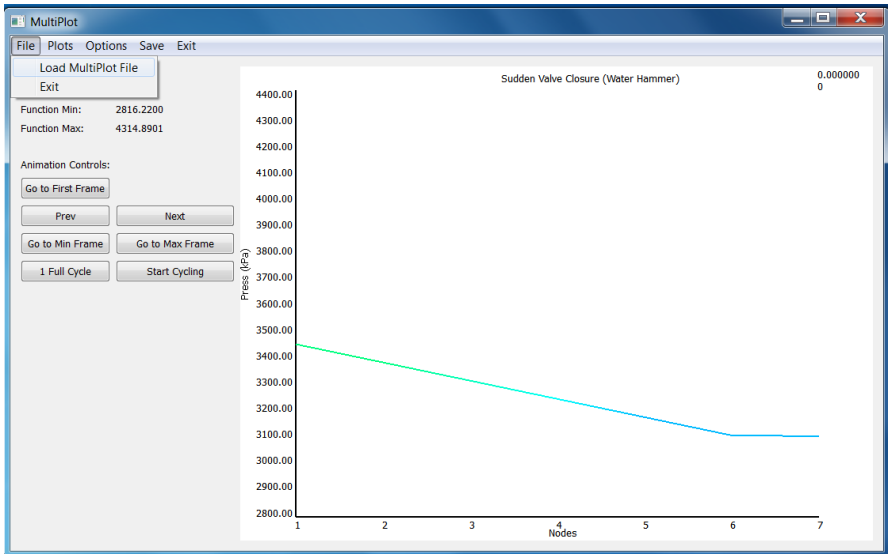


Figure 1.3.2. How to Read in MultiPlot File

1.3.3 Navigating Through Animation

Once the data file has been read in, a default plot is constructed. The default plot represents the first time step and all nodes for the first function listed in the input file. Figure 1.3.3 shows such a plot generated from the data shown in Figure 1.3.1. The title on the plot is "Valve Opening". The default function is the first one listed in the data file. In this example that function is pressure. The values for all twelve nodes are displayed. Once this plot has been displayed it is then possible to move through each time step as well as view an animation of the full solution over time.

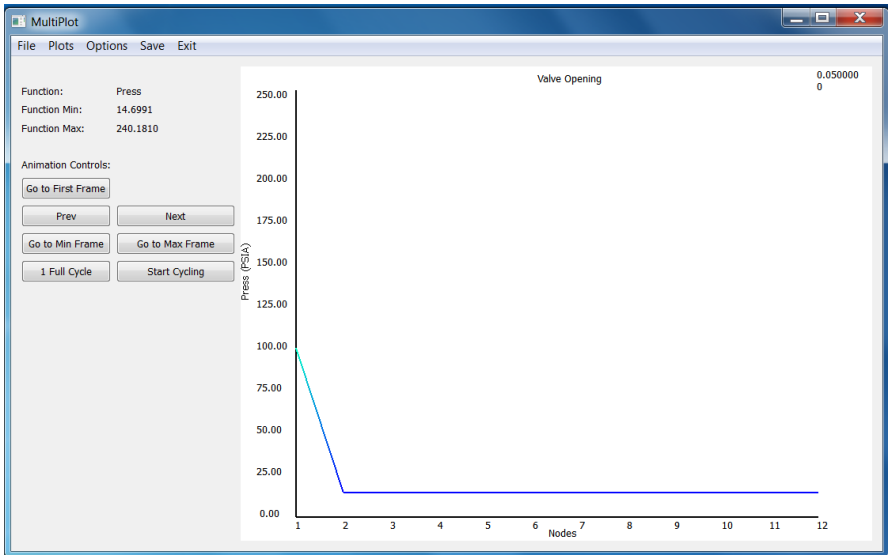


Figure 1.3.2. Default Plot of First Time Step

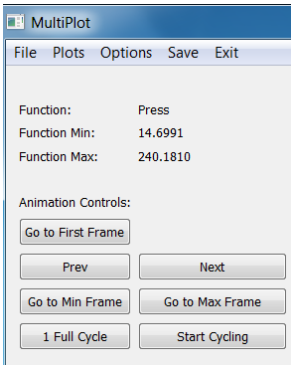


Figure 1.3.3. Control Options

Figure 1.3.4 is a close up view of the control options provided for moving through the different time steps of the animation. The button labeled ***Go to First Frame*** does just that. It moves the animation to the first frame. In the case of a long simulation where there are numerous time steps the ability to move all the way back to the beginning is sometimes useful.

The buttons labeled ***Prev*** and ***Next*** allow the user to move to the previous frame, or time step, and the following frame respectively. These options move only a single frame either backwards or forwards. If the user is currently looking at the first frame, as in Figure 1.3.3, and the ***Prev*** button is selected the display will move to the last frame of the simulation. Figure 1.3.5 shows the resulting plot if the ***Prev*** button were selected.

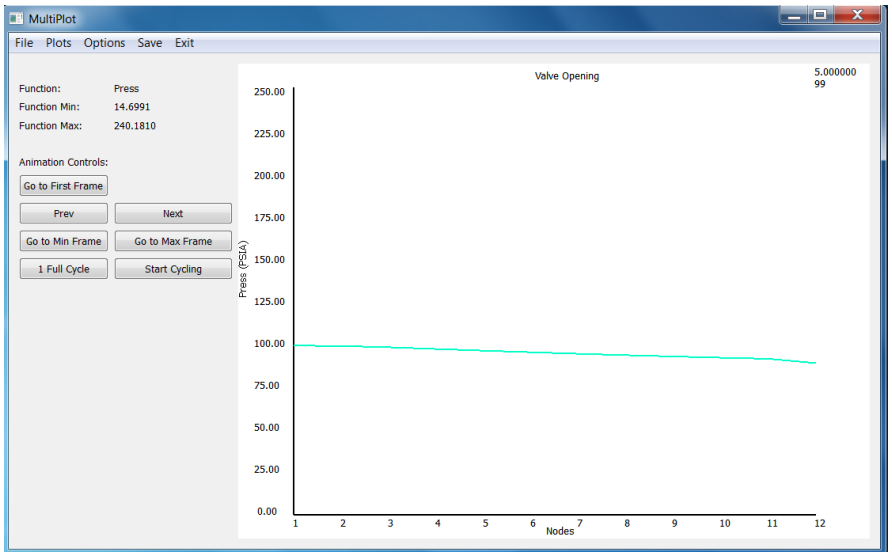


Figure1.3.4. Last Frame of Animation

Similarly, if the *Next* button is pressed the animation moves ahead to the next time step and cycles back to the initial time step if currently at the last frame.

The next two buttons, ***Go to Min Frame*** and ***Go to Max Frame***, show the specific frames where the maximum and minimum values, respectively, are reached. It is sometimes useful to be able to identify the point where these relative values are reached, and this allows the user to go directly to that frame in the animation.

The remaining two buttons, ***1-Full-Cycle*** and ***Start Cycling*** allow for MultiPlot to move through multiple frames without having to step through each frame individually. The ***1-Full-Cycle*** button will start at the initial frame and show all time frames in sequence until the last frame has been reached. This option will always cycle through the entire sequence of time steps, starting at the first frame and ending with the last frame.

The ***Start Cycling*** button will start at the current frame being displayed and continually show the next frame. Once this option is selected the animation will start to sequence through each frame and the text on the button will change from ***Start Cycling*** to ***Stop Cycling***. The animation will continuously cycle through all of the frames until the ***Stop Cycling*** button is pressed.

1.3.4 Customization of Animation

The ability to customize the display of the animation is also possible. One of the tabs in the menu of the main window is *Options*. This tab currently has the single choice *Animation Options*. The selection of this panel is shown in Figure 1.3.6.

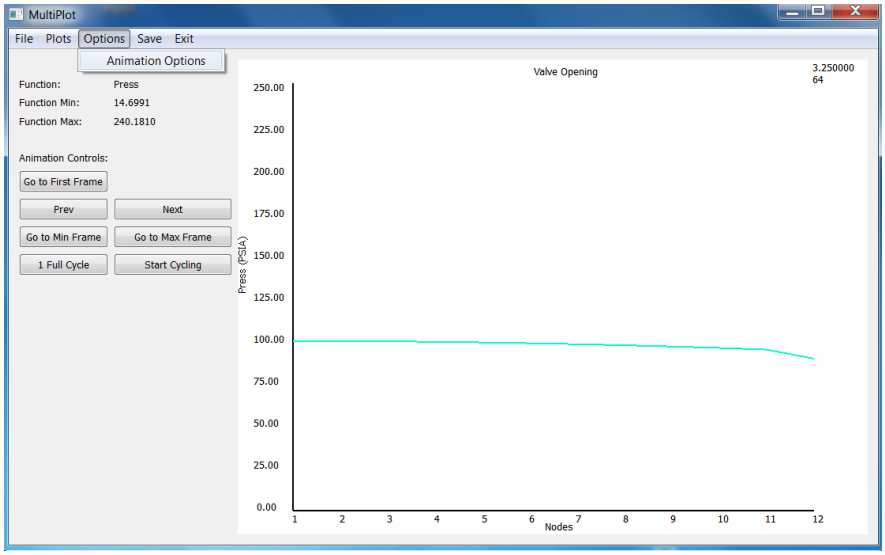


Figure 1.3.5. How to Open Animation Control Panel

The panel that allows the user to make modifications to the format of each frame is shown in Figure 1.3.7.

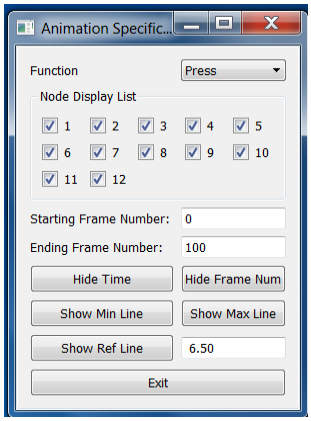


Figure 1.3.6. Animation Control Panel

Function Selection

When the input file is read in all function values are stored. The *Function* option in the *Animation Options* panel allows the user to select which function is to be displayed in the plot. By default, the first function is plotted initially. In the partial data file shown in Figure 1.3.1, the user has the option of plotting Press or Temp.

Whenever a new function is selected the values displayed in the main window will be updated to reflect the change in function. In the example data file used in this manual the values for pressure are shown in Figure 1.3.8.

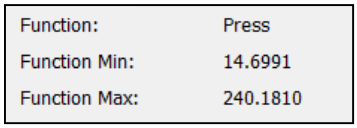


Figure 1.3.7. Current Function Values

Node Display List

When the default plot is constructed the values for each node in the GFSSP simulation are displayed. The section of the *Animation Options* panel titled *Node Display List* allows the user to select which node's values are actually displayed.

Each node is listed individually. By default each node is displayed. To turn off the display of a particular node the use simply clicks on the check box for that particular node.
Note that when the display status of a node is modified the maximum and minimum values are recalculated to include or exclude that node's values from the calculation.

Frame Selection

By default all time steps are shown in the animation. The starting frame is frame 0 and the ending frame is the last time step in the animation. The user has the ability to set the starting and ending frames. The cycle options will display only those frames between the starting and ending values.

Note that when the starting or ending frame value is modified the maximum and minimum values are recalculated to reflect these changes.

Display Values

There are five additional items that can be displayed each frame of the animation. The frame number and the time step value appear by default in the upper right hand corner of each plot. These values are highlighted in Figure 1.3.9. The display of each of these values can be turned on and off by toggling the associated buttons, ***Hide Time/Show Time*** and ***Hide Frame Num/Show Frame Num***.

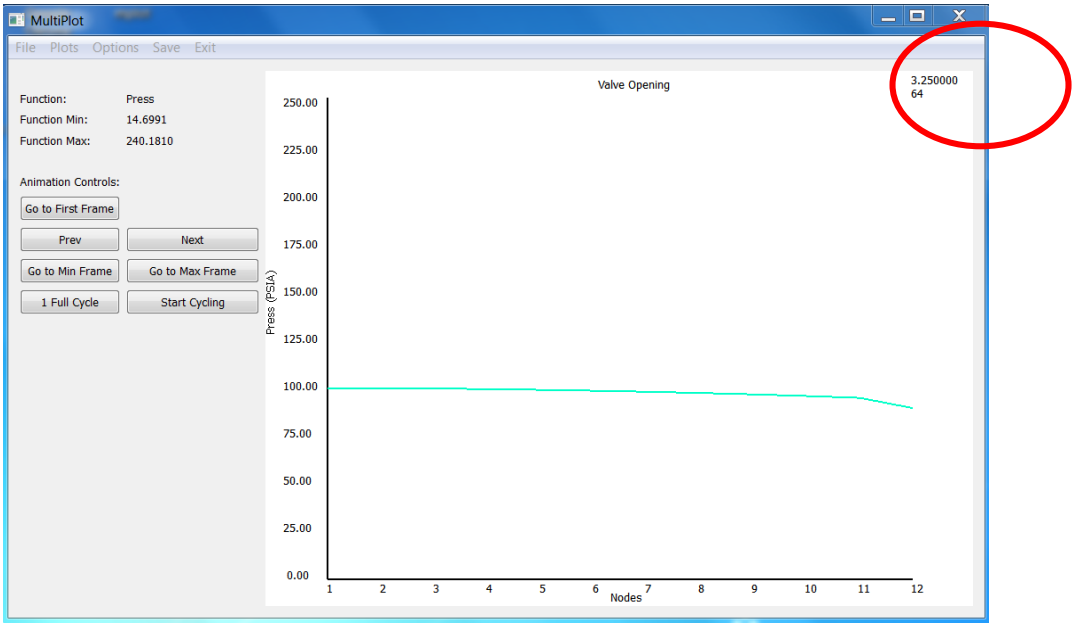


Figure 1.3.8. Frame Number and Time Step Value

It is also possible to display lines on the plot that represent the maximum and minimum function value. These lines are turned on or off by toggling the ***Show Min Line/Hide Min Line*** and ***Show Max Line/Hide Max Line*** buttons on the *Animation Options* panel. Turning on the display of both of these lines is shown in Figure 1.3.10. The maximum fuction value line is drawn in red and the mimimum function value line is drawn in blue. The final item that can be displayed on each frame is a user defined reference line. In many cases there is a value of interest that is not necessarily the maximum or minimum function value. The display of this line is controlled by toggling the ***Show Ref Line/Hide Ref Line*** button. The specification of the lines value is done in the text box on the side of the toggle button. Figure 11 shows the display of a reference line. The reference line is the horizontal line drawn in green. In this example the reference line is drawn at the value 130.0.

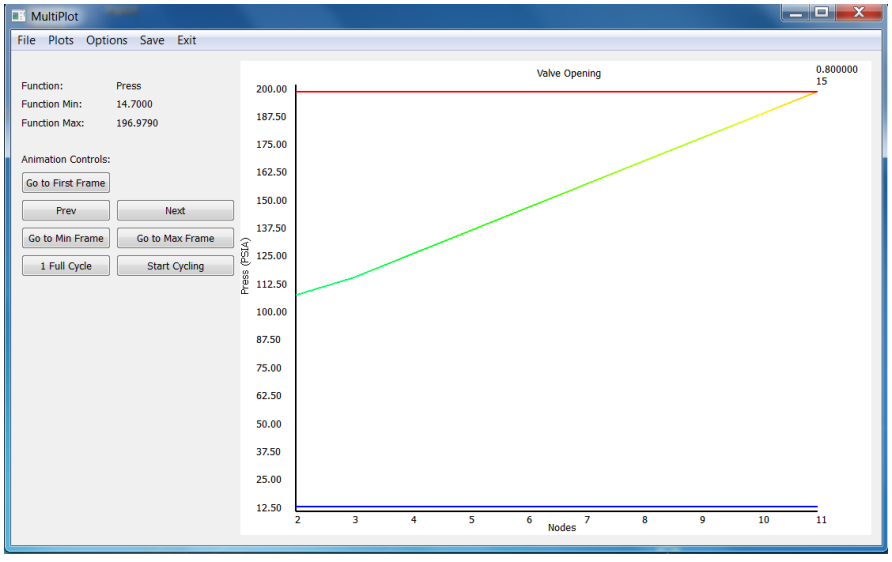


Figure 1.3.9. Display of Max and Min Value Lines

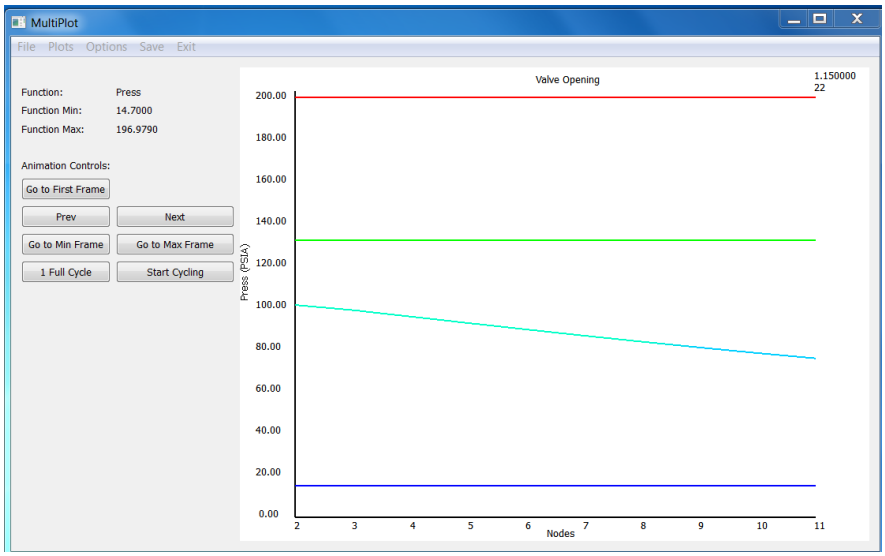


Figure 1.3.10. Display of a Reference Line

1.3.5 Save Options

In addition to generating plots in the display window, MultiPlot gives the user two options to save individual frames as JPEG files. The use of this option is shown in Figure 1.3.12.

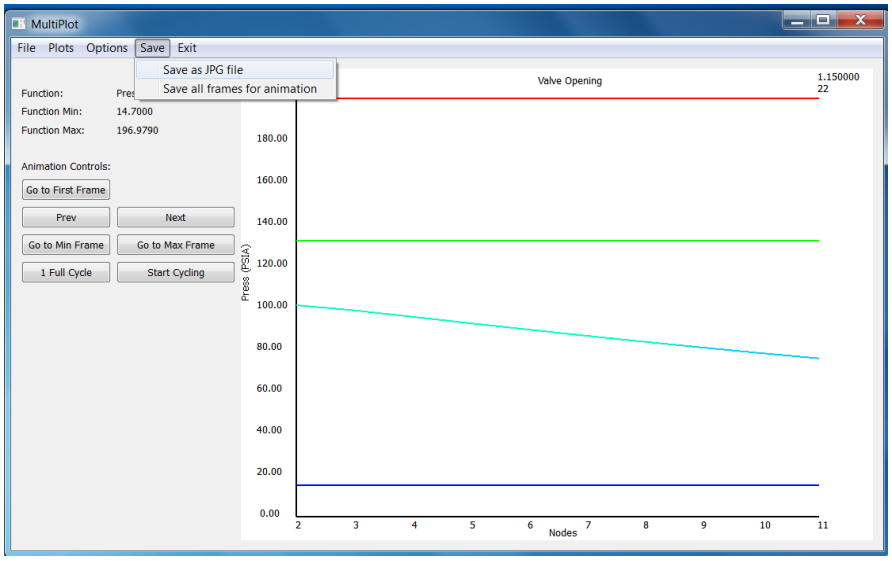


Figure 1.3.12. Save Options

The first option, *Save as JPG file*, will save the current frame as a .jpg file. The second option, *Save all frames for animation*, will save a separate .jpg file for each frame in the current screen animation. One file for each from between the start and end frame values. Once this collection of files has been constructed the user can use a separate tool, ffmpeg[4], to generate an mpeg animation file.

1.4 Revised property package for hydrogen-peroxide

Previous versions of GFSSP allowed the user to select hydrogen peroxide (H2O2) as a fluid, with properties calculated by the GASPAK property package. GASPAK handles peroxide properties as a mixture of peroxide and water, with some user specified mole fraction. In GFSSP, boundary and initial properties are based on the input pressure and temperature. But GFSSP's energy equation is based on enthalpy, and after the energy equation is solved, the property package is asked to return the temperature given pressure and enthalpy. Unfortunately, for a given pressure, GASPAK's peroxide properties are not self-consistent between calls given temperature and calls enthalpy. This leads to incorrect internal node temperatures or failure of the model to converge.

For v701, GFSSP's calls to the GASPAK property package were modified so that they were always done given pressure and temperature. When properties are needed given enthalpy, an iterative solution scheme varies the temperature until the proper enthalpy is found. Note that this method precludes the ability to model phase change, during which enthalpy varies while temperature stays constant. In any case, GASPAK's peroxide properties have not been validated in the vapor phase, when decomposition makes property predictions difficult. For this reason, the maximum peroxide temperature is limited to the boiling point at any pressure.

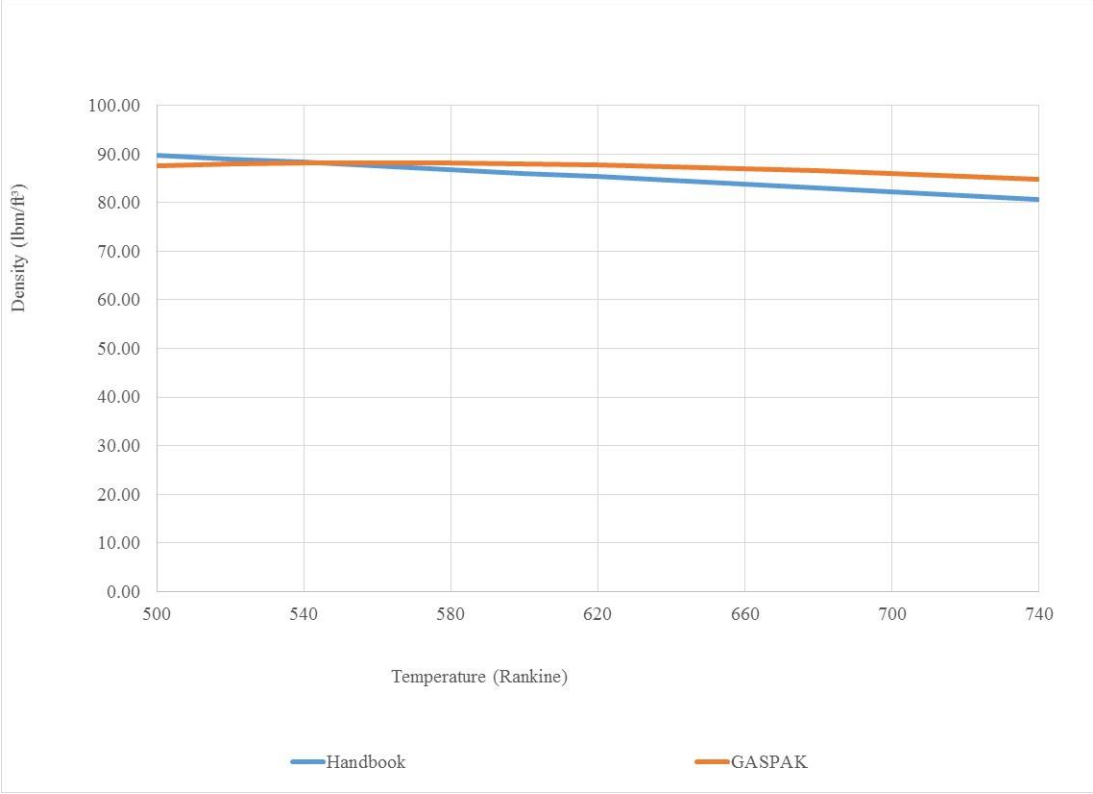


Figure 1.4.1: Density of 95 wt% H2O2 as a Function of Temperature

A study was made to compare GASPAK's predicted peroxide properties to those given in the literature, mainly Rocketdyne's Hydrogen Peroxide Handbook, 1967. The plots below compare the properties of 95 wt% peroxide (approximately 9 mol% water) at a pressure of 14.7 psia. It is seen that there is some disagreement; however, properties should still be accurate enough for many engineering calculations.

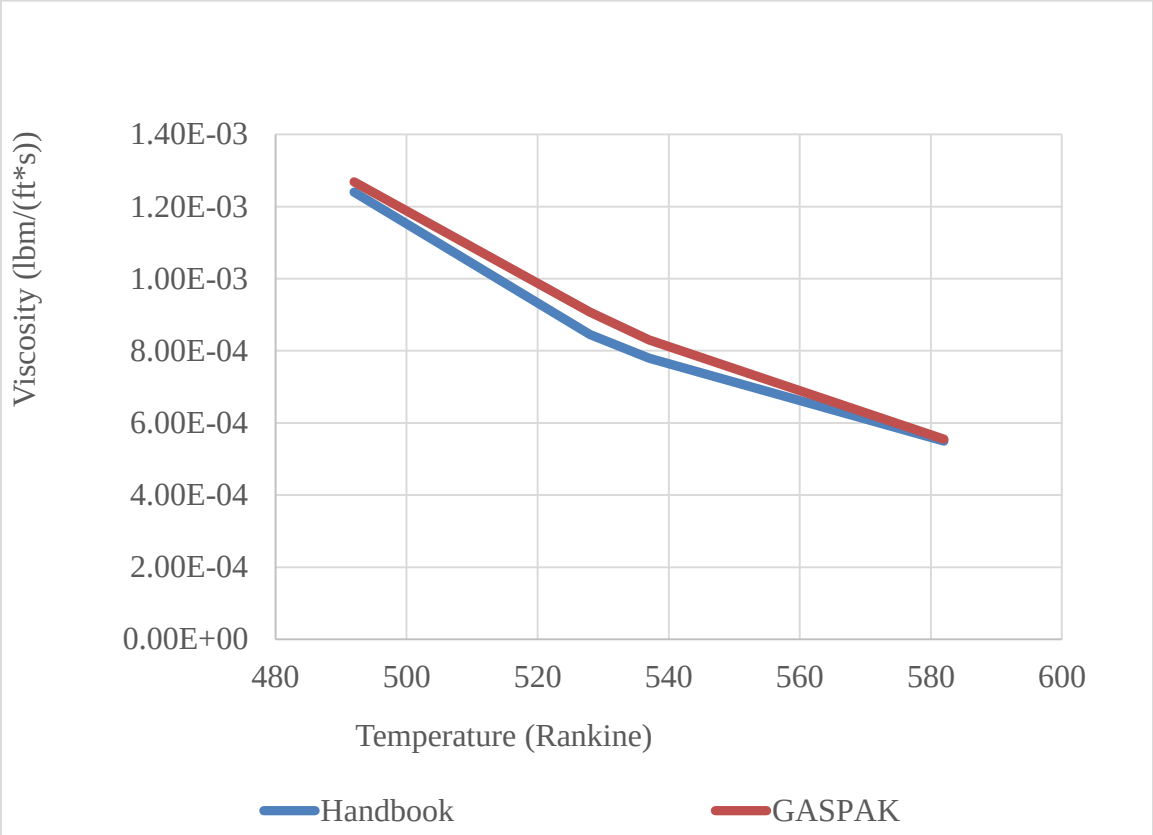


Figure 1.4.2: Viscosity of 95 wt% H2O2 as a Function of Temperature

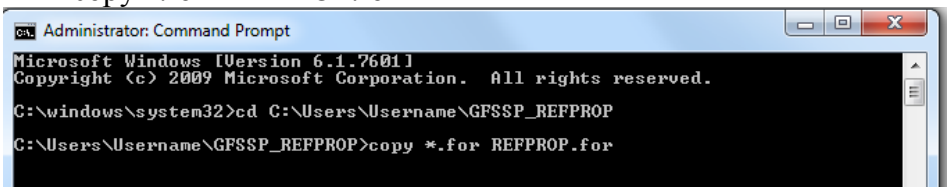
1.5 Instructions to use REFPROP from User Subroutine

This section provides instructions for calling REFPROP Fortran code from GFSSP's user subroutines. You must have REFPROP installed on your computer and a Fortran compiler (Intel, Compaq, or G95).

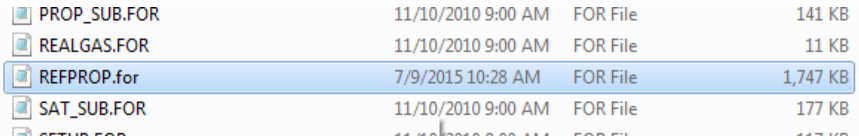
PART 1: CREATE THE REFPROP FORTRAN FILE

1. In your GFSSP model folder, create a subfolder to hold the REFPROP Fortran files.
2. Copy all files from the REFPROP Fortran folder (usually C:\Program Files(x86)\REFPROP\Fortran) into your working folder.
3. Delete the file "Pass_FTN.for" from the copied files in your working folder.

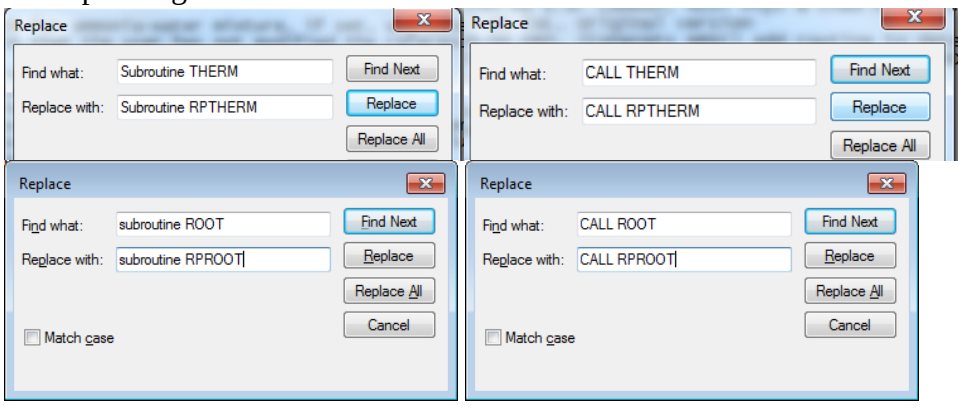
4. Combine all remaining Fortran files into a file called “REFPROP.for”. This can be achieved using the command prompt.
    a. Open the command prompt.
    b. Change the directory in the command prompt to your working directory by typing “cd“ followed by a space, then the address of your working directory, for example:
        “cd C:\Users\Username\Documents\GFSSP_REFPROP”
    c. Once the directory is set to your working directory, type the command:
        “copy *.for REFPROP.for”



    d. A new file named ‘REFPROP.for’ will be in your working directory.



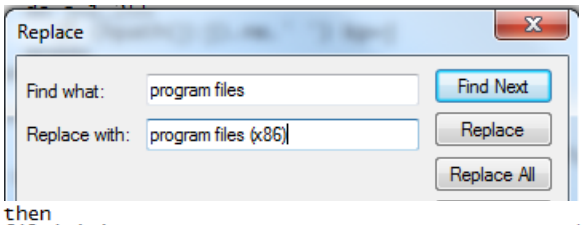
5. Open the newly created file “REFPROP.for” file with a text editor such as Notepad.
6. Since GFSSP and REFPROP use the same subroutine name for two subroutines, it will be necessary to rename those two in the REFPROP.for file to prevent any conflicts. The subroutines are “THERM” and “ROOT”. The easiest way to change these is by using the replace feature in notepad. Find all mentions of “Subroutine THERM” and “Call THERM” and replace them with “Subroutine RPTHERM” and “Call RPTHERM” respectively. Do the same for “ROOT”. Be aware that there are three additional subroutines “THERM0”, “THERM2”, and “THERM3” which will be renamed. This will have no adverse effects, since all calls will match their corresponding subroutines.



7. Next the REFPROP directory should be updated to ensure that the correct location is given to call the fluid and mixture files. The location is set in the subroutine OPENFL. The directory in the REFPROP.for file should match REFPROP’s installation directory. Please note that the location is set twice, once for the fluids and once for the mixtures. A common difference is that the directory is given as

"C:\Program Files\" when REFPROP was installed in the directory "C:\Program Files (x86)\".

```
      if (idir.ne.2) then
        if (k.eq.1) hfilei=' '                                  ! Current directory
        if (k.eq.2) hfilei='fluids'//bs                         ! Fluids subdirectory
        if (k.eq.3) hfilei=bs//'refprop'//bs//'fluids'//bs      ! From \Refprop\Fluids\
        if (k.eq.4) hfilei='..'//bs//'fluids'//bs               ! From ..\Fluids\
        if (k.eq.5) hfilei=':fluids:'                           ! Mac
        if (k.eq.6) hfilei='c:'//bs//'Program Files'//bs//      ! C:\Program files\Refprop\Fluids
&                          'Refprop'//bs//'fluids'//bs
      else
        if (k.eq.1) hfilei=' '
        if (k.eq.2) hfilei='mixtures'//bs
        if (k.eq.3) hfilei=bs//'refprop'//bs//'mixtures'//bs
        if (k.eq.4) hfilei='..'//bs//'mixtures'//bs
        if (k.eq.5) hfilei=':mixtures:'
        if (k.eq.6) hfilei='c:'//bs//'Program Files'
&                            //bs//'Refprop'//bs//'mixtures'//bs
      endif
```




```
      if (idir.ne.2) then
        if (k.eq.1) hfilei=' '                                  ! Current directory
        if (k.eq.2) hfilei='fluids'//bs                         ! Fluids subdirectory
        if (k.eq.3) hfilei=bs//'refprop'//bs//'fluids'//bs      ! From \Refprop\Fluids\
        if (k.eq.4) hfilei='..'//bs//'fluids'//bs               ! From ..\Fluids\
        if (k.eq.5) hfilei=':fluids:'                           ! Mac
        if (k.eq.6) hfilei='c:'//bs//'Program Files (x86)'//bs// ! C:\Program files\Refprop\Fluids
&                          'Refprop'//bs//'fluids'//bs
      else
        if (k.eq.1) hfilei=' '
        if (k.eq.2) hfilei='mixtures'//bs
        if (k.eq.3) hfilei=bs//'refprop'//bs//'mixtures'//bs
        if (k.eq.4) hfilei='..'//bs//'mixtures'//bs
        if (k.eq.5) hfilei=':mixtures:'
        if (k.eq.6) hfilei='c:'//bs//'Program Files (x86)'
&                            //bs//'Refprop'//bs//'mixtures'//bs
```

8. The "REFPROP.for" file should now be ready. Save the file and copy it to your model's working directory.

PART 2: CREATE AND COMPILE THE USER SUBROUTINE

1. Instead of the standard GFSSP user subroutine file ("userrtn701.for"), copy the file "userrtn701.REFPROPcall.for" to your model working directory. This file contains code in subroutine PRPUSER to overwrite the fluid properties calculated by the property codes built into GFSSP.

```
       IF (I_GIVEN .EQ. 1) THEN
          CALL REFPROP_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)
       ELSE IF (I_GIVEN .EQ. 2) THEN
          CALL REFPROP_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)
       ELSE IF (I_GIVEN .EQ. 3) THEN
          CALL REFPROP_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)
 END IF
```

This code calls one of three subroutines, depending on whether properties are to be calculated as a function of P/T, P/H, or P/S.

2. The three subroutines already contain the REFPROP filenames for helium, nitrogen, oxygen, hydrogen, and water. If you will be using a different REFPROP fluid, modify the code that stores the filename in variable hf(1) in each of the three subroutines.

```
IF (I_NFLUID .EQ. 1) THEN
   hf(1) = 'helium.fld'
ELSE IF (I_NFLUID .EQ. 4) THEN
   hf(1) = 'nitrogen.fld'
ELSE IF (I_NFLUID .EQ. 6) THEN
   hf(1) = 'oxygen.fld'
ELSE IF (I_NFLUID .EQ. 10) THEN
   hf(1) = 'hydrogen.fld'
ELSE IF (I_NFLUID .EQ. 11) THEN
   hf(1) = 'water.fld'
ELSE
   RETURN
END IF
```

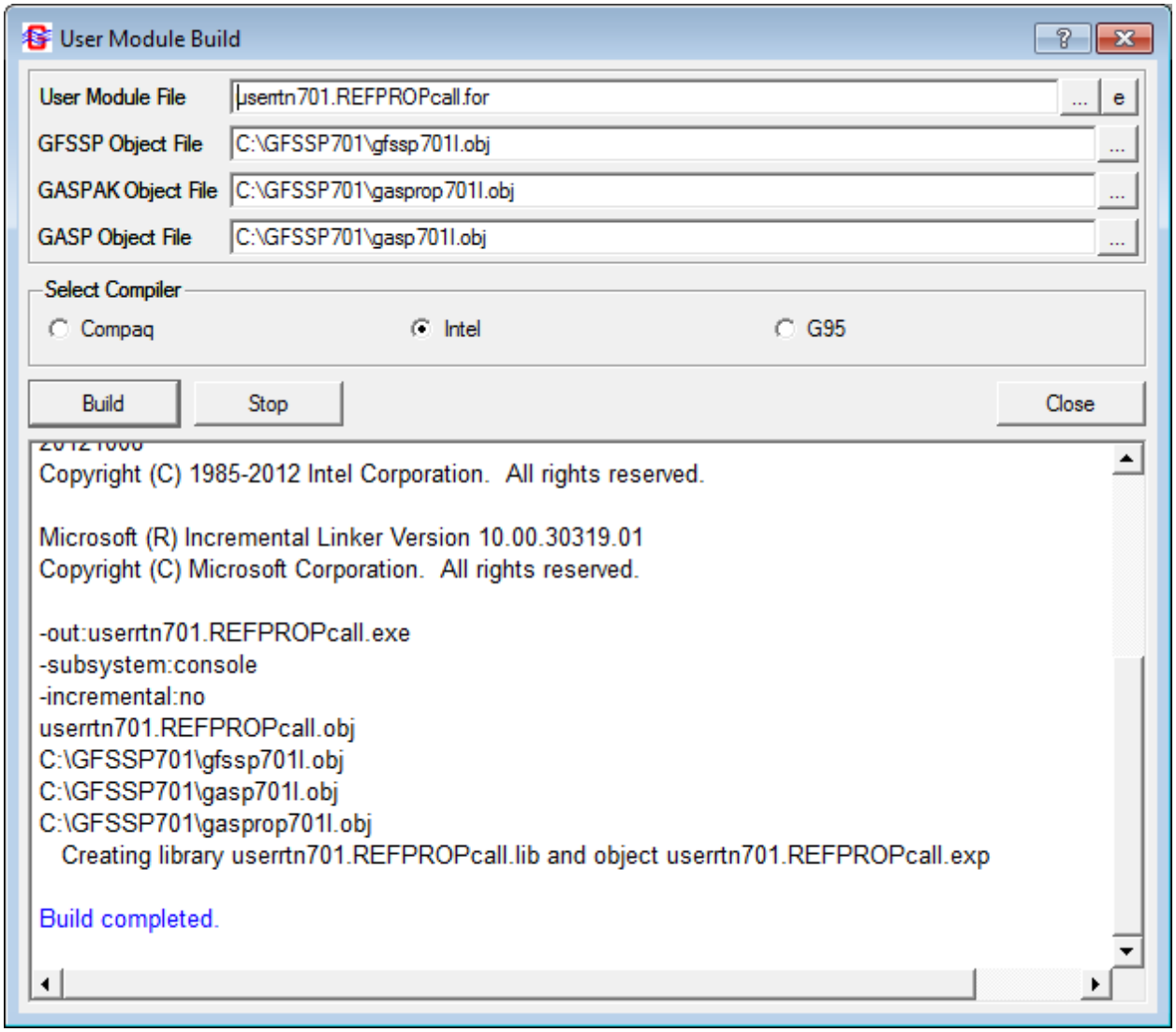
Note that I_NFLUID is the GFSSP ID number of the fluid whose properties will be overwritten. If desired, the above IF…THEN block may be deleted, and the fluid filename simply hard-coded, for example:

```
hf(1) = 'water.fld'
```

3. At the bottom of the user subroutine is the statement to include the "REFPROP.for" file that you created in Part 1.

```
Include 'REFPROP.for'
```

4. In VTASC, click Module/Build. Identify your user subroutine, select your compiler, and click Build. Compiling may take a minute, as there is a lot of REFPROP code being included.
5. On the User Information page, GFSSP's standard executable has been replaced with "userrtn701.REFPROPcall.exe". When the user runs the model with this executable, fluid properties will first be calculated with the built-in property package (such as GASP or WASP), and then overwritten by a call to REFPROP.



GENERAL CONSIDERATIONS WHEN CALLING REFPROP

The table below shows the results of testing several of GFSSP's transient example problems using properties from REFPROP. It is noted that the results agree quite well with those obtained when running with GASP/WASP. However, the REFPROP models run considerably more slowly.

Table 1.5.1: Result and Run Time Comparison for REFPROP Subroutine.

	Results				Model Run Time		
Ex.	Result	GASP	REFPROP	Diff (%)	GASP	REFPROP	REFPROP/ GASP
8	P_{1final} (psia)	38.458	38.455	0.008	0.02s	0.17s	11
10	T_{2final} (°F)	-82.45	-82.29	0.193	0.70s	3m 15.34s	278.27
12	$P_{29final}$ (psia)	52.27	52.53	0.479	23m 18.52s	3h 8m 9.47s	8.07
	$P_{54final}$ (psia)	67.15	66.3	1.226			
14	$F_{12final}$ (lb/s)	0.336	0.3355	0.149	5m 15.56s	28m 29.10s	5.42
15	P_{6max} (psia)	625.8	627.5	0.27	1.31s	14.71s	11.23
16	P_{1final} (psia)	50.65	50.65	0.006	0.02s	1m 24.71s	5430
22tr	P_{2final} (psia)	22.35	22.35	0.01	0.42s	5m 5.61s	725.56
28	$P_{5@4000s}$ (psia)	5.813	5.814	0.021	28m 23.94s	2h 57m 31.62s	6.25

When possible, the user is advised to use the GASP property package during initial model development, as it can save considerable time. It is not uncommon for GFSSP to reach out-of-range enthalpy values during iteration, but REFPROP is not as forgiving as GASP. Users may find that the same model requires more under-relaxation to guard against these out-of-range values when the model is run with REFPROP properties. If this strategy fails, setting limits on pressure and enthalpy through user subroutines PADJUST and HADJUST may also be required.

2.0 Example Problems

In this section several example problems are described. A few of the old example problems have been revised. Revised example problems include: a) Example 4 - Simulation of the Mixing of Combustion Gases and a Cold Gas Stream; b) Example 10 - Pressurization of a Propellant Tank; c) Example 12 – Helium Pressurization of LOX and RP-1 Propellant Tanks; d) Example 29 - Self-Pressurization of a Cryogenic Propellant Tank Due to Boil-off. Two new example problems were added: 1) Example 31 - Modeling Psychrometrics of Air-Water Vapor mixture with GFSSP, 2) Example 32 – Flow Distribution in Manifold. User subroutines were needed to run several example problems. The user subroutine code is provided in the Appendix.

2.1 Example 4 - Simulation of the Mixing of Combustion Gases and a Cold Gas Stream

Problem Considered:

In the previous examples, the fluid systems we considered employed a single fluid. In this example we will consider the simulation of multiple fluids in a mixing process. A mixture

of hot combustion products, consisting of water vapor and carbon dioxide (CO_2), is mixed with cooler CO_2 gas. The mixture temperature and composition are required to be calculated. We will demonstrate how to use the MIXTURE logical option in the code to simulate the mixing of combustion gases and a cold gas stream by utilizing the flow system shown in Figure 6.4.1.

A hot gas mixture consisting of 90 percent water vapor and 10 percent CO_2 (by mass) at 500 psia and 1500 $^{\circ}$F mixes with pure CO_2 at 500 psia and 80 $^{\circ}$F. We want to use GFSSP to predict the flow rate, mixture temperature, and composition of the mixture. We will demonstrate the use of the Temperature and Enthalpy 1 energy equation options for calculating the temperature of a mixture. We will also perform a hand calculation of the mixture temperature and the composition of the mixture to verify GFSSP's predictions.

GFSSP Model:

The mixing chamber shown in Figure 2.1.1 can be simulated with a GFSSP model consisting of four nodes and three branches as shown in Figure 2.1.2. Nodes 1, 2, and 4 are the boundary nodes representing the inlet and outlet of the mixing chamber and node 3 is the internal node representing the mixing chamber. Branches 13 and 23 are represented by option 2 (Restriction) using a flow coefficient of 0.6 and area of 1 in^2. Branch 34 also uses a flow coefficient of 0.6 and area of 1 in^2, but it is modeled using option 22 (Compressible Orifice) to account for the possibility of choked flow in the branch.

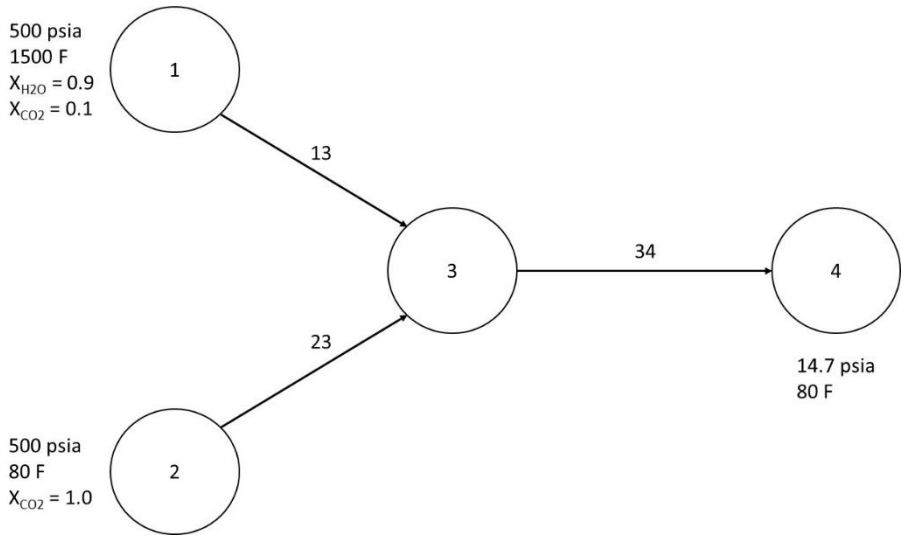


Figure 2.1.1 - Mixing Problem Schematic (Example 4)

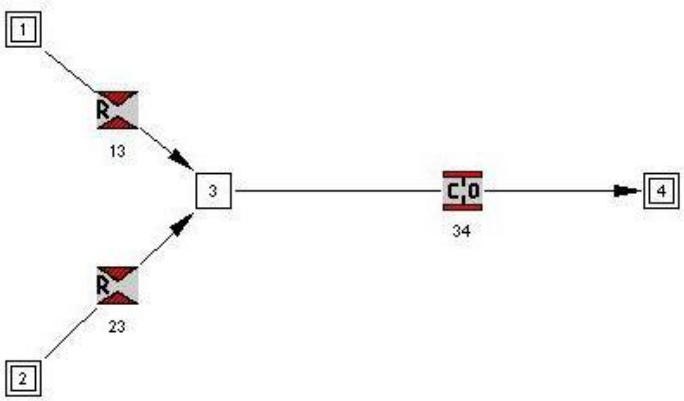


Figure 2.1.2 - GFSSP Model of Mixing Problem

Results:

The input and output files of this example are included in APPENDIX - I as ex4.dat and ex4.out. Ex4.dat uses the Temperature option for the mixture energy equation. The additional file Ex4.Enthalpy1.dat is found in the Examples folder of the GFSSP installation directory and reruns the model using the Enthalpy 1 option for the mixture energy equation. The differences will be discussed below.

With the mixture Temperature option (Ex4.dat), the predicted flow rates in branches 13, 23, and 34 are 1.15, 3.70 and 4.85 lbm/s respectively. The predicted temperature at the outlet of the mixing chamber, node 3, is 674 $^{\circ}$F and the composition is 21.4 % water vapor and 78.6% CO_2 (by mass). The mixture will not vary between node 3 and 4.

Now we would like to verify the predicted results by performing hand calculations of the mixing process. The temperature of the mixture can be calculated from the energy conservation equation for the mixing chamber, written in terms of temperature and specific heat. The energy conservation equation for node 3 can be written as:

$$x_{H_2O}\,\dot{m}_{13}\,c_{p,H_2O}T_1 + x_{CO_2}\,\dot{m}_{13}\,c_{p,CO_2}T_1 + \dot{m}_{23}\,c_{p,CO_2}T_2 = \dot{m}_{34}\,c_{p,mix}T_3 \qquad (2.1.1)$$

The above equation can be rearranged to find T_3.

$$T_3 = \frac{x_{H_2O}\,\dot{m}_{13}\,c_{p,H_2O}T_1 + x_{CO_2}\,\dot{m}_{13}\,c_{p,CO_2}T_1 + \dot{m}_{23}\,c_{p,CO_2}T_2}{\dot{m}_{34}\,c_{p,mix}} \qquad (2.1.2)$$

$$= \frac{(0.9)(1.15\frac{lb}{\sec})(0.570\frac{Btu}{lb-R})(1960R) + (0.1)(1.15\frac{lb}{\sec})(0.302\frac{Btu}{lb-R})(1960R) + (3.70\frac{lb}{\sec})(0.298\frac{Btu}{lb-R})(540R)}{(4.85\frac{lb}{\sec})(0.331\frac{Btu}{lb-R})}$$

= 1134 °R or 674 °F

Equation 2.1.2 calculates the temperature to be 674 ˚F, in agreement with the GFSSP model. The value of 0.331 BTU/lb-R for the specific heat of the mixture is the mass-weighted average of the specific heats of water vapor and CO_2 at 674 ˚F: 0.266 and 0.569 BTU/lb-R, respectively.

Note that Equation 2.1.1 is an approximation that assumes that the change in enthalpy of a fluid can be expressed as the specific heat multiplied by the change in temperature. However, this is strictly correct only if the specific heat is constant over the temperature range of the model. Although this is essentially true for water vapor, the specific heat of CO2 is not quite constant. It varies from 0.298 BTU/lb-R at 80 ˚F, to 0.266 at 674 ˚F, to 0.302 at 1500 ˚F.

To deal with the issue of non-constant specific heat, GFSSP also offers the option of a mixture energy equation written in terms of enthalpy. Once the mixture enthalpy is determined, an iterative scheme is necessary to determine the mixture temperature.

With the mixture Enthalpy 1 option (Ex4.Enthalpy1.dat), the predicted flow rates in branches 13, 23, and 34 are 1.176, 3.776 and 4.952 lbm/s respectively. The predicted temperature at the outlet of the mixing chamber, node 3, is 633 ˚F and the composition is 21.4 % water vapor and 78.6% CO_2 (by mass). These results are similar to those found with the mixture Temperature option, but the mixing chamber temperature is 41 ˚F cooler.

Again we will verify the predicted results by performing hand calculations. The temperature of the mixture can be calculated from the energy conservation equation for the mixing chamber, written in terms of enthalpy. The energy conservation equation for node 3 can be written as:

$$x_{H_2O}\,\dot{m}_{13}\,h_{H_2O} + x_{CO_2}\,\dot{m}_{13}\,h_{CO_2} + \dot{m}_{23}\,h_{CO_2} = \dot{m}_{34}\,h_{mix} \qquad (2.1.3)$$

The above equation can be rearranged to find the mixture enthalpy, h_{mix}.

$$h_{mix} = \frac{x_{H_2O}\,\dot{m}_{13}\,h_{H_2O} + x_{CO_2}\,\dot{m}_{13}\,h_{CO_2} + \dot{m}_{23}\,h_{CO_2}}{\dot{m}_{34}} \qquad (2.1.4)$$

$$= \frac{(0.9)(1.176\frac{lb}{\sec})(1800\frac{Btu}{lb}) + (0.1)(1.176\frac{lb}{\sec})(722\frac{Btu}{lb}) + (3.776\frac{lb}{\sec})(332\frac{Btu}{lb})}{(4.952\frac{lb}{\sec})}$$

= 655 BTU/lb

Having determined the enthalpy of the mixture, GFSSP then iterates until finding the temperature at which the mass-weighted average of the enthalpies is equal to the mixture enthalpy. This occurs at 633 $^{\circ}$F, when the enthalpies of water vapor and CO_2 are 1320 and 474 BTU/lb, respectively.

$$h_{mix} = x_{H_2O} h_{H_2O} + x_{CO_2} h_{CO_2} \qquad (2.1.5)$$

$$= (0.214)(1320 \frac{Btu}{lb}) + (0.786)(474 \frac{Btu}{lb})$$

= 655 BTU/lb

2.2 Example 10 - Pressurization of a Propellant Tank

Problem Considered:

In Example 8, we demonstrated the use of GFSSP's unsteady formulation by predicting the pressure and temperature history during the blow down of a pressurized tank. In this example, we will consider a more complex unsteady process, the pressurization of a propellant tank [25]**. This example will also illustrate the use of User Subroutines to construct a model of mass transfer due to evaporation of propellant to the ullage space.

The tank pressurization option incorporated in GFSSP models the following physical processes:

a. Change in ullage and propellant volume,
b. Change in gravitational head in the tank,
c. Heat transfer from pressurant to propellant,
d. Heat transfer from pressurant to the tank wall,
e. Heat conduction between the pressurant exposed tank surface and the propellant exposed tank surface,
f. Mass transfer from propellant to ullage (with user subroutine).

A schematic of a propellant pressurization system is shown in Figure 2.2.1. It is assumed that initially the ullage space is filled with pressurant at propellant temperature. As the warm pressurant enters the ullage space, it mixes with the cold ullage gas and the temperature of the ullage gas starts to increase due to mixing and compression. Initially, the walls of the tank are also at propellant temperature. Heat transfer from the ullage gas to the propellant and the tank wall and mass transfer from the propellant to the ullage start immediately after the pressurant begins flowing into the tank. Propellant flows from the tank to the engine under the influence of ullage pressure and gravitational head in the tank. In the current model, condensation of propellant vapor has been neglected.

** See Reference Section of GFSSP Users' Manual for this reference

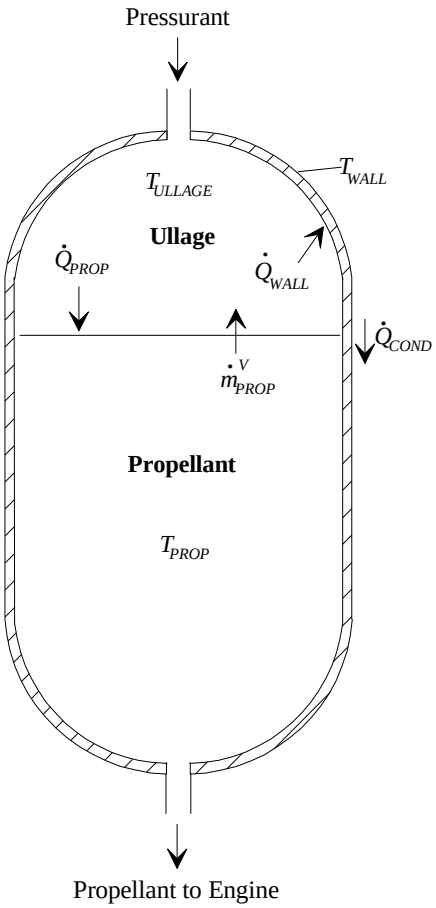


Figure 2.2.1 - Schematic of Propellant Tank Pressurization System

GFSSP Model:

A 5-node pressurization system GFSSP test model, as shown in Figure 2.2.2.a, was developed to test the implementation of the pressurization option. Helium at 95 psia and 120° F enters the ullage space, which is initially filled with helium at 67 psia and -264° F. Node 2 represents the ullage space, which has an initial volume of 25 ft^3. A pseudo boundary node (Node 3) has been introduced to exert ullage pressure on the initial propellant volume of 475 ft^3, which is represented by Node 4. The pressure at the pseudo boundary node is calculated from the ullage pressure and gravitational head and is the driving force to supply the propellant to the engine. This pressure is calculated at the beginning of each time step. Branch 12 models the tank inlet, Branch 34 represents the propellant surface and Branch 45 represents the line to the engine. All three branches were modeled using a Flow through a Restriction (Option 2). The flow coefficient of Branch 45 is adjusted to restrict the propellant flow such that all propellant is expelled from the tank over the course of the run. In this test model, the engine inlet pressure was set at 50 psia. Figure 2.2.2.b shows how the model looks in VTASC. Figure 2.2.3 shows the VTASC tank pressurization dialog and inputs for Example 10.

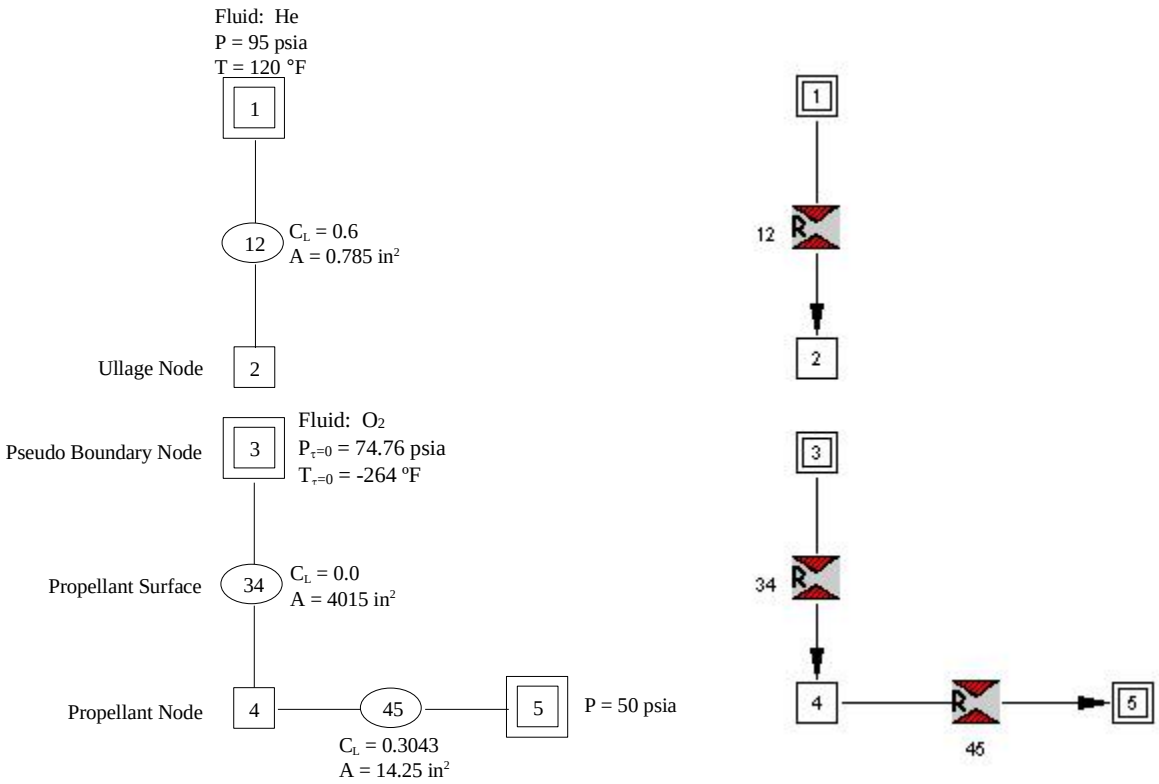


Figure 2.2.2 - Simple Pressurization System Test Model

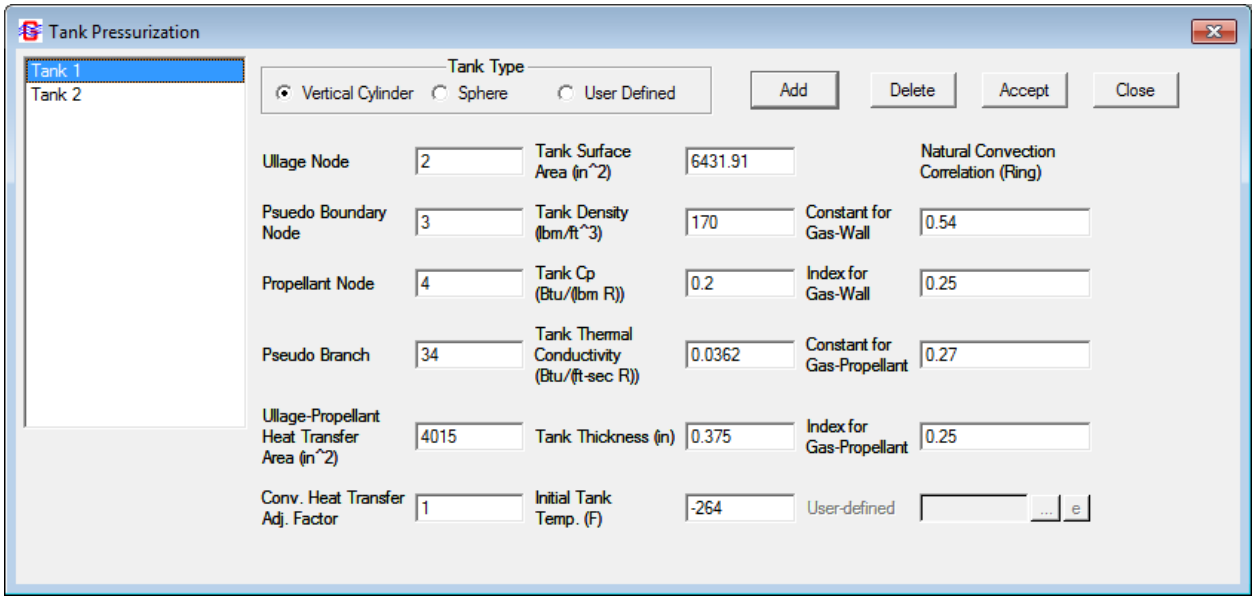


Figure 2.2.3 – Example 10 Tank Pressurization Dialog

User Subroutine

The calculation of mass transfer from propellant to ullage is not a capability that is available with the pressurization option, so in the course of developing this test model separate coding was written to account for the mass transfer. This coding was included in the user routine SORCEC. This routine uses the heat transfer rate from the ullage to the propellant to calculate the mass transfer rate of vaporized propellant to the ullage. It is assumed that the propellant is vaporized from the surface and the heat transfer from the ullage only contributes to the vaporization of propellant. The mass transfer due to vaporization is expressed as:

$$\dot{m}_{prop}^{v} = \frac{\dot{Q}_{prop}}{h_{fg} + c_{pf}\left(T_{sat} - T_{prop}\right)} \qquad (2.2.1)$$

The saturation temperature in Equation 2.2.1 is determined by calling the GFSSP utility subroutine PROPS_PSATX:

```
      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)
```

This subroutine calls GFSSP's property packages at a given saturation pressure Z_P and returns the saturation temperature Z_T, as well as properties of the saturated liquid and vapor. The heat of vaporization h_{fg} is calculated from the saturated liquid and vapor enthalpies.

$$h_{fg} = h_v - h_l \qquad (2.2.2)$$

Subroutine SORCEC

This subroutine is called from subroutine MASSC. The purpose of this subroutine is to calculate the rate of mass transfer of propellant, $\dot{m}_{prop}^{v}$, into the ullage space due to evaporation. The saturation temperature and heat of vaporization are determined by a call to subroutine PROPS_PSATX. SORCECON(IPUL,2) is the source term for oxygen (the second species in the model) in the ullage node and SORCEMAS(IPUL) is the mass source in the ullage node. There is also code to copy the current values of ullage and propellant volume, heat rates, and tank wall temperature to user-defined plot variables.

Subroutine BNDUSER

This subroutine is called at the beginning of each time step. It is used to initialize the user-defined plot variables with names and units for the header of the Winplot file.

Results

The User Subroutine, input and output files including history files of Example 10 have been attached in Appendix - O. The pressurization system transient test model was run for 200 seconds with 0.1-second time step.

Figure 2.2.4 shows both the ullage pressure and tank bottom pressure histories for the test model. After an initial pressure rise due to a "ramping up" transient effect, both pressures begin a slow but steady decline for the remainder of the run. It should be noted that tank bottom pressure was calculated by adding ullage pressure with pressure due to gravitational head. Figure 2.2.4 shows that as the gravitational head decreases, the ullage and tank bottom pressures slowly converge until all propellant is drained from the tank. The slow decline in ullage pressure is mainly due to the expanding ullage volume.

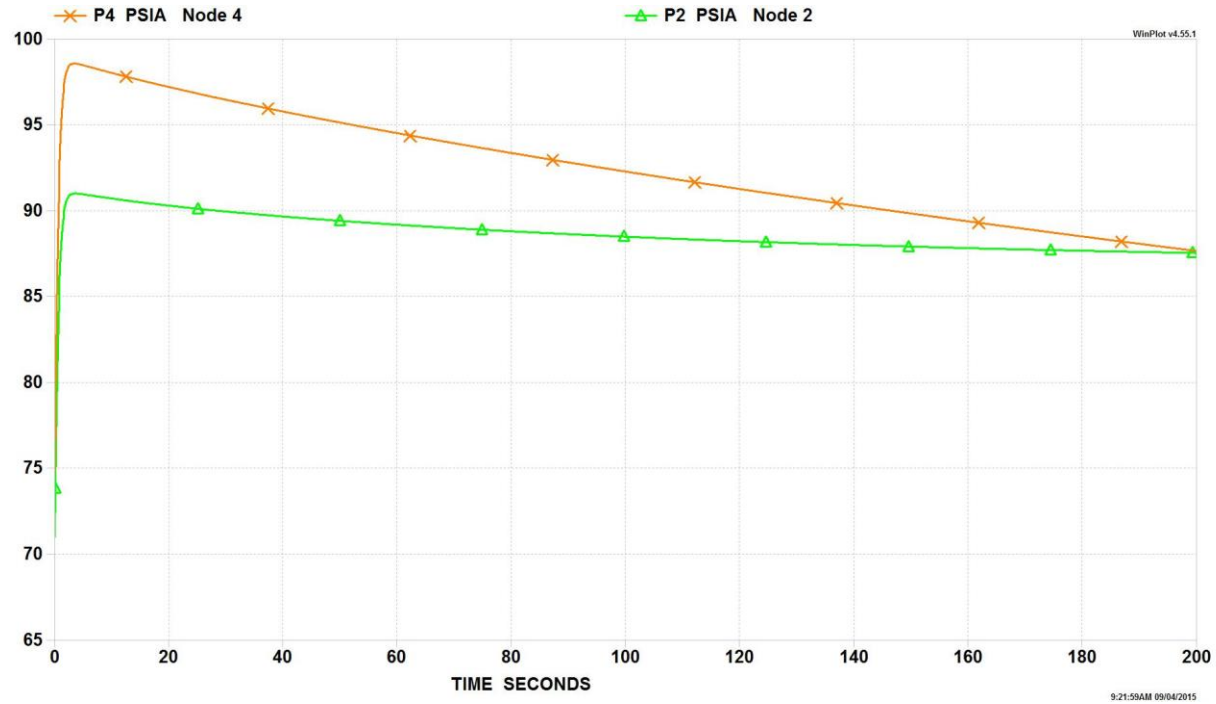


Figure 2.2.4 – Ullage (Green) and Tank Bottom (Orange) Pressure History

Figure 2.2.5 shows the histories for the ullage temperature and the tank wall temperature. This figure shows that the tank wall temperature rises twenty-two degrees over the course of the model run. It reveals that the 120 °F helium gas entering the tank has an increasing effect on the tank wall as propellant is drained from the tank and the wall surface area exposed to the warmer ullage gas grows. This effect is somewhat dampened, however, because the heat gained by the wall is conducted to the portion of the tank that is submerged in LOX, which acts as a heat sink. The ullage temperature rises 210° F

during the first sixty seconds of tank pressurization before beginning a slow decline for the remainder of the simulation. This large initial temperature rise is primarily due to the mixing of hot helium gas with the relatively cold gas present in the ullage. The decline in temperature is a result of expansion due to continuous increase of the ullage volume.

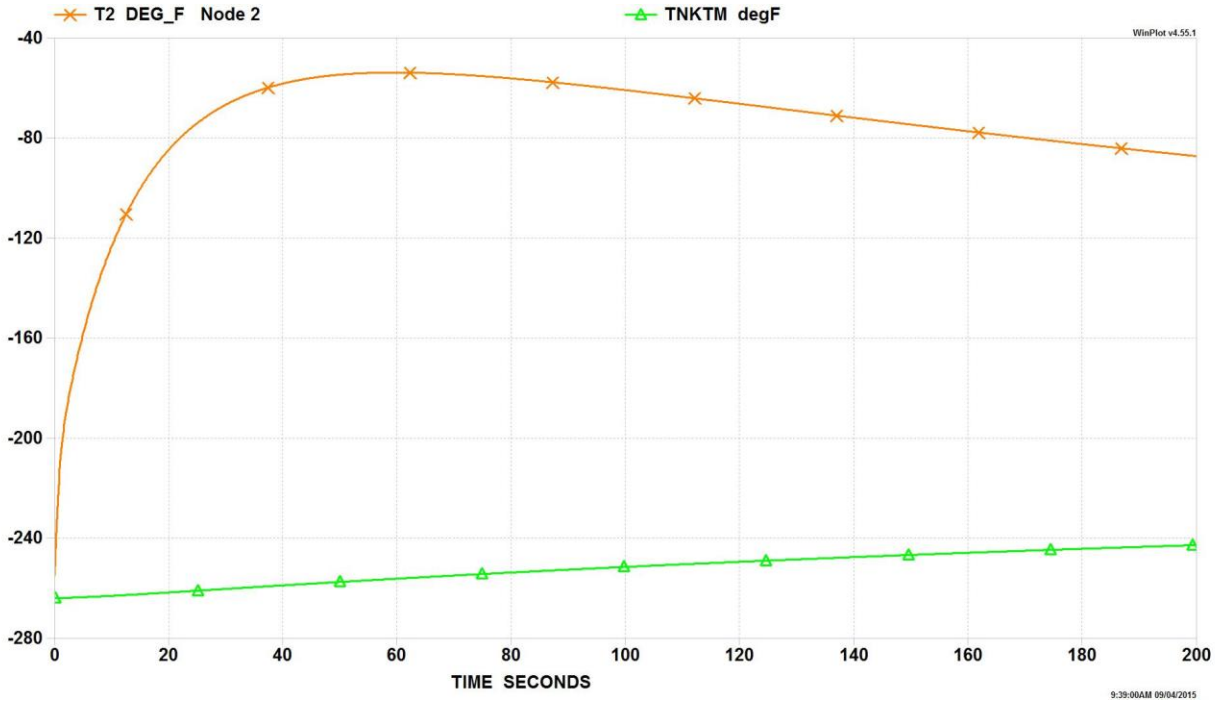


Figure 2.2.5 Ullage (Orange) and Tank Wall (Green) Temperature History

Helium flow rate into the tank is shown in Figure 2.2.6. The helium flow rate was found to drop initially as the start transient takes place, which is consistent with the "ramp up" effect noted in Figure 2.2.4. Then the flow rate begins to gradually increase as ullage pressure drops due to the expanding ullage volume. LOX flow rate into the engine is shown in Figure 2.2.7. The LOX flow rate curve mirrors the ullage and tank bottom pressure curves, rising through an initial start transient to a peak value and then declining for the remainder of the run as tank pressure drops.

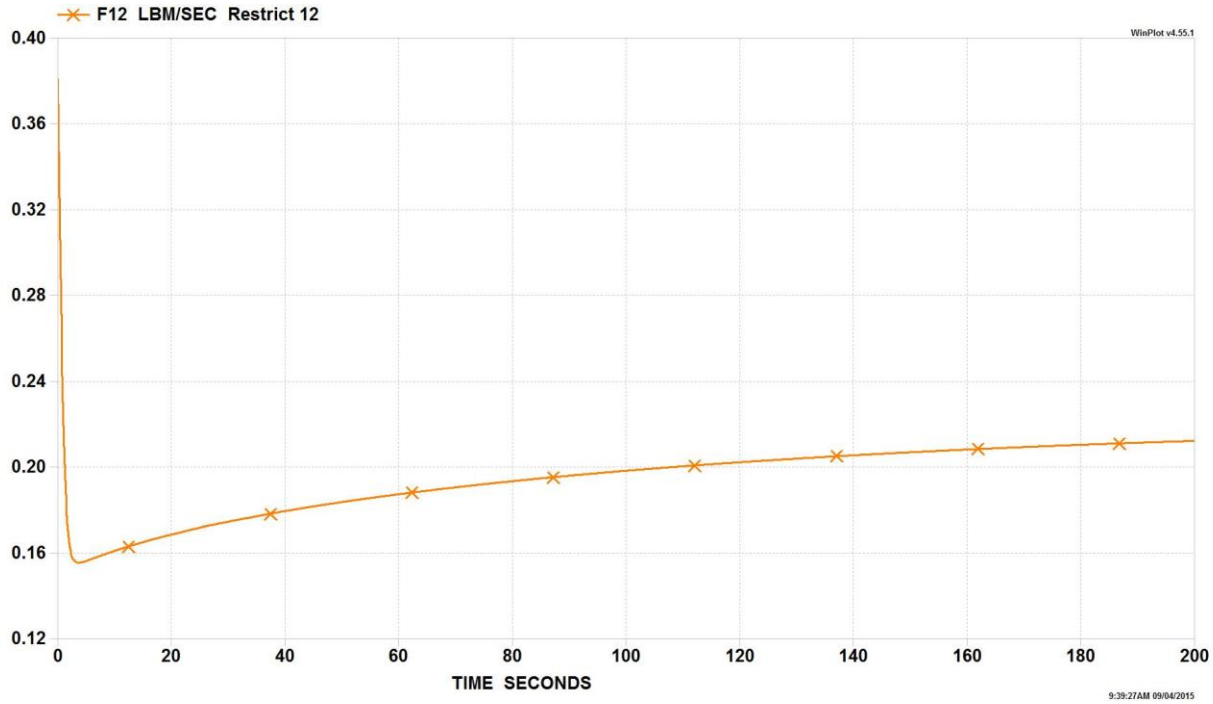


Figure 2.2.6. - Helium Mass Flow Rate History

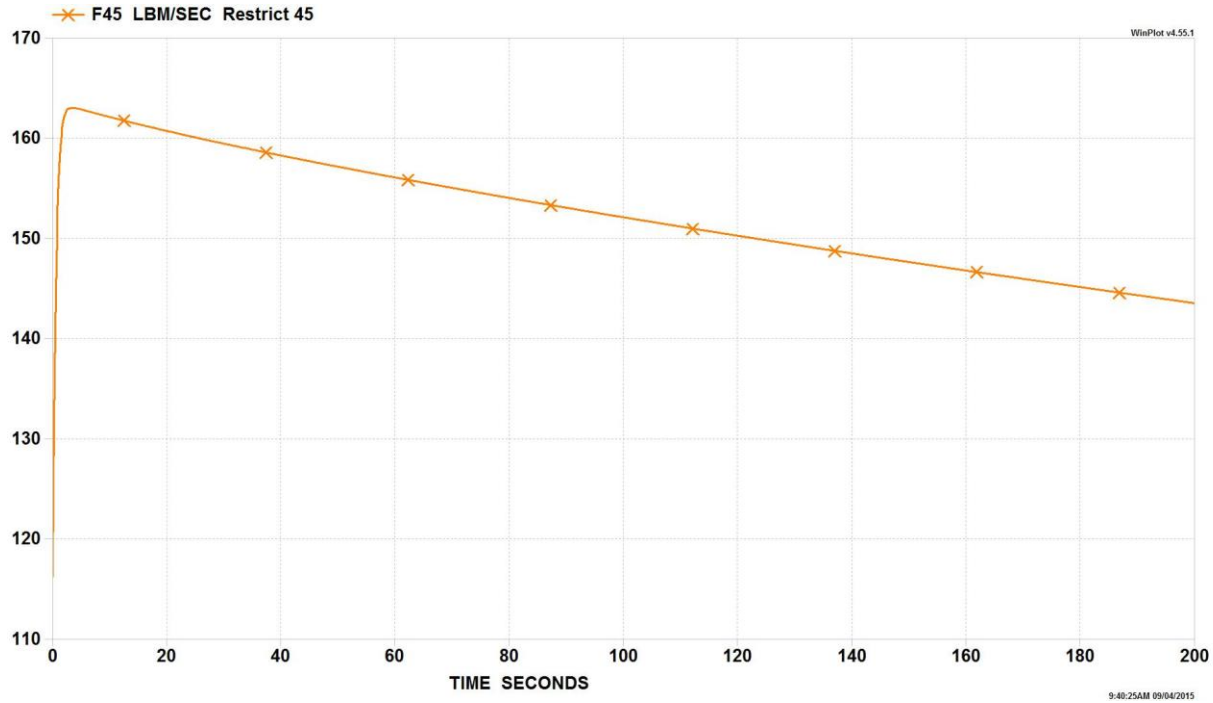


Figure 2.2.7. - LOX Mass Flow Rate History

Figure 2.2.8 shows the mass transfer rate of Gaseous Oxygen (GOX) into the ullage space over the duration of the run. The mass transfer rate curve mirrors the ullage temperature curve, which is what one expects since the mass transfer is based on the

ullage to propellant heat transfer, which is based on ullage temperature. GFSSP predicts a final GOX mass concentration of 0.15 in the ullage.

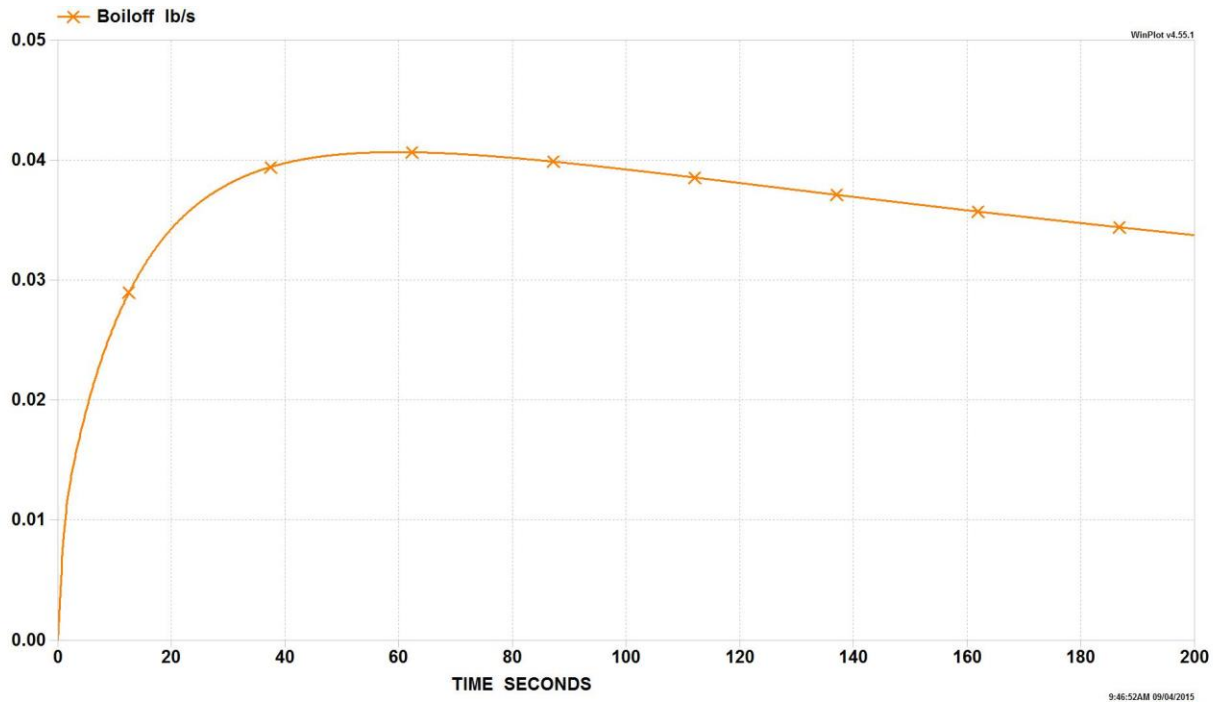


Figure 2.2.8. - GOX Mass Transfer Rate History

As a validation, the model results were compared with a published correlation of pressurant requirements for a given displacement of propellant as published by Epstein and Anderson [40]**. The correlation calculates the collapse factor, which is defined by Van Dresar [41]** as a ratio of the actual pressurant consumption to an ideal pressurant consumption where no heat or mass transfer from the pressurant occurs. This correlation takes the form shown in equations 2.2.3 through 2.2.7.

$$\frac{w_p}{w_p^0} = \left\{\left(\frac{T_0}{T_s} - 1\right)\left[1 - \exp\left(-p_1 C^{p_2}\right)\right] \times \left[1 - \exp\left(-p_3 S^{p_4}\right)\right] + 1\right\} \times \exp\left[-p_5\left(\frac{1}{1+C}\right)^{p_6}\left(\frac{S}{1+S}\right)^{p_7} Q^{p_8}\right] \tag{2.2.3}$$

where

$$w_p^0 = \rho_G^0 \Delta V \tag{2.2.4}$$

$$C = \frac{\left(\rho c_p^0 \delta\right)_{wall}}{\left(\rho c_p\right)_G^0 D_{eq}} \frac{T_s}{T_0} \tag{2.2.5}$$

$$S = \frac{h\theta_T}{\left(\rho c_p\right)_G^0 D_{eq}} \frac{T_s}{T_0} \tag{2.2.6}$$

$$Q = \frac{\dot{q}\theta_T}{\left(\rho c_p\right)_G^0 D_{eq} T_0} \tag{2.2.7}$$

Van Dresar [41][**] later modified this correlation by redefining D_{eq} as shown in equation 2.2.8.

$$D_{eq} = 4\frac{\Delta V}{A_{sw}} \tag{2.2.8}$$

For this validation exercise, the tank is assumed to be cylindrical and therefore the tank diameter is used in place of Van Dresar's equivalent diameter definition. The tank characteristics used are those values utilized in the GFSSP test model. Also, the ideal pressurant properties of $c_{p_G}^0 = 1.24\frac{Btu}{lbm - R}$ and $\rho_G^0 = 0.06087\frac{lbm}{ft^3}$ are found using the helium inlet conditions of P_0 = 95 psia and T_0 = 120 °F. The saturation temperature of LOX (T_s) is taken to be -264 °F. The heat transfer coefficient is calculated to be $h = 8.36e\text{-}04\frac{Btu}{ft^2\text{-}s\text{-}R}$ by taking the average value of the heat transfer coefficients calculated by GFSSP at each time step. The change in propellant volume is the value predicted by the GFSSP test model and ambient heat flux is neglected in this model. The constants p_1 through p_8 are provided by Epstein and Anderson [40] and are shown in Table 2.2.1 below.

Table 2.2.1. - Constants for LOX Propellant

p_1	0.775
p_2	0.209
p_3	3.57
p_4	0.790
p_5	0.755
p_6	0.271
p_7	0.236
p_8	0.895

Solving equations 2.2.4 through 2.2.8 and substituting into equation 2.2.3 gives $\frac{w_p}{w_p^0} = 1.51$. Solving equation 2.2.4 gives $w_p^0 = 28.9\,\text{lbm}$. The GFSSP output file

predicts a required pressurant mass of approximately 42.1 lbm. Dividing this number by the ideal pressurant mass gives a GFSSP predicted collapse factor of 1.46. Therefore the predicted discrepancy of GFSSP with respect to Epstein's method is -3.3%. It is believed that this discrepancy is due mainly to the sensitivity of the pressurization process to the heat transfer coefficient, which is difficult to calculate accurately.

** See Reference Section of GFSSP Users' Manual for this reference

2.3 Example 12 – Helium Pressurization of LOX and RP-1 Propellant Tanks

Problem Considered:

Example 10 illustrates the use of the pressurization option in modeling ullage and propellant conditions in a tank. In this example we will construct an integrated model consisting of two propellant tanks, a flow network for the ullage pressurant supply from a Facility interface, and engine propellant feed lines. The pressurization system of Propulsion Test Article 1(PTA1) consists of a liquid oxygen (LOX) tank and an RP-1 tank that are both pressurized by helium. This configuration is represented in the schematic shown in Figure 2.3.1. The objective of the present example is to develop an integrated mathematical model from the helium supply line to the engine inlet to model sixty seconds of engine operations. The model has three primary functions. They are:

a. To predict the flow rate and pressure distribution of the helium supply line feeding both the LOX and RP-1 tanks,
b. To predict the ullage conditions considering heat transfer between the ullage, propellant and the tank wall, as well as mass transfer from propellant to ullage,
c. To predict the propellant conditions leaving the tank.

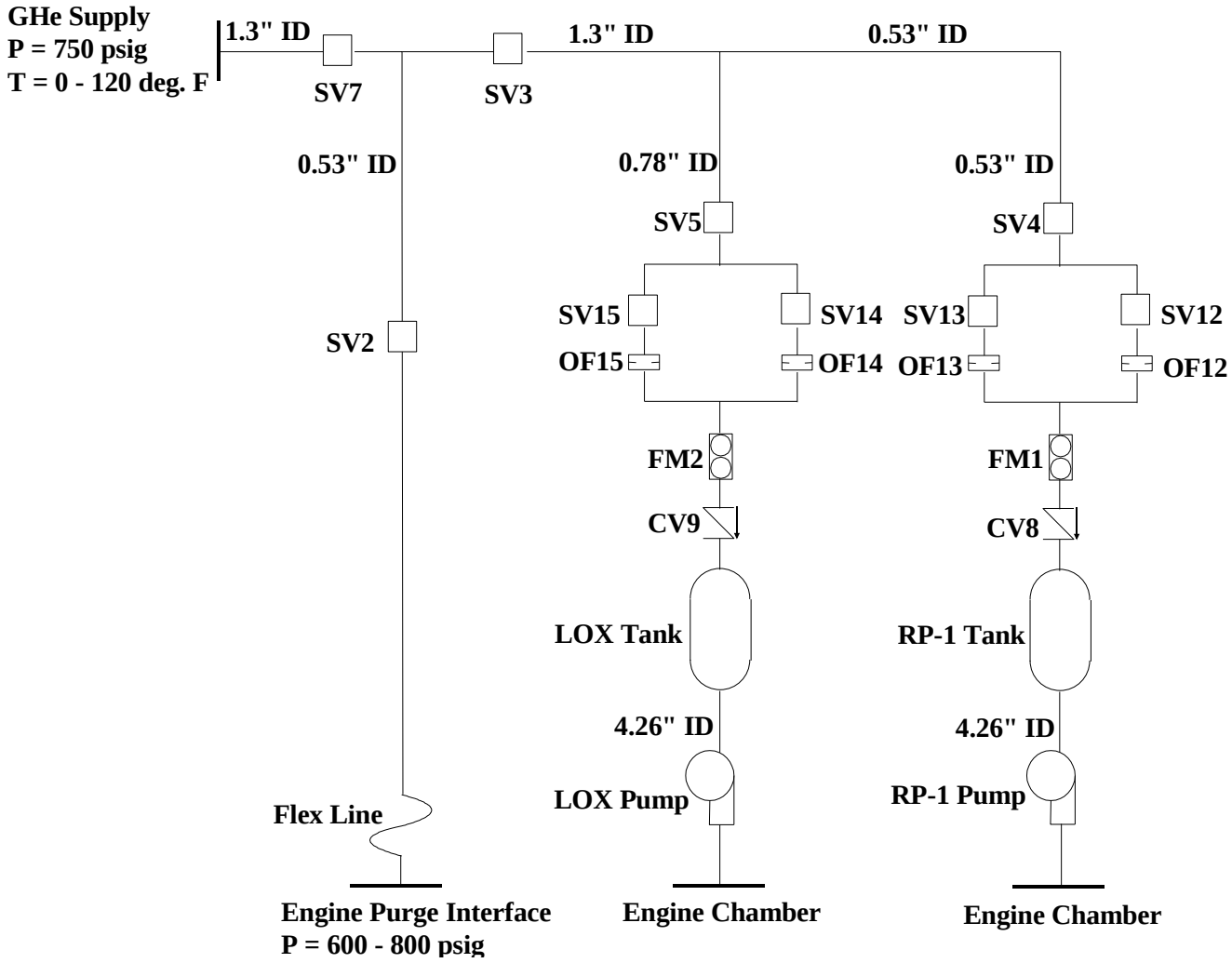
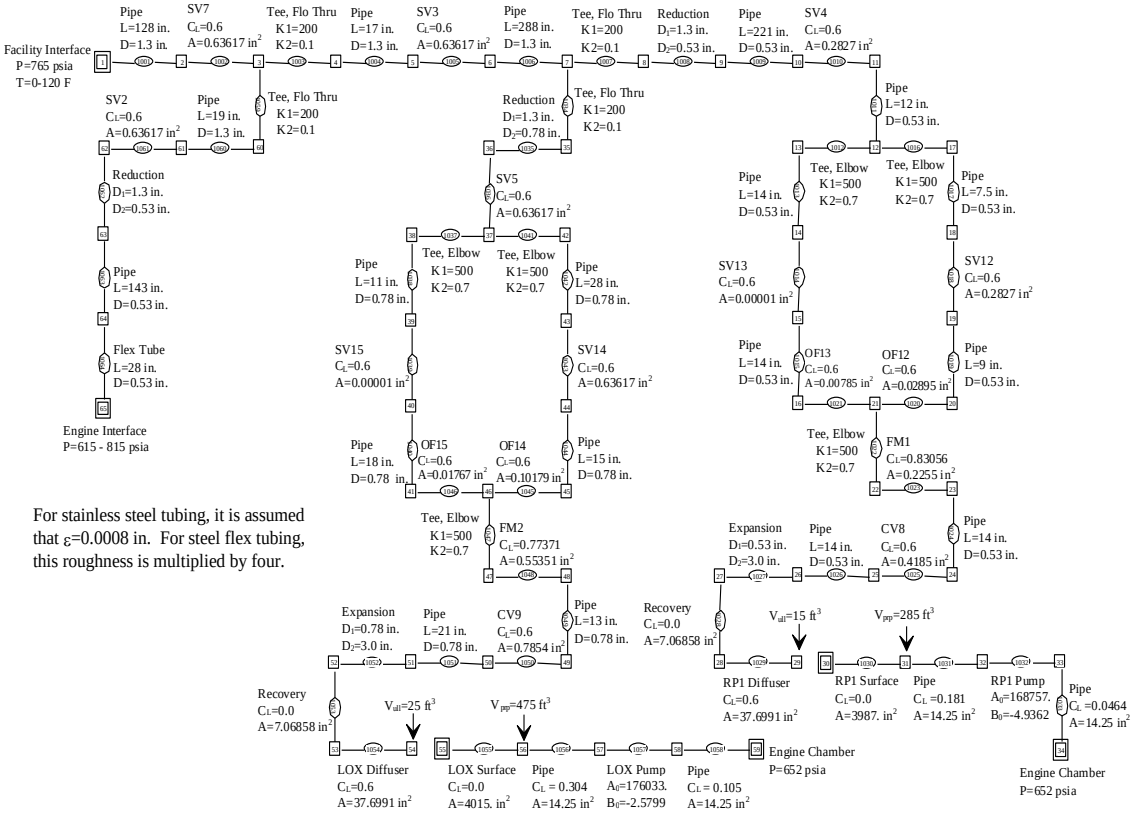


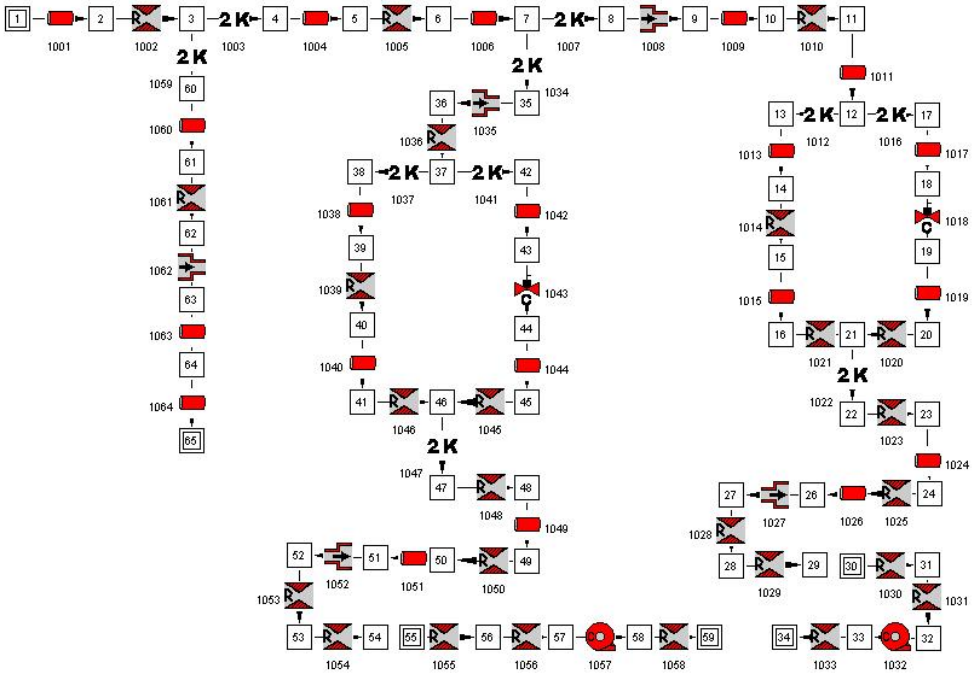
Figure 2.3.1. - Propulsion Test Article 1 Helium Pressurization System Schematic

GFSSP Model:

A GFSSP model of the Helium Pressurization System of LOX and RP-1 Tanks is shown in Figure 2.3.2.a. The model consists of 65 nodes and 64 branches. The model extends from facility interface to engine purge and engine chamber interfaces. It includes all piping and its fittings, orifices and valves. Both RP-1 and Liquid Oxygen (LOX) tanks and pumps are included in the model. Each propellant tank has a diffuser and control valve. Pressure and temperatures are specified at the interfaces, which are represented by six boundary nodes listed in Table 2.3.1.



a) Detailed Model Schematic



b) VTASC Model

Figure 2.3.2 - GFSSP Model of the Pressurization System of Example 12

Table 2.3.1. - Boundary Nodes of Helium Pressurization Flow Circuit

Boundary Node	Interface
1	Facility Interface
65	Engine Interface (Purge)
55	Ullage-propellant interface (LOX Tank)
59	LOX Engine Chamber Interface
30	Ullage-propellant interface (RP-1 Tank)
34	RP-1 Engine Chamber Interface

It may be noted that the nodes representing ullage-propellant interface (Node 55 and 30) are pseudo boundary nodes. The code uses the calculated ullage pressure at the previous time step instead of pressures provided by the user through history files. Helium enters into the system from the facility interface where it is distributed into three parallel branches. The first branching takes place after 128 inches of pipeline and this branch supplies helium to the engine for engine purges. The second branching takes place 305 inches downstream of the first branch and this branch supplies helium to the LOX tank. The rest of the helium goes to pressurize the RP-1 tank. The lines leading to the LOX and RP-1 tanks each have two parallel legs, one of which remains closed during operation. The left leg of the circuit is used to pressurize the tank during pre-pressurization operation while the right leg of the circuit is used to pressurize the tank during pressurization stand-by and engine operations. In the model discussed in this report, setting a high resistance in the appropriate branches eliminated the flow to the left leg. Figure 2.3.2.b shows how this model looks in VTASC.

Results:

The input and output files including history files of Example 12 have been attached in Appendix Q. The GFSSP model shown in Figure 2.3.2 was run for a 60-second engine operation period. At the beginning of the model run, the control valve nominal set points are 72 psia for the LOX tank and 55 psia for the RP-1 tank with plus or minus 3 psi tolerances. After 3 seconds they drop 5 psi to 67 psia for the LOX tank and 50 psia for the RP-1 tank with plus or minus 3 psi tolerances. The output file contains pressure, temperature and density at all nodes as well as flow rate, velocity and pressure drop at all branches for selected time steps.

Figure 2.3.3 shows the predicted pressure history of the RP-1 ullage, RP-1 tank bottom, LOX ullage and LOX tank bottom pressures. The difference in pressure between the tank bottom and ullage is the gravitational head, which slowly reduces as propellant is drained from the tank. The cyclic nature of the pressure profiles is due to the control valves, which are set to close or open as the tank bottom pressures exceed prescribed tolerances. It is observed that the frequency of pressure oscillation is larger in the LOX tank than the RP-1 tank. This observation is attributable to the higher flow rates associated with the LOX tank as compared to those required for the RP-1 tank.

Figure 2.3.4 shows the predicted ullage temperature history in the RP-1 tank. Initially wall and propellant temperatures were assumed equal at 70 °F. Heat transfer between ullage gas and wall is not very significant in the RP-1 tank and as a result the tank wall temperature rises less than a

degree during the 60 second engine operation. Ullage temperature, on the other hand, increases by about 42 °F due to mixing and pressurization. Ullage temperature diminishes during the period of valve closure because of the heat transfer from ullage gas to wall.

Figure 2.3.5 shows the heat transfer history for the RP-1 tank. The ullage to propellant heat transfer rise mirrors the RP-1 ullage temperature behavior, reaching a peak value of 0.3 Btu/s. The ullage to wall heat transfer grows continuously throughout engine operation, achieving a maximum value of 1.8 Btu/s. This continuous rise is due to the ever-increasing tank wall area exposed to ullage gas as propellant is expelled from the tank. Conduction from the ullage exposed tank wall to the wetted wall is negligible compared to the heat transfer between the ullage and the wall.

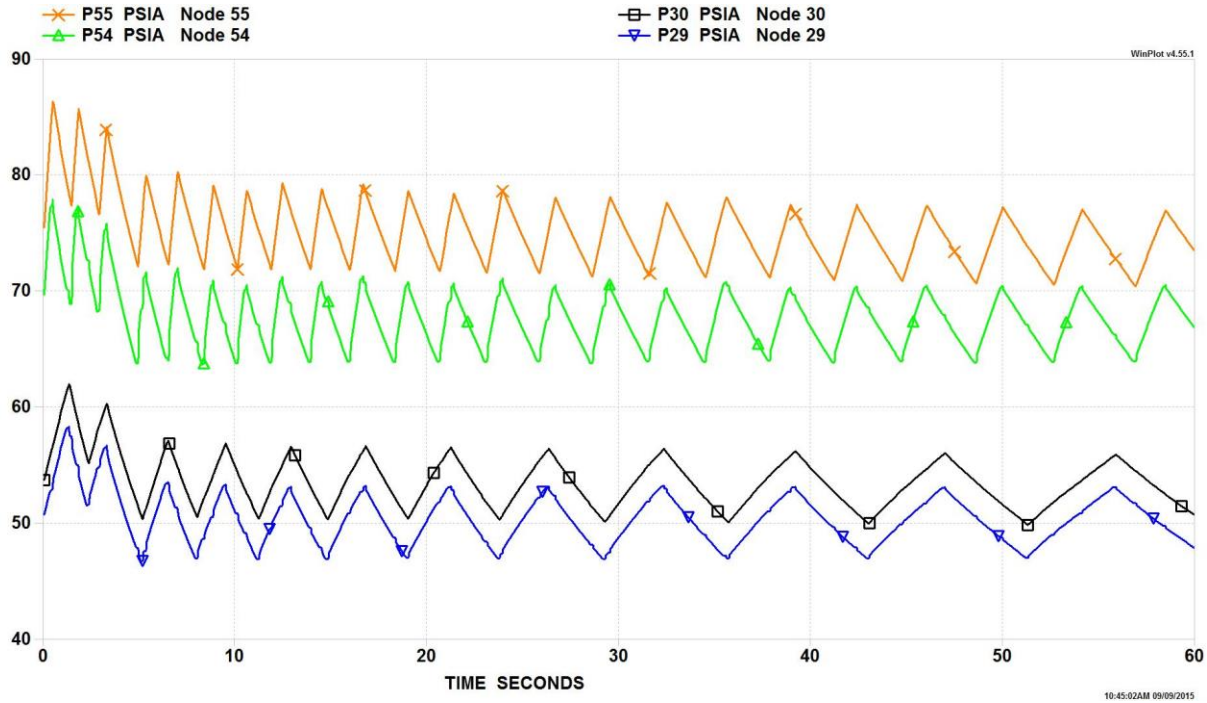


Figure 2.3.3. - Propellant Tank Pressure History: LOx Tank Bottom (Orange), LOx Tank Ullage (Green), RP-1 Tank Bottom (Black), and RP-1 Tank Ullage (Blue)

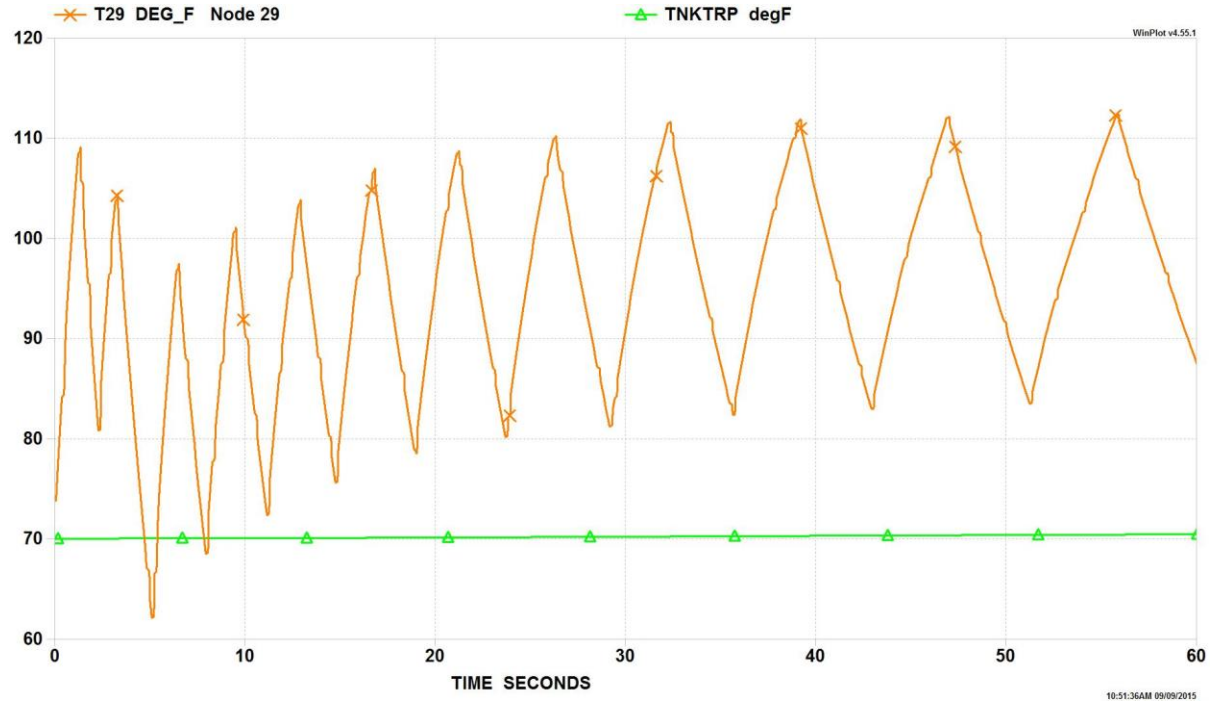


Figure 2.3.4 - RP-1 Temperature History: Ullage (Orange), Tank Wall (Green)

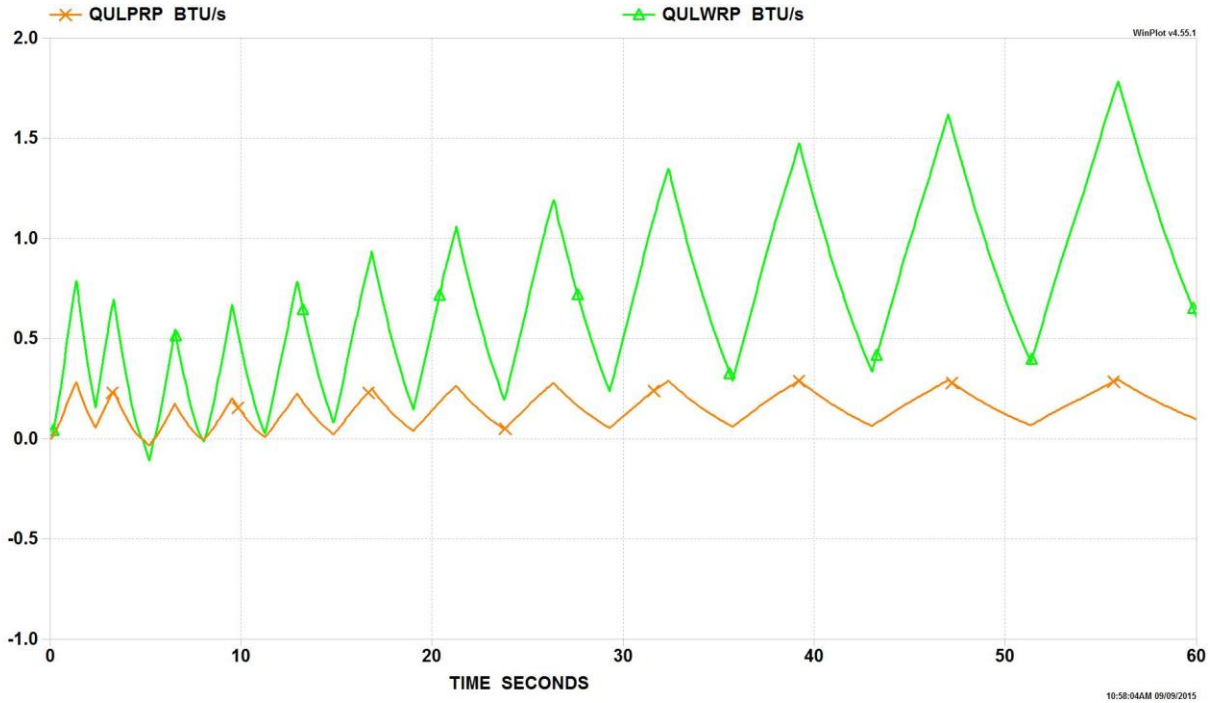


Figure 2.3.5 - RP-1 Heat Transfer History: Ullage to Propellant Surface (Orange), Ullage to Wall (Green)

The predicted ullage temperature history in the LOX tank is shown in Figure 2.3.6. The LOX ullage temperature is assumed to be initially at -260 °F and the tank wall temperature is assumed to be initially at -300 °F. The tank wall temperature rise is more pronounced in the LOX tank than the RP-1 tank, rising 7°F over the course of the 60-second run. The ullage temperature, on the other hand, rises about 173 °F. The higher temperature rise in the LOX tank is primarily due to the fact that the LOX ullage is initially assumed to be at -260 °F and mixes with helium at 120 °F. On the other hand, the initial temperature difference in the RP-1 ullage is much smaller. The other contributing factor is the higher helium flow rate into the LOX tank.

Figure 2.3.7 shows the LOX tank heat transfer history. The LOX heat transfer curves follow a similar pattern to the RP-1 tank heat transfer curves, but on a much greater scale. The ullage to propellant heat transfer achieves a maximum value of 3.1 Btu/s and ullage to wall heat transfer peaks at 26.1 Btu/s.

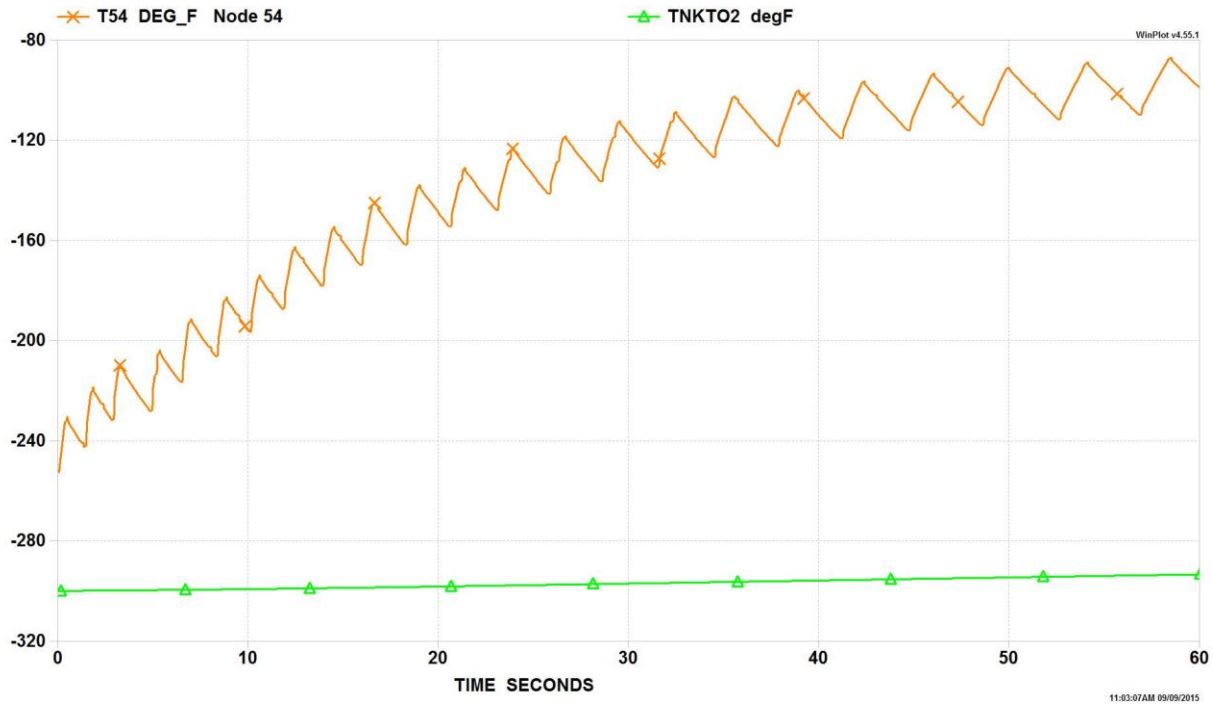


Figure 2.3.6 - LOX Temperature History: Ullage (Orange), Tank Wall (Green)

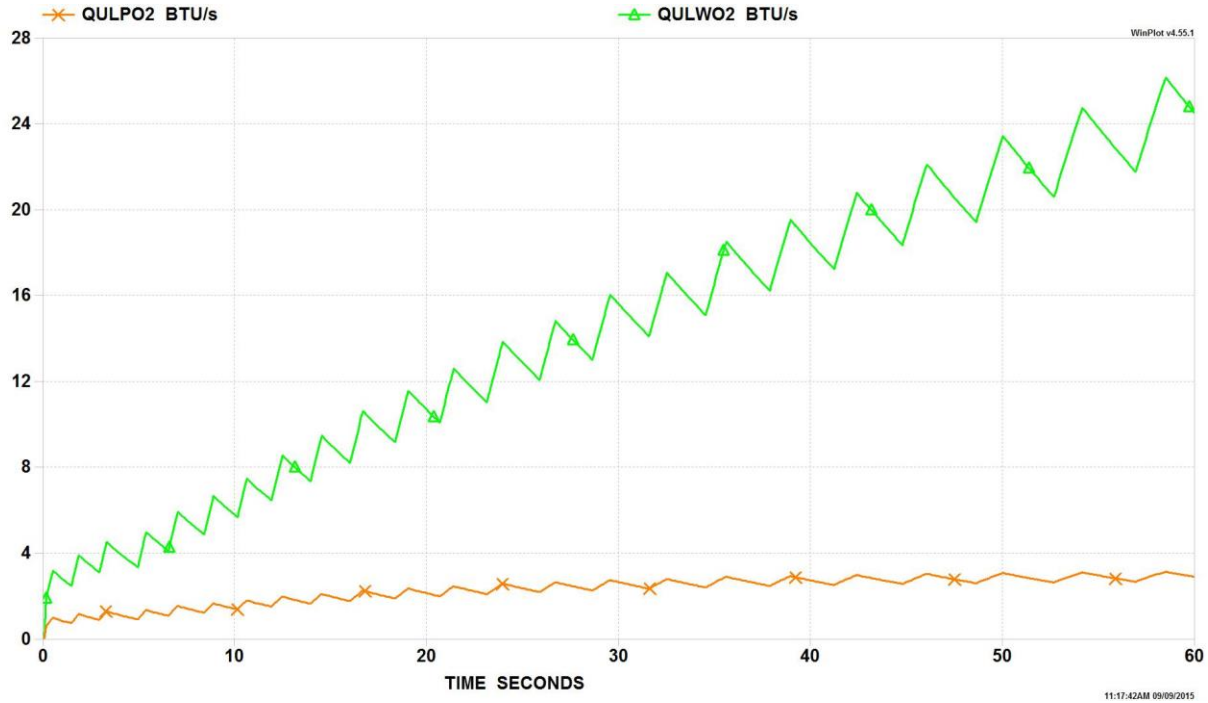


Figure 2.3.7 - LOX Tank Heat Transfer History: Ullage to Propellant Surface (Orange), Ullage to Wall (Green)

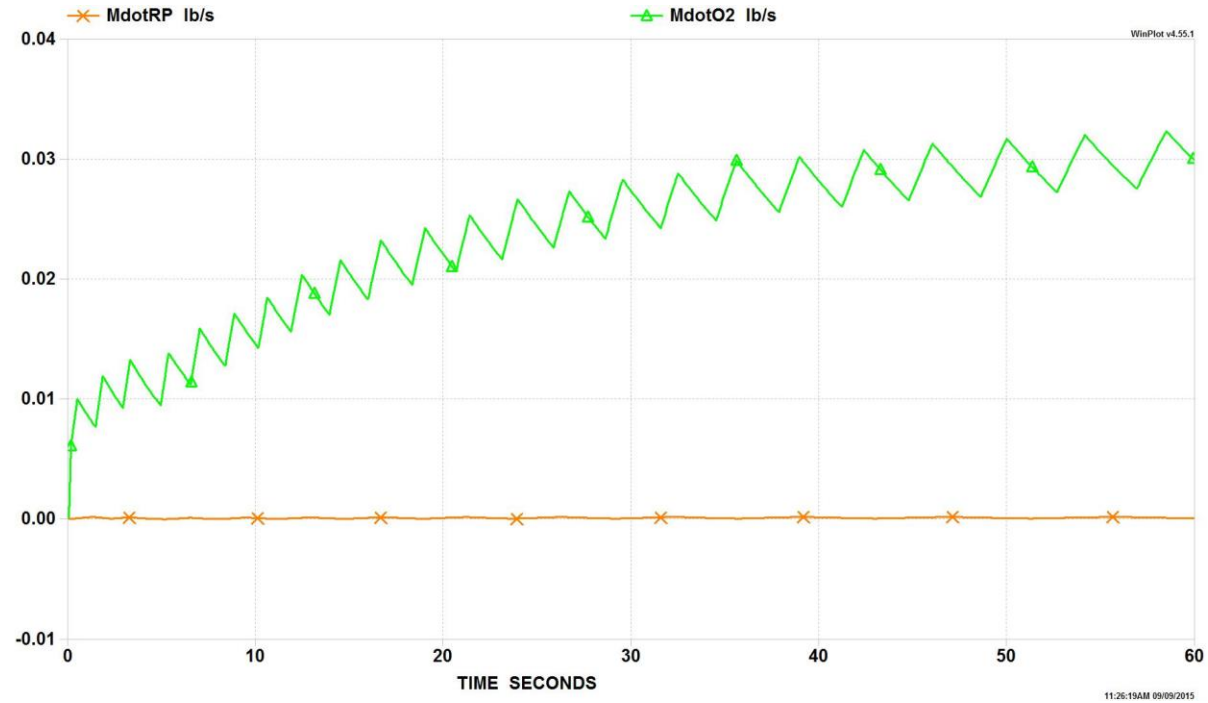


Figure 2.3.8 - Mass Transfer History of Propellant: RP-1 (Orange), Oxygen (Green)

The mass transfer history of propellant into the ullage for the LOX and RP-1 tanks is shown in Figure 2.3.8. The mass transfer of propellant to ullage was calculated using the user subroutines discussed in Section 6.10 (Example 10). The mass transfer rate of GOX into the LOX tank ullage is much larger than that of vaporized RP-1 into the RP-1 tank ullage due to the higher heat transfer rates seen in the LOX tank. At the end of the 60-second run, the mass fraction of GOX in the LOX tank ullage is 0.13 while the mass fraction of vaporized RP-1 in the RP-1 tank ullage is 0.002.

The propellant flow rates are shown in Figure 2.3.9 to be 139 lbm/s for LOX and 64 lbm/s for RP-1. The constant propellant flow rate predictions are a result of the RP-1 and LOX pumps. The branches upstream and downstream of the pumps have been adjusted to reproduce the pressure drops associated with the flow paths between the tanks and pump inlets and pump exits and engine chamber. This was done because of a lack of detailed flow path geometry downstream of the propellant tanks.

While propellant is discharged to the engine, ullage volume increases. The increase in the ullage volume in the RP-1 and LOX tanks is shown in Figure 2.3.10. The initial ullage volume of the RP-1 tank was assumed to be 15 ft^3 while the LOX tank initial ullage volume was assumed to be 25 ft^3. The ullage volumes increase linearly to 90 ft^3 and 141 ft^3 for the RP-1 and LOX tanks respectively.

Figure 2.3.11 shows the helium flow rates in the system. Helium flow rate varies over time due to the opening and closing of the control valves during this time period. The flow from the facility interface is distributed to three branches. A nearly constant flow rate (about 0.4 lbm/sec) is predicted to the engine purge interface for engine purges. The maximum flow rates to the LOX and RP-1 tanks are about 0.34 lbm/sec and 0.085 lbm/sec respectively. Table 2.3.2 shows a comparison of GFSSP helium flow predictions with McRight's [42]** pressurization analysis model.

Table 2.3.2 Comparison between GFSSP and McRight's [42] Helium Flowrates**

GFSSP (lbm/sec)				McRight (lbm/sec)			
Facility	LOX	RP-1	Purge	Facility	LOX	RP-1	Purge
0.825	0.34	0.085	0.4	1.00	0.35	0.1	0.55

The comparison shown in Table 2.3.2 appears reasonable considering the fact that McRight's analysis did not consider pressure loss in lines and fittings and choked flow rate through the orifice was calculated based on a facility pressure of 765 psia. GFSSP calculates pressure drop through the line and therefore the choked flow rate at lower pressure is evidently less than McRight's prediction.

** See Reference Section of GFSSP Users' Manual for this reference

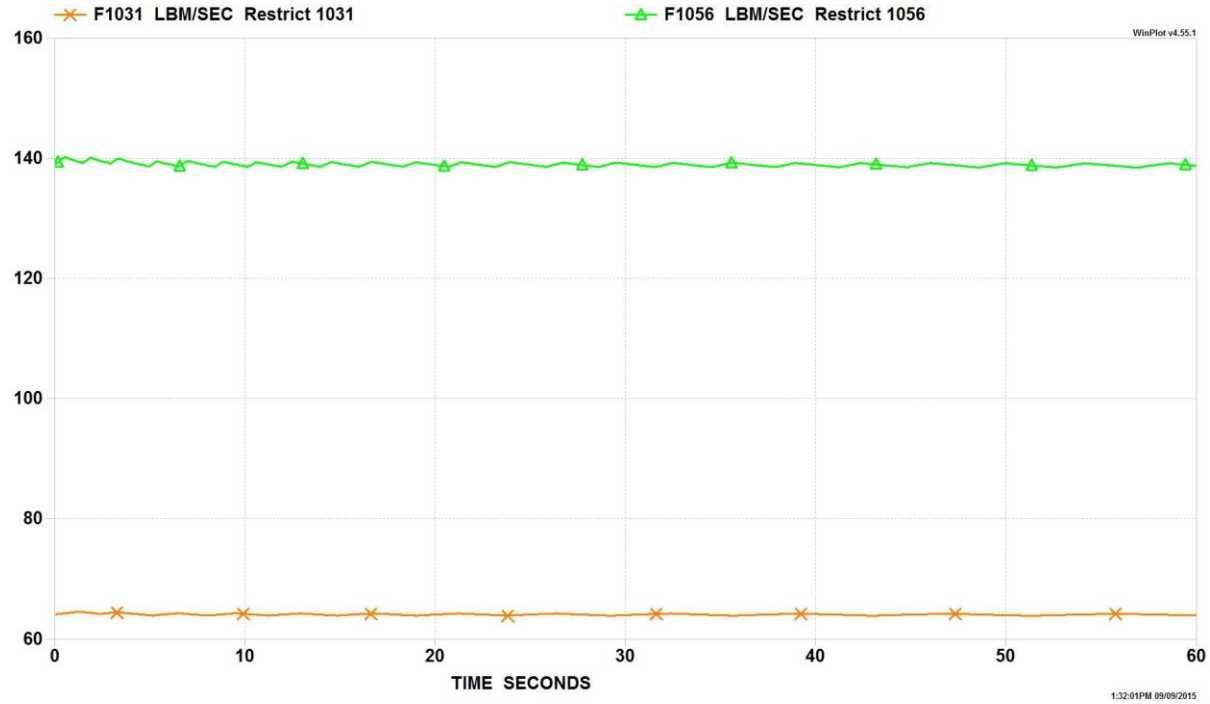


Figure 2.3.9 - Propellant Flowrate History: RP-1 (Orange), LOx (Green)

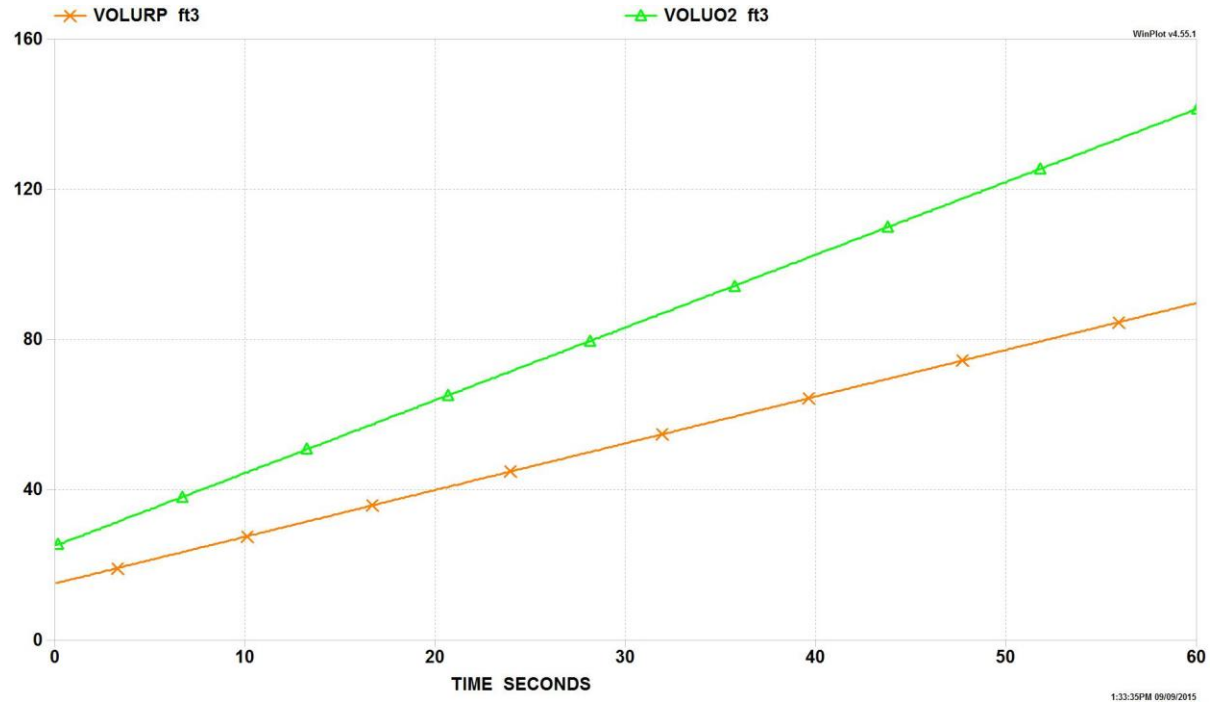


Figure 2.3.10 - Ullage Volume History in Propellant Tanks: RP-1 (Orange), LOx (Green)

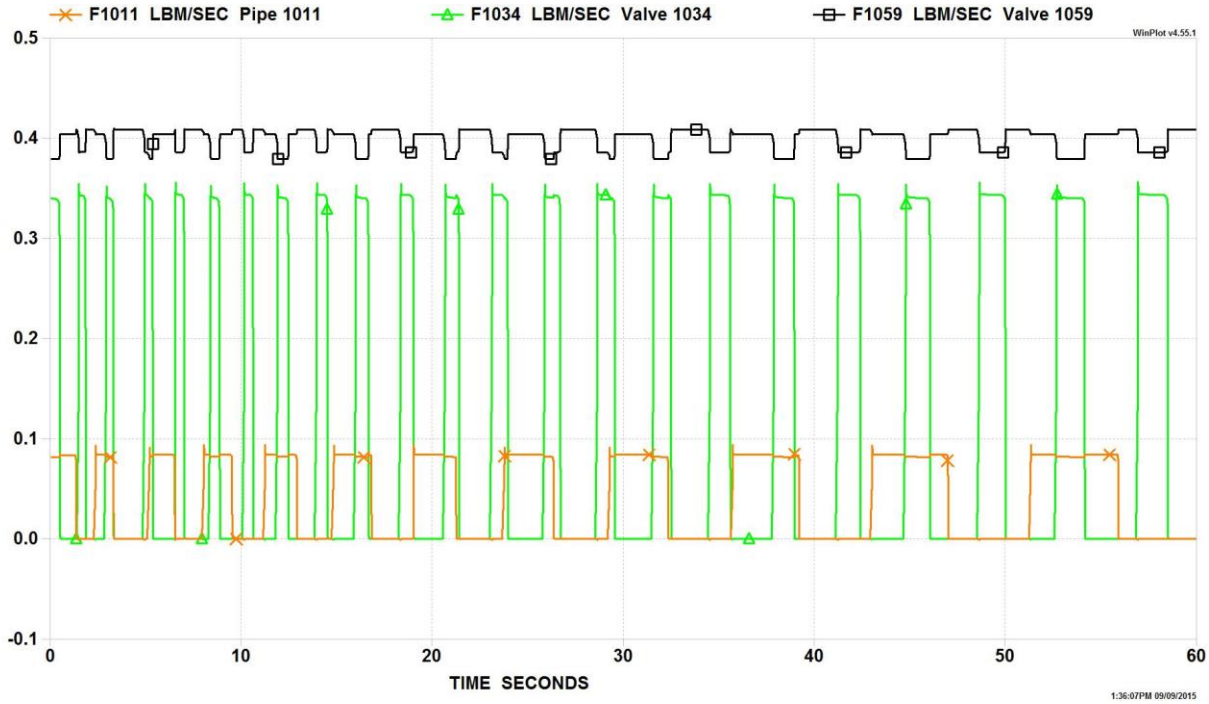


Figure 2.3.11 - Helium Flowrate History: To RP-1 Tank (Orange), To LOx Tank (Green), To Engine (Black)

2.4 Example 29 - Self-Pressurization of a Cryogenic Propellant Tank Due to Boil-off 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 [4] shown in Figure 2.4.1. 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 2.4.2.

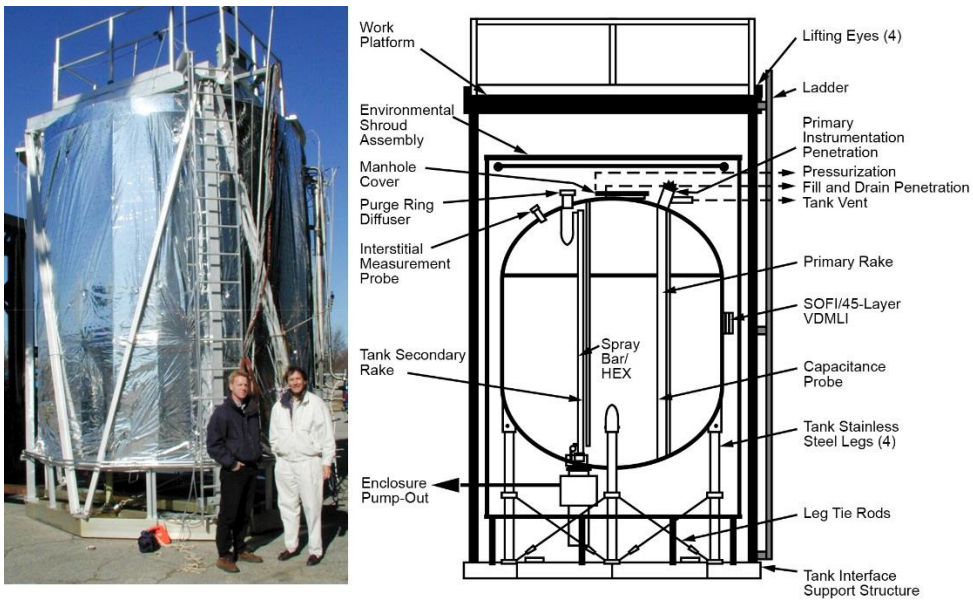


Figure 2.4.1. Multi-Purpose Hydrogen Test Bed (MHTB)

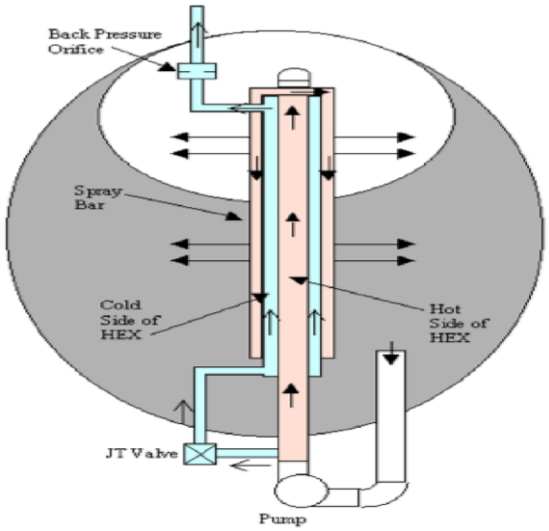


Figure 2.4.2 Thermodynamic Vent System in MHTB Tank

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. The passive thermal control system of the MHTB is comprised of a combination of spray-on foam insulation (SOFI) and a multilayer insulation (MLI) system. The SOFI is a robotically sprayed-on application similar to that used in the Space Shuttle external tank foam application process. It has a nominal thickness of 0.56 inch throughout the surface area of the tank. The MLI is comprised of a 45-layer variable density blanket placed over the SOFI. The blanket is comprised of 0.5 mil double-aluminized Mylar as the radiation shield with B4A Dacron netting as the spacer layer between the Mylar. B4A Dacron bumper strips are used to create the variable density effect where there are fewer layers of MLI closer to the tank. In total, there are three sub-blankets of 10, 15, and 20 layers with a layer density of 8 layers/cm, 12 layers/cm, and 16 layers/cm, respectively.

The active pressure control system used in the MHTB is a spray bar heat exchanger (HEX) thermodynamic vent system (TVS) concept shown in Figure 2.4.2 . It consists of a liquid hydrogen (LH_2) pump, Joule-Thompson (J-T) device, concentric HEX, and longitudinal spray bar system. During the ullage mixing mode, liquid is extracted from the bottom of the tank and fed to the longitudinal HEX via the liquid pump. The fluid is then expelled radially throughout the spray bar to both the ullage and liquid. If the passive thermal protection is sufficient, then the ullage mixing mode is enough to control tank ullage pressures with no propellant loss. When mixing alone cannot control tank pressure, during the TVS operation, a small portion of liquid is fed through the J-T device, expanding the liquid and thus lowering its pressure and temperature. The expanded fluid is then passed through the HEX to condition the mixing fluid portion of the HEX loop and expelled through a pressure control orifice to space.

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. Once the spray starts, a separate GFSSP model of the TVS was run in parallel with the Self-Pressurization model to provide the necessary spray input such as flowrate and temperature of the spray. The TVS model receives a boundary condition of LH_2 pressure and temperature and ullage pressure from the self-pressurization model. The GFSSP model results were then compared with the test data. A 50% fill level case was modeled to simulate the self-pressurization and TVS pressure cycling test.

GFSSP Model

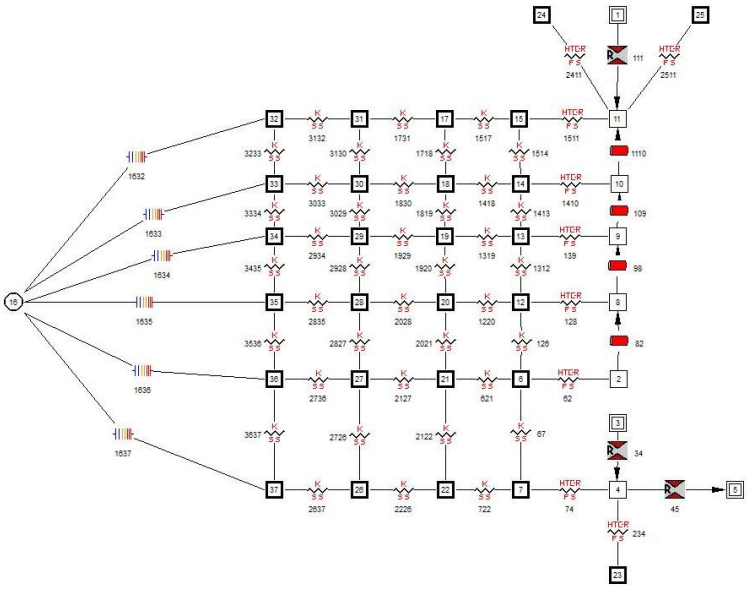


Figure 2.4.3 GFSSP Model of Self-Pressurization

Figure 2.4.3 shows the GFSSP model of 1 g (settled) self-pressurization in the MHTB tank at the 50% fill level. Node 4 represents LH_2; nodes 2, 8, 9, 10, and 11 represent the ullage at different fill levels (54%, 65%, 80%, 92%, and 98%). Node 3 is a pseudo-boundary node separating LH_2 from vapor hydrogen in the ullage space. Each fluid node is connected with a solid node through a solid-fluid conductor. There are four layers of solid nodes representing the Al wall and SOFI. For example, nodes 7 and 22 represent the Al wall while nodes 26 and 37 represent SOFI. MLI was wrapped around the SOFI. Node 37 is connected to ambient node with MLI Conductor, 1637. It may be noted that MLI conductor is a new addition to Version 701. Figure 2.4.4 shows the properties of the MLI used in the MHTB Tank.

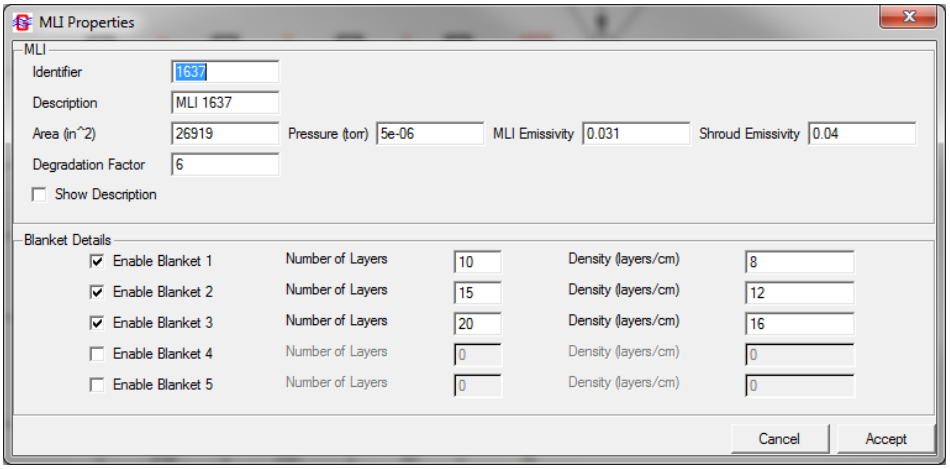


Figure 2.4.4 Properties of MLI Conductor

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.

Evaporative Mass Transfer at Liquid-Vapor Interface

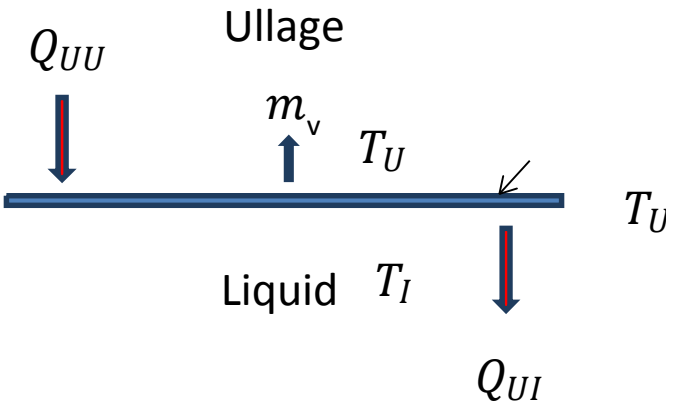


Figure 2.4.5 Schematic of Evaporative Mass Transfer Process at Liquid-Vapor Interface

Figure 2.4.5 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.

Heat transfer from ullage to interface layer:

$$Q_{UI} = h_{UI}A(T_U - T_I) \qquad (2.4.1)$$

Heat transfer from interface to liquid:

$$Q_{IL} = h_{IL}A(T_I - T_L) \qquad (2.4.2)$$

The evaporative mass transfer is expressed as $\dot{m} = \dfrac{Q_{UI} - Q_{IL}}{h_{fg}}$.

h_{fg} is the enthalpy of evaporation, and the heat transfer coefficients h_{UI} and h_{IL} are computed from natural convection correlations given by:

$$\dot{m} = \frac{Q_{UI} - Q_{IL}}{h_{fg}} \qquad (2.4.3)$$

$$h_{UI} = K_H C \frac{k_f}{L_s} \mathrm{Ra}^n = h_{IL} \,. \qquad (2.4.5)$$

Heat Transfer Coefficient Correlation

The heat transfer coefficient between the wall and ullage was computed from a natural convection correlation for a vertical plate [5]. The following set of equations was used for this correlation:

$$\mathrm{Nu} = [(\mathrm{Nu}_l)^m + (\mathrm{Nu}_t)^m]^{1/m} \quad m = 6\,, \tag{2.4.6}$$

$$\mathrm{Nu}_t = C_t^V \,\mathrm{Ra}^{1/3}/(1 + 1.4\ ´\ 10^9\ \mathrm{Pr}/\mathrm{Ra})\,, \tag{2.4.7}$$

$$\mathrm{Nu}_l = 2/\ln(1 + 2/\mathrm{Nu}^T)\,, \tag{2.4.8}$$

$$\mathrm{Nu}^T = \bar{C}_l \mathrm{Ra}^{1/4}\,, \tag{2.4.9}$$

$$C_t^V = \frac{0.13\,\mathrm{Pr}^{0.22}}{\left(1 + 0.61\,\mathrm{Pr}^{0.81}\right)^{0.42}}\,, \tag{2.4.10}$$

$$\bar{C}_l = \frac{0.671}{[1 + (0.492/Pr)^{9/16}]^{4/9}} \tag{2.4.11}$$

where $\mathrm{Gr} = \mathrm{L}^3\rho^2 g\beta\Delta T/\mu^2$; $\mathrm{Pr} = C_p\mu/k$; $\mathrm{Ra} = \mathrm{GrPr}$, and $\mathrm{Nu} = hL/k$.

Thermodynamic Vent System (TVS) Model

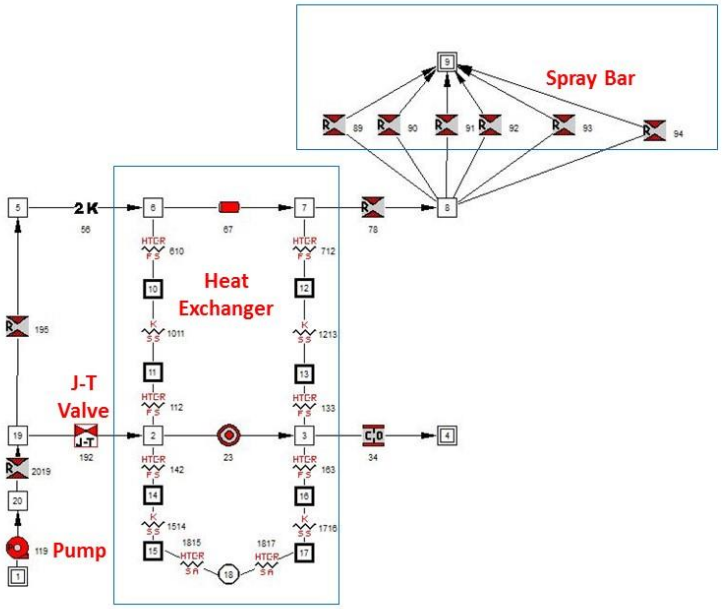


Figure 2.4.6 GFSSP Model of Thermodynamic Vent System (TVS)

GFSSP model of TVS is shown in Figure 2.4.6. The main component of the TVS model is Pump, Joule Thompson Valve, Heat Exchanger and Spray Bar. A small portion of liquid

hydrogen is sent through J-T valve to produce a colder stream of two-phase mixture that cools the bulk of the liquid that is pumped to spray bar.

Integrated Self-Pressurization and Thermodynamic Vent System Model

The integrated model is shown in Figure 2.4.7. The integration of two models is done through an exchange of boundary conditions by writing and reading model output data from files generated while the model is running. The Self-Pressurization model on the left is the driver model. When ullage pressure reaches the maximum allowable pressure, it makes a call to run the TVS model shown on the right. Details of the integration process are shown in Figure 2.4.8.

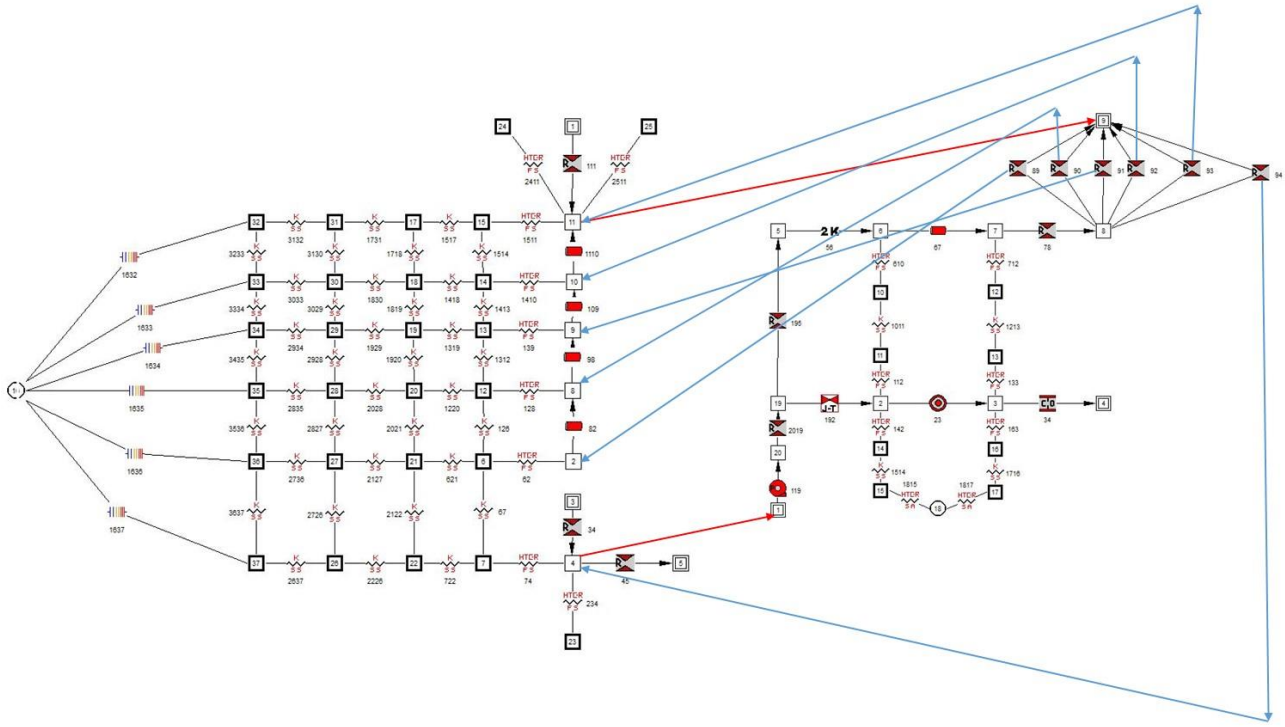


Figure 2.4.7. Integrated Self-Pressurization and TVS models.

Figure 2.4.9 shows the comparison between GFSSP predictions (in green) and the MHTB Test Data (in orange). GFSSP predictions of pressure are shown for a Degradation Factor of 6.0. The Degradation Factor is defined in Equation 2.2 to represent the degradation of performance of the MLI.

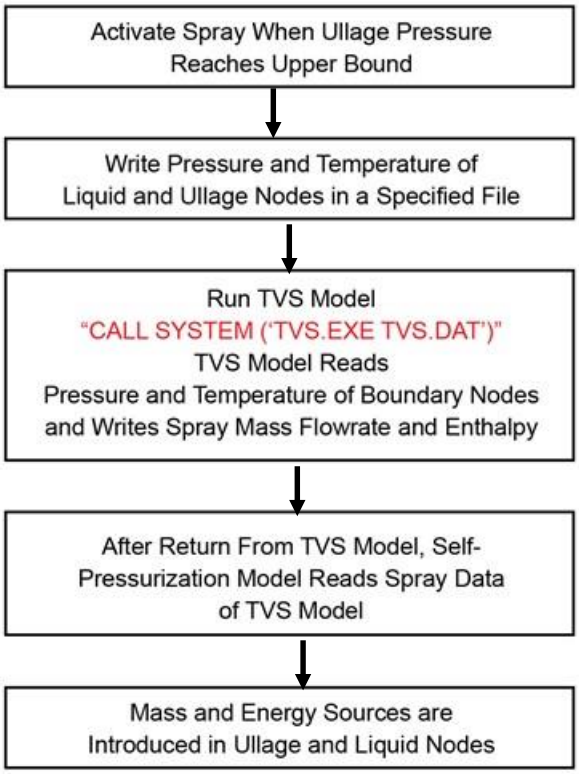


Figure 2.4.8. The main steps of model integration.

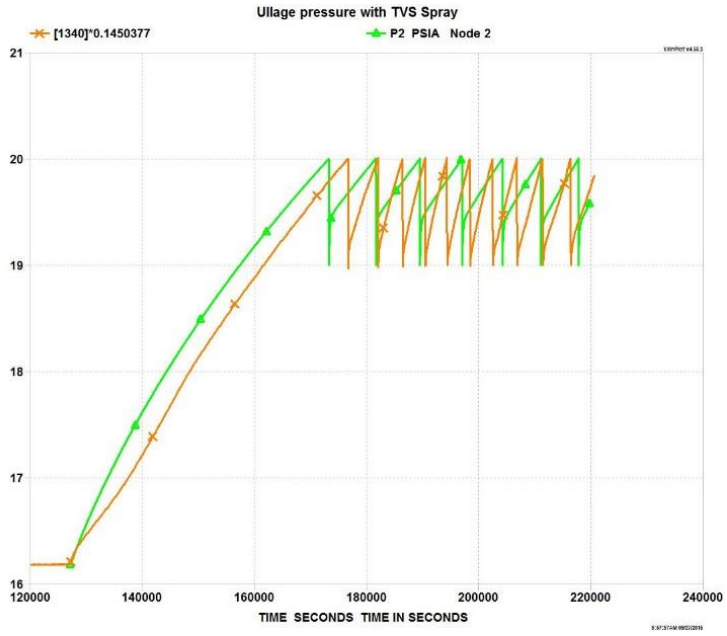


Figure 2.4.9 Application Results for MHTB Self Pressurization Model

2.5 Example 31 - Modeling Psychrometrics of Air-Water Vapor mixture with GFSSP

This example demonstrates the use of GFSSP's built-in psychrometric properties to model an air-conditioning problem. Cold and dry air enters into an air-conditioning system where air is first heated and then humidified by injecting steam as shown in Figure 2.5.1. The purpose of the GFSSP model is to calculate the temperature and relative humidity of the air at the exit of the air conditioner.

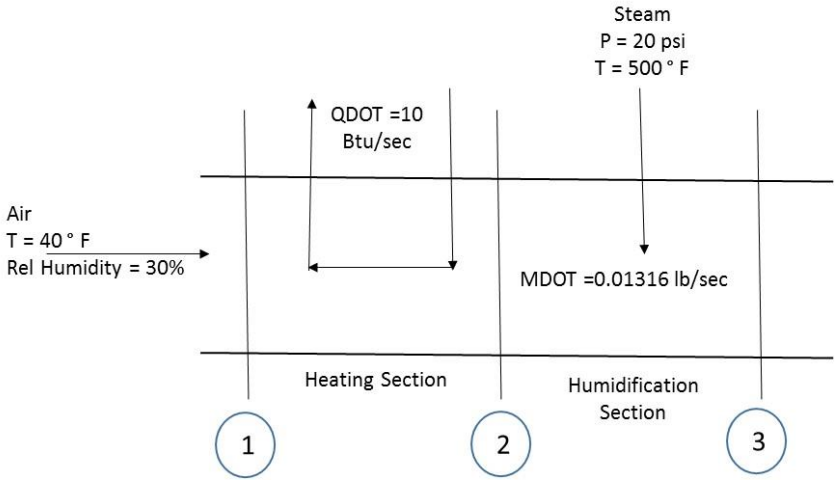


Figure 2.5.1 Air-conditioning problem with heating and humidification

GFSSP model is shown in Figure 2.5.2 consists of heating section (Node 2) and humidifying section (Node 4)

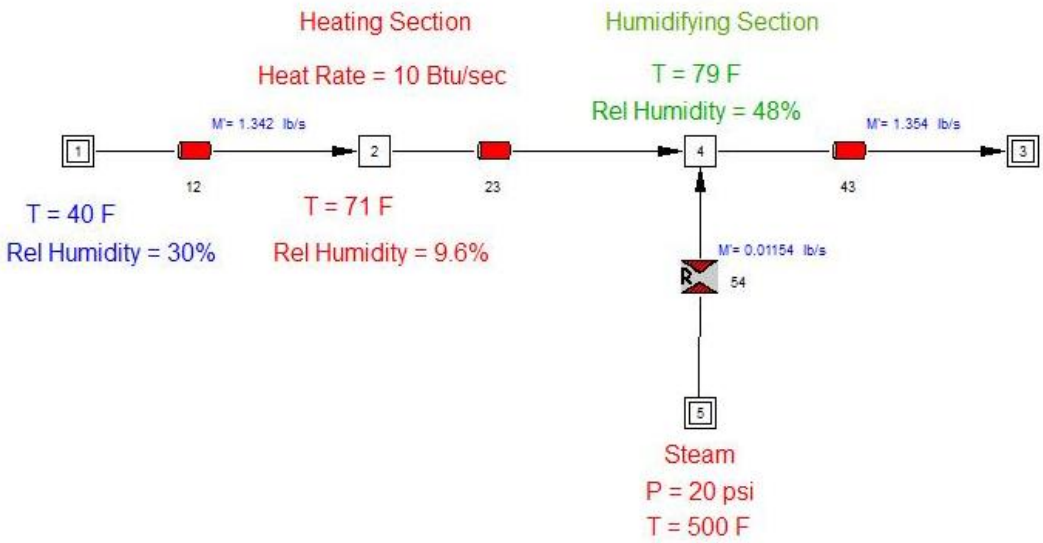


Figure 2.5.2 GFSSP Model of Air-conditioning System

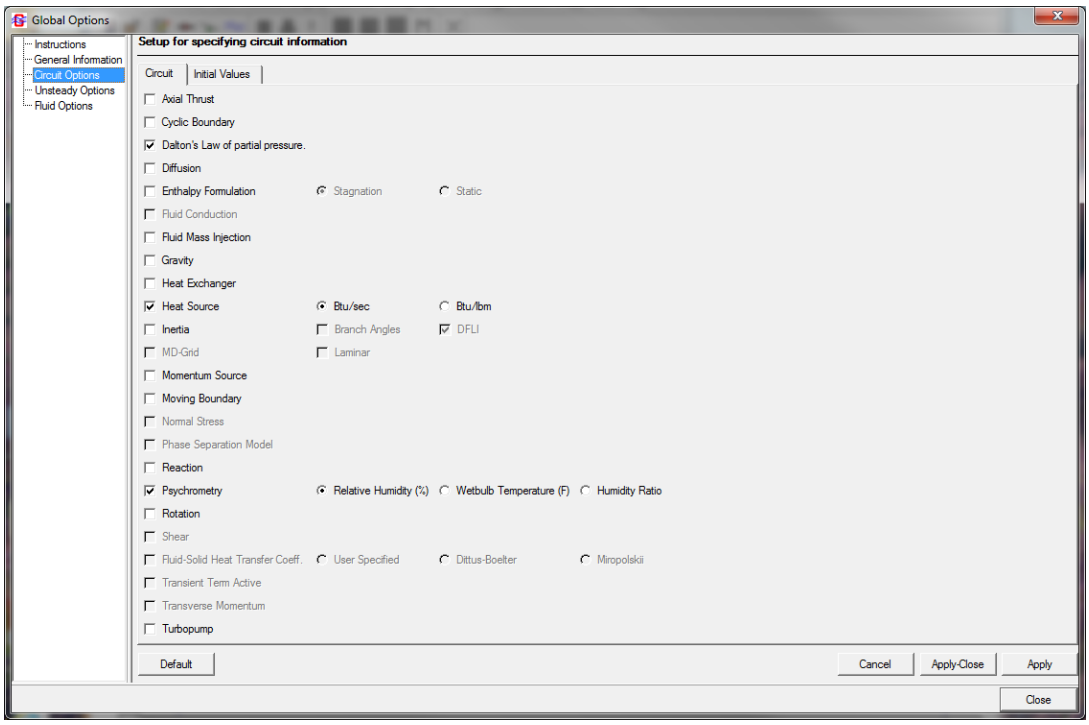


Figure 2.5.3 Activation of Psychrometry in Circuit Option

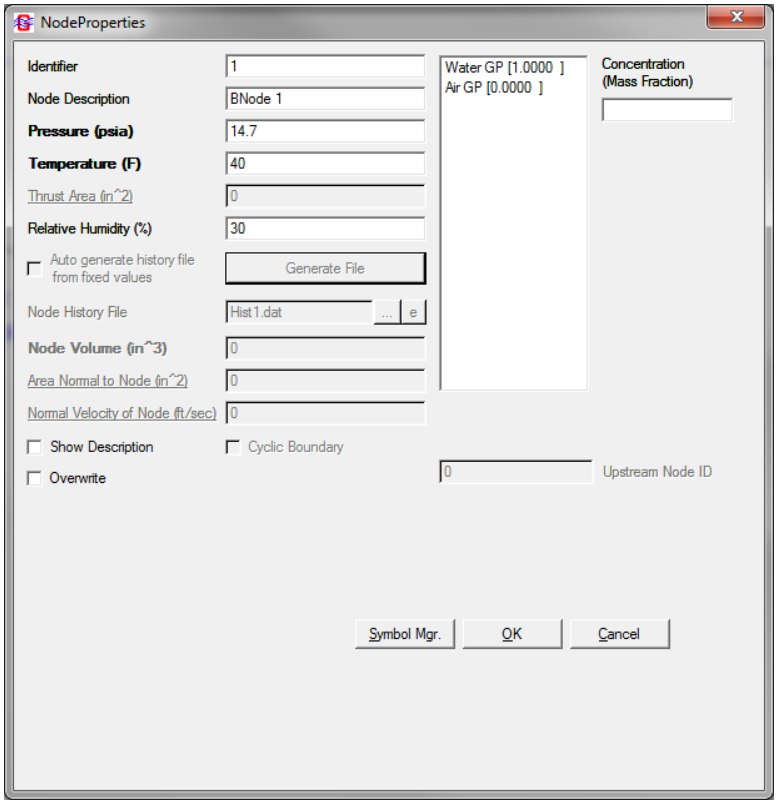


Figure 2.5.4 Boundary Node Properties for Psychrometrics

Psychrometrics was activated in Circuit Option (Figure 3). It may be noted that Relative Humidity option was selected for this model. Therefore, in Boundary Properties (Figure 3), relative humidity was assigned in addition to pressure and temperature. Heat input to node 2 was specified in Node Properties.

Comparison of GFSSP prediction with hand calculation

The air-conditioning problem described in Figure 2.5.1 is solved using mass and energy conservation equation of air-water vapor mixture and psychrometric property calculator, PSYCHRO.

Condition at 1 from PSYCHRO:

$\omega_1 = 0.0015$

$h_1 = 3.595 \frac{Btu}{lb}$

$\dot{m}_a = (1-\omega_1)\dot{m} = (1-0.0015)(1.342) = 1.340\ lb/sec$

$\dot{m}_v = \omega_1 m = (0.0015)(1.342) = 0.002\ lb/sec$

Heater

Energy conservation between (1) and (2)

$$\dot{Q} + \dot{m}_a h_1 = \dot{m}_a h_2$$

$$h_2 = \frac{\dot{Q}}{\dot{m}_a} + h_1 = \frac{10}{1.34} + 3.595 = 11.06\ Btu/lb$$

GFSSP calculates (Figure 2.5.2) $T_2 = 71\ F\ and\ \varphi_2 = 9.6\%$
For this condition PSYCHRO calculates $h_2 = 11.002\ Btu/lb$

Humidifier

Water mass conservation between (2) and (3)

$$\dot{m}_a \omega_2 + \dot{m}_{H2O} = \dot{m}_a \omega_3$$

$$\omega_3 = \omega_2 + \frac{\dot{m}_{H2O}}{\dot{m}_a} = 0.0015 + \frac{0.01154}{1.34} = 0.0101$$

GFSSP calculates $\omega_3 = 0.0102$

Energy conservation (2) and (3)

$$\dot{m}_a h_2 + \dot{m}_{H2O} h_v = \dot{m}_a h_3$$

$$h_3 = h_2 + \frac{\dot{m}_{H2O}}{\dot{m}_a} h_v = 11.02 + \frac{0.01154}{1.34}(1287.3) = 22.10\ Btu/lb$$

$h_v\ is\ the\ enthalpy\ of\ steam\ used\ for\ humidification$

GFSSP calculates $T_3 = 79\ F\ , \varphi_3 = 48\%\ and\ \omega_3 = 0.0102$

According to PSYCHRO,

$$h_3 = 22.13 \frac{Btu}{lb} and\ \omega_3 = 0.01017$$

2.6 Example 32 – Flow Distribution in Manifold

Problem Considered:

This example demonstrates use of GFSSP in modeling flow distribution in a dividing manifold. In a dividing manifold, pressure increases in the direction of flow. This observation may be counter-intuitive from the experience of pipe flow where pressure always drops along the direction of flow. In a dividing flow manifold (Figure 2.6.1), pressure increases because of gradual loss of longitudinal momentum due to lateral flow through the ports. Therefore, accounting for longitudinal inertia is critical for modeling flow distribution in manifold. In this example, GFSSP model has been developed to simulate experiments of Kubo and Ueda.

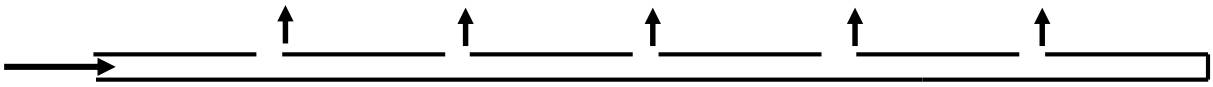


Figure 2.6.1 Schematic of Dividing Flow Manifold

GFSSP Model:

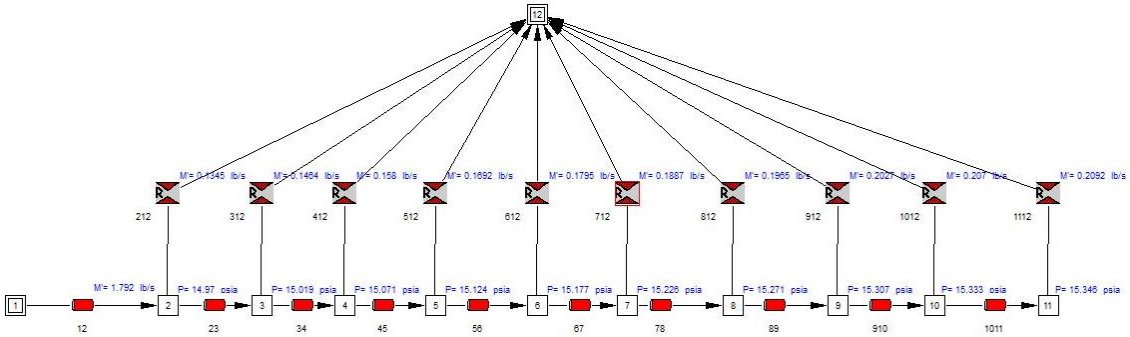


Figure 2.6.2 GFSSP Model of a Dividing Flow Manifold

Figure 2.6.2 shows the GFSSP model of a dividing flow manifold tested by Kubo and Ueda. The test rig has a 18 inch long, ¾ inch internal diameter pipe has 10 lateral orifices of 0.3 inch diameter. GFSSP model is constructed with 10 pipe segments and 10 restrictions.

Inertia Formulation in GFSSP

Inertia term in the momentum equation is expressed as:

$$\frac{\dot{m}_{i+1}\left(u_{i+1}-u_i\right)}{g_c},$$

where suffix (i+1) represents the branch in consideration where suffix (i) represents the immediate upstream branch. In case of multiple upstream branches, GFSSP calculates an average velocity assuming all upstream branches are parallel. This assumption, however, is not correct for manifold where one of the upstream branch (orifice) is at 90 degree with the longitudinal pipe segment branch. Therefore, in this model inertia option is not activated. Instead inertia term was introduced in the user subroutine.

User Subroutine

Inertia term in the pipe segment branches (23, 34, 45, 56, 67, 78, 89, 910, and 1011) is calculated in Subroutine SORCEF (Appendix – A4). TERM1 (Longitudinal Inertia) is calculated for each of those branches. It may be noted that average upstream velocity in the inertia term is the velocity at the upstream pipe segment. For example, the upstream velocity of branch 23 is the velocity at Branch 12 (not average of Branch 12 and Branch 212). It may also be noted that an empirical turning loss coefficient CT also appears in the equation. CT represents viscous loss due to turning.

Results

The results are shown in Figure 2.6.2. It may be noted that pressure increases in the direction of flow which causes the increase in lateral flow towards the closed end of the pipe. The comparison with test data appears in Figure 2.6.3

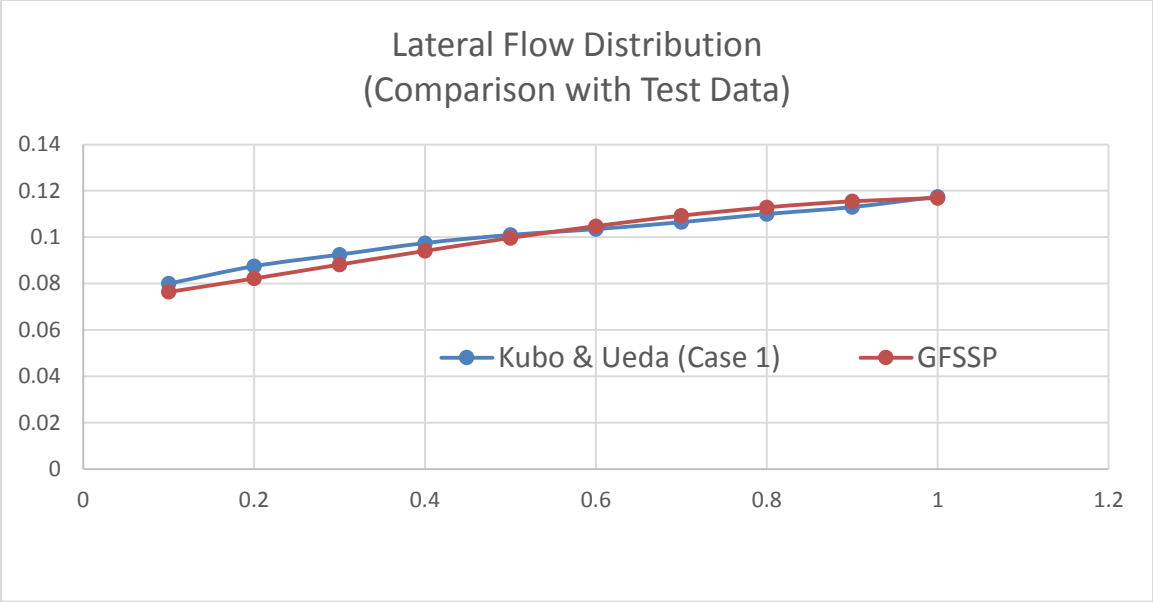


Figure 2.6.3 Comparison of predicted lateral flow distribution with test data

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. Moran, M.J. and Shapiro, H.N., "Fundamentals of Engineering Thermodynamics" 3rd Edition, John Wiley & Sons, Inc, 1996
3. 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.
4. 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

Appendix

A.1 User Subroutine for Example 10

```
      SUBROUTINE SORCEC
C     PURPOSE: ADD CONCENTRATION SOURCES
C***********************************************************************
      INCLUDE 'COMBLK.FOR'
C***********************************************************************
C     ADD CODE HERE
C  PURPOSE:  COMPUTE MASS TRANSFER OF PROPELLANT INTO THE ULLAGE
C            DURING TANK PRESSURIZATION

C  Loop over the number of tanks in the model

      DO 10 I = 1, NTANK

C  Get pointers to ullage and propellant nodes
```

```
         CALL INDEXI(NODUL(I), NODE, NNODES, IPUL)
         CALL INDEXI(NODPRP(I), NODE, NNODES, IPRP)

C  Get the saturation properties of oxygen at the current ullage
C  pressure.  To prevent errors in PROPS_PSATX, pressure is limited
C  by the critical pressure of oxygen:  737.2 psi.

         I_NFLUID = 6                     ! ID number of GASP O2
         Z_P = MIN(P(IPUL), 737.1 * 144)  ! Current ullage pressure (psf)
         Z_XV = 0.0                       ! Dummy value for quality

         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)

         TSAT(I) = Z_T            ! Saturation temperature, deg R
         HFG(I) = Z_HV - Z_HL     ! Heat of vaporization, BTU/lb

C  Estimate mass transfer as Q / [HFG + CP * (Tsat - Tprp)].
C  SORCECON is the mass source in the species concentration equation.
C  The index 2 refers to oxygen being the second fluid.
C  SORCEMAS is the mass source for the overall mass equation.

         SORCECON(IPUL,2) = QULPRP(I) /
     +      (HFG(I) + CPNODE(IPRP) * MAX(TSAT(I)-TF(IPRP),0.0) )
         SORCEMAS(IPUL) = SORCECON(IPUL,2)
         SORCEMAS(IPRP) = -SORCEMAS(IPUL)

C  Copy the current properties of the first tank to user-defined
C  plot variables for plotting in Winplot.

         IF (I .EQ. 1) THEN
            USRPVAR(1) = VOLUME(IPUL)
            USRPVAR(2) = VOLUME(IPRP)
            USRPVAR(3) = QULPRP(1)
            USRPVAR(4) = QULWAL(1)
            USRPVAR(5) = TNKTM(1) - 459.67  ! Convert degR to degF
            USRPVAR(6) = SORCEMAS(IPUL)
         END IF

   10 CONTINUE

      RETURN
      END
C***********************************************************************
      SUBROUTINE BNDUSER
C     PURPOSE: MODIFY BOUNDARY CONDITIONS
C***********************************************************************
      INCLUDE 'COMBLK.FOR'
C***********************************************************************
C     ADD CODE HERE

C  In first time step, initialize user-defined plot variables for Winplot.

      IF (ISTEP .EQ. 1) THEN
         USRVAR = .TRUE.
         USRVARSNUM = 6
         USRPVARNAME(1) = 'VOLULL'
         USRPVARUNIT(1) = 'ft3'
         USRPVARNAME(2) = 'VOLPRP'
         USRPVARUNIT(2) = 'ft3'
         USRPVARNAME(3) = 'QULPRP'
         USRPVARUNIT(3) = 'BTU/s'
         USRPVARNAME(4) = 'QULWAL'
         USRPVARUNIT(4) = 'BTU/s'
         USRPVARNAME(5) = 'TNKTM'
         USRPVARUNIT(5) = 'degF'
         USRPVARNAME(6) = 'Boiloff'
         USRPVARUNIT(6) = 'lb/s'
```

```
      END IF

      RETURN
      END
```

A.2 User Subroutine for Example 12

```
      SUBROUTINE SORCEC
C     PURPOSE: ADD CONCENTRATION SOURCES
C**********************************************************************
      INCLUDE 'COMBLK.FOR'
C**********************************************************************
C     ADD CODE HERE
C  PURPOSE:  COMPUTE MASS TRANSFER OF PROPELLANT INTO THE ULLAGE
C            DURING TANK PRESSURIZATION

      CHARACTER*8, FLUID

C  Loop over the number of tanks in the model

      DO 10 I = 1, NTANK

C  Get pointers to ullage and propellant nodes

         CALL INDEXI(NODUL(I), NODE, NNODES, IPUL)
         CALL INDEXI(NODPRP(I), NODE, NNODES, IPRP)

C  For the first tank, set approximate saturation properties for RP-1

         IF (I .EQ. 1) THEN
            TSAT(I) = 601.7     ! deg R
            HFG(I) = 1601.4     ! BTU/lbm
            INDEXGAS = 3        ! RP-1 is the third fluid in the model

C  For the second tank, get the saturation properties of oxygen at
C  the current ullage pressure.  To prevent errors in PROPS_PSATX,
C  pressure is limited by the critical pressure of oxygen:  737.2 psi.

         ELSE
            I_NFLUID = 6                     ! ID number of GASP O2
            Z_P = MIN(P(IPUL), 737.1 * 144)  ! Current ullage pressure (psf)
            Z_XV = 0.0                       ! Dummy value for quality

            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)

            TSAT(I) = Z_T            ! Saturation temperature, deg R
            HFG(I) = Z_HV - Z_HL     ! Heat of vaporization, BTU/lb
            INDEXGAS = 2             ! Oxygen is the second fluid in the model
         END IF

C  Estimate mass transfer as Q / [HFG + CP * (Tsat - Tprp)].
C  SORCECON is the mass source in the species concentration equation.
C  INDEXGAS refers to oxygen being the second fluid or RP-1 the third fluid.
C  SORCEMAS is the mass source for the overall mass equation.

         SORCECON(IPUL,INDEXGAS) = QULPRP(I) /
     +      (HFG(I) + CPNODE(IPRP) * MAX(TSAT(I)-TF(IPRP),0.0) )
         SORCEMAS(IPUL) = SORCECON(IPUL,INDEXGAS)
         SORCEMAS(IPRP) = -SORCEMAS(IPUL)

C  Copy the current properties of the first or second tank to user-defined
C  plot variables for plotting in Winplot.

         IF (I .EQ. 1) THEN
            USRPVAR(1) = VOLUME(IPUL)
            USRPVAR(2) = VOLUME(IPRP)
            USRPVAR(3) = QULPRP(1)
            USRPVAR(4) = QULWAL(1)
            USRPVAR(5) = TNKTM(1) - 459.67  ! Convert degR to degF
            USRPVAR(6) = SORCEMAS(IPUL)
         ELSE IF (I .EQ. 2) THEN
            USRPVAR(7) = VOLUME(IPUL)
            USRPVAR(8) = VOLUME(IPRP)
            USRPVAR(9) = QULPRP(2)
            USRPVAR(10) = QULWAL(2)
```

```
            USRPVAR(11) = TNKTM(2) - 459.67  ! Convert degR to degF
            USRPVAR(12) = SORCEMAS(IPUL)
         END IF

   10 CONTINUE

      RETURN
      END
C***********************************************************************
      SUBROUTINE BNDUSER
C     PURPOSE: MODIFY BOUNDARY CONDITIONS
C***********************************************************************
      INCLUDE 'COMBLK.FOR'
C***********************************************************************
C     ADD CODE HERE

C  In first time step, initialize user-defined plot variables for Winplot.

      IF (ISTEP .EQ. 1) THEN
         USRVAR = .TRUE.
         USRVARSNUM = 12
         USRPVARNAME(1) = 'VOLURP'
         USRPVARUNIT(1) = 'ft3'
         USRPVARNAME(2) = 'VOLPRP'
         USRPVARUNIT(2) = 'ft3'
         USRPVARNAME(3) = 'QULPRP'
         USRPVARUNIT(3) = 'BTU/s'
         USRPVARNAME(4) = 'QULWRP'
         USRPVARUNIT(4) = 'BTU/s'
         USRPVARNAME(5) = 'TNKTRP'
         USRPVARUNIT(5) = 'degF'
         USRPVARNAME(6) = 'MdotRP'
         USRPVARUNIT(6) = 'lb/s'
         USRPVARNAME(7) = 'VOLUO2'
         USRPVARUNIT(7) = 'ft3'
         USRPVARNAME(8) = 'VOLPO2'
         USRPVARUNIT(8) = 'ft3'
         USRPVARNAME(9) = 'QULPO2'
         USRPVARUNIT(9) = 'BTU/s'
         USRPVARNAME(10) = 'QULWO2'
         USRPVARUNIT(10) = 'BTU/s'
         USRPVARNAME(11) = 'TNKTO2'
         USRPVARUNIT(11) = 'degF'
         USRPVARNAME(12) = 'MdotO2'
         USRPVARUNIT(12) = 'lb/s'
      END IF

      RETURN
      END
```

A.3 User Subroutine for Example 29

From file Ex29.for:

```
      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
      DIMENSION NODEULG(5)

C     PUT ULLAGE NODES INTO AN ARRAY FOR ACCESSING
      DATA (NODEULG(I),I = 1,5)/2,8,9,10,11/
      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
      IF (ISTEP.LE.1) RETURN
      CALL INDEXI(NUMUL,NODE,NNODES,IPUL)
      CALL INDEXI(NUMPRP,NODE,NNODES,IPPRP)
C     ESTIMATE MASS TRANSFER FROM PROPELLANT TO ULLAGE
```

```
C     USE ULLAGE TO INTERFACE HEAT TRANSFER CALCULATED IN SORCEQ
C     ESTIMATE EVAPORATION RATE FROM ULLAGE TO PROPELLANT
      EMDOTGH2 = (QULPRP(1)-QULPRP(2))/USRVAR1(10) ! "USRVAR1(10) = HFGFILM" IN SORCEQ
      VOLDOT = EMDOTGH2/RHO(IPPRP)
      DVOL = VOLDOT*DTAU
C     STORE MASS TRANSFER RATE TO USE IN SORCEQ AND BNDUSER
      USRVAR1(2) = EMDOTGH2
      IF (NODE(IPN).EQ. 2) THEN
        EMS(IPN) = EMDOTGH2
        VOLUME(IPN) = VOLUMEM(IPN) + DVOL
      ENDIF
      IF (NODE(IPN).EQ. 4) THEN
        EMS(IPN) = -EMDOTGH2
        VOLUME(IPN) = VOLUMEM(IPN) - DVOL
      ENDIF

      DO I = 1,5
        NUMBER = NODEULG(I)
        CALL INDEXI(NUMBER,NODE,NNODES,IPULG)
        IF (SATURATED(IPULG)) THEN
          EMLIPU = EM(IPULG)*(1.-XV(IPULG))
          SORCEMAS(IPULG) = -EMLIPU
          IF (XV(IPULG).LT.0.95) THEN
            USRVAR1(50) = 1.0
           ELSE
            USRVAR1(50) = 0.0
          ENDIF
         ELSE
          SORCEMAS(IPULG) = 0.0
        ENDIF
       ENDDO ! DO I = 1,5

      RETURN
      END
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
      DIMENSION NODEULG(5)


C     PUT ULLAGE NODES INTO AN ARRAY FOR ACCESSING
      DATA (NODEULG(I),I = 1,5)/2,8,9,10,11/
C     DEFINE ULLAGE AND PROPELLANT NODE POINTERS
      DATA HVAPOR/83.89/
      DATA TIL,HFGLH2/38.07,191.30/
      IF (ISTEP.EQ.1) THEN
        NODUL(1) = 2
        NODPRP(1) = 4
        CALL INDEXI(NODUL(1), NODE, NNODES, IPUL)
        CALL INDEXI(NODPRP(1), NODE, NNODES, IPPRP)
C     ASSIGN SURFACE AREA BETWEEN ULLAGE AND PROPELLANT (10 FEET DIA)
        ELHC(1) = 10.00
        ARHC(1) = 78.54
        FCTHC(1) = 0.5
        FCTHC(2) = 0.75
        CIP = 0.27
        FNIP = 0.25
      ENDIF

C     COMPUTE HEAT TRANSFER BETWEEN PRESSURANT AND PROPELLANT
C     OBTAIN SATURATION TEMPERATURE AT ULLAGE PRESSURE
       Z_P = P(IPUL)
       I_NFLUID = NFLUID(1)
```

```
        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)
C     COMPUTE HEAT TRANSFER COEFFICIENT BETWEEN ULLAGE AND SATURATED FILM
        TSFILM = Z_T
        HVAPOR = Z_HV
        HFGFILM = Z_HV - Z_HL
        USRVAR1(10) = HFGFILM  ! STORE IN USER VARIABLE ARRAY FOR USE IN SORCEM
        USRVAR1(11) = TSFILM
        BETA = 1./Z_T
        DELTAT = ABS(TF(IPUL)-TSFILM)
        IF (DELTAT .GT. 0.0) THEN
          GR = (ELHC(1)**3*Z_RHOV**2*G*BETA*DELTAT)/Z_MUV**2
          PRNDTL = Z_CPV*Z_MUV/Z_KV
          HCUP = FCTHC(1)*CIP*Z_KV*(GR*PRNDTL)**FNIP/ELHC(1)
          Z_NU=CIP*(GR*PRNDTL)**FNIP
          USRVAR1(20) = Z_NU
C          HCUF = 100.*Z_KV/ELHC(1)
          HCUF = HCUP
          QULPRP(1) = HCUF*ARHC(1)*(TF(IPUL)-TSFILM)
        ENDIF ! IF (DELTAT .GT. 0.0)
C     COMPUTE HEAT TRANSFER COEFFICIENT BETWEEN SATURATED FILM AND PROPELLANT
        DELTAT=ABS(TSFILM - TF(IPPRP))
        IF (DELTAT .GT. 0.0) THEN
          GR = (ELHC(1)**3*Z_RHOL**2*G*BETA*DELTAT)/Z_MUL**2
          PRNDTL = Z_CPL*Z_MUL/Z_KL
          HCFP = FCTHC(1)*CIP*Z_KL*(GR*PRNDTL)**FNIP/ELHC(1)
C          HCFP = Z_KL/ELHC(1)
          QULPRP(2) = HCUF*ARHC(1)*(TSFILM - TF(IPPRP))
        ENDIF ! IF (DELTAT .GT. 0.0)

C     DEFINE HEAT SOURCE (NOTE: USRVAR1(2) = EMDOTGH2)

         HSORCE(IPUL) = USRVAR1(2)*HVAPOR-QULPRP(1)
         HSORCE(IPPRP) = -USRVAR1(2)*HVAPOR+QULPRP(2)


      DO I = 1,5
        NUMBER = NODEULG(I)
        CALL INDEXI(NUMBER,NODE,NNODES,IPULG)
        IF (SATURATED(IPULG)) THEN
          EMLIPU = EM(IPULG)*(1.-XV(IPULG))
          HREQ = EMLIPU*HFGLH2/DTAU
          SORCEH(IPULG) = SORCEH(IPULG) + HREQ
          IF (XV(IPULG).LT.0.95) THEN
            USRVAR1(50) = 1.0
           ELSE
            USRVAR1(50) = 0.0
          ENDIF
        ENDIF
       ENDDO ! DO I = 1,5
100    FORMAT(2I6,8F10.4)
      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),NODESPR(6),IPSPR(6),SPRAYFL(6),SPRAYHL(6)
      LOGICAL SPRAY
      DATA SPRAY/.FALSE./
      DATA (MLINODE(I),I = 1,6)/32,33,34,35,36,37/
      DATA (NODESPR(I),I=1,6)/2,8,9,10,11,4/

C     UPDATE PRESSURE OF THE PSEUDO-BOUNDARY NODE
```

```
      DATA TIL,HFGLH2/38.07,191.30/
      DATA HAREA/ 27.882/
      DATA FRACSPRAY/1.0/
      OPEN (UNIT = NUSR3, FILE = 'DEBUG.DAT',STATUS = 'UNKNOWN',
     &      ACTION = 'WRITE')
      PULMAX = 20*144
      PULMIN = 19*144
      OPENARU = 0.003526
      OPENARL = 0.003526
      CLAREA = 1.E-16
      USRVAR2(1) = 0.0

      NUMUL=2
      NUMPSN = 3
      NUMPRP = 4

      SUMQMLI = 0.0
      DO I = 1,NMLI
        SUMQMLI = SUMQMLI + QDOTMLI(I)
      ENDDO

      USRPVAR(7) = SUMQMLI

      USRPVAR(8) = QULPRP(1)

      USRPVAR(9) = USRVAR1(2)

      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     NODE 2 IS PRESSURE MONITORING NODE

      CALL INDEXI(2, NODE, NNODES, IPN2)

C     NODE 4 IS LIQUID PROPELLANT NODE

      CALL INDEXI(4, NODE, NNODES, IPN4)

C     ACTIVATE OR DEACTIVATE SPRAY BASED ON ULLAGE PRESSURE

      IF (P(IPN2).GE.PULMAX) SPRAY = .TRUE.
      IF (P(IPN2).LE.PULMIN.OR. USRVAR1(50).EQ.1.0) SPRAY = .FALSE.

      IF (SPRAY) THEN

C     SET TIME STEP TO LOWER VALUE

      DTAU = 0.01

C     VARIABLE TO ACTIVATE MLI HEAT LOAD CALCULATION
      USRVAR2(1) = 1.00

C     GET POINTERS FOR ULLAGE NODES
      DO I = 1,6
        CALL INDEXI(NODESPR(I),NODE,NNODES,IPSPR(I))
      ENDDO

C     WRITE PRESSURE & TEMPERATURE FOR LIQUID AND ULLAGE NODE FOR TVS MODEL

      OPEN (UNIT = NUSR1, FILE = 'TO_TVS.DAT',STATUS = 'REPLACE',
     &      ACTION = 'WRITE')
      WRITE(NUSR1,*) P(IPN4),TF(IPN4),P(IPN2)
      CLOSE(NUSR1)

C     RUN TVS MODEL
```

```
      CALL SYSTEM ('TVS.EXE TVS.DAT > Dummy.txt')

C     READ DATA FROM TVS MODEL

      OPEN (UNIT = NUSR2, FILE = 'FROM_TVS.DAT', STATUS = 'OLD',
     &      ACTION = 'READ')

      READ(NUSR2,*) (SPRAYFL(I),SPRAYHL(I), I = 1,6)
      CLOSE (NUSR2)

C     SET MASS & ENERGY SOURCES TO ULLAGE NODES

      DO I = 1,5
        SORCEMAS(IPSPR(I)) = FRACSPRAY*SPRAYFL(I)
        HSORCE(IPSPR(I)) = FRACSPRAY*SPRAYFL(I)*SPRAYHL(I)
      ENDDO

      ELSE

C    SET TIME STEP TO ORIGINAL VALUE

      DTAU = 0.1

      DO I = 1,5
        SORCEMAS(IPSPR(I)) = 0.0
        HSORCE(IPSPR(I)) = 0.0
      ENDDO

      ENDIF ! IF (SPRAY) THEN

      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 = ANU * CONDF(IPN) / HL

      RETURN
      END
```

From file TVS.for:

```
      SUBROUTINE PRNUSER
```

```
C     PURPOSE: ADD NEW OUTPUT
C*********************************************************************
      INCLUDE 'comblk.for'
C*********************************************************************
C     ADD CODE HERE
      DIMENSION ISPRAY(6),IBSPRAY(6),SPRAYFL(6),SPRAYHL(6)
      DATA (ISPRAY(I),I = 1,6) /89,90,91,92,93,94/
C     WRITE DATA FROM TVS MODEL FOR SELF PRESSURIZATION MODEL INPUT

      OPEN (UNIT = NUSR2, FILE = 'FROM_TVS.DAT', STATUS = 'REPLACE',
     &      ACTION = 'WRITE')

C     OBTAIN INDICES FOR THE SPRAY NOZZLE
      DO I = 1,6
        CALL INDEXI(ISPRAY(I),IBRANCH,NBR,IBSPRAY(I))
      ENDDO

C     OBTAIN POINTER OF NODE 8 FOR ENTHALPY
      CALL INDEXI(8,NODE,NNODES,IPN8)

C     GENERATE ARRAY FOR MASS FLOWRATE & ENTHALPY
      DO I = 1,6
       SPRAYFL(I) = FLOWR(IBSPRAY(I))
       SPRAYHL(I) = H(IPN8)
      ENDDO

      WRITE(NUSR2,100) (SPRAYFL(I),SPRAYHL(I), I = 1,6)
      CLOSE (NUSR2)
      RETURN
100   FORMAT(2E14.5)
      END
```

A.4 User Subroutine for Example 32

From file Ex32.for:

```
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

C    DEFINE TURNING LOSS COEFFICIENT
      DATA CT/0.91/

      CALL INDEXI(12,IBRANCH,NBR,IB12)
      CALL INDEXI(23,IBRANCH,NBR,IB23)
      CALL INDEXI(34,IBRANCH,NBR,IB34)
      CALL INDEXI(45,IBRANCH,NBR,IB45)
      CALL INDEXI(56,IBRANCH,NBR,IB56)
      CALL INDEXI(67,IBRANCH,NBR,IB67)
      CALL INDEXI(78,IBRANCH,NBR,IB78)
      CALL INDEXI(89,IBRANCH,NBR,IB89)
```

```
 CALL INDEXI(910,IBRANCH,NBR,IB910)
 CALL INDEXI(1011,IBRANCH,NBR,IB1011)

 IF(IBRANCH(I).EQ.23)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB12))/GC

 IF(IBRANCH(I).EQ.34)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB23))/GC

 IF(IBRANCH(I).EQ.45)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB34))/GC

 IF(IBRANCH(I).EQ.56)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB45))/GC

 IF(IBRANCH(I).EQ.67)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB56))/GC

 IF(IBRANCH(I).EQ.78)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB67))/GC

 IF(IBRANCH(I).EQ.89)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB78))/GC

 IF(IBRANCH(I).EQ.910)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB89))/GC

 IF(IBRANCH(I).EQ.1011)
&   TERM1=CT*MAX(FLOWR(I),0.0)*(VEL(I)-VEL(IB910))/GC

 RETURN
 END
```