

# **AC 2009-2295: EXCEL IN ME: EXTENDING AND REFINING UBIQUITOUS SOFTWARE TOOLS**

**Kenny Mahan, University of Alabama**

**Jesse Huguet, University of Alabama**

**Joseph Chappell, University of Alabama**

**Keith Woodbury, University of Alabama**

**Robert Taylor, University of Alabama**

# **Excel in ME: Extending and Refining Ubiquitous Software Tools**

(Excel Modules for Thermodynamic Properties of Refrigerants R134a and R22 and Compressible Ideal Gas Flow)

## **Abstract**

Microsoft Excel is a ubiquitous software tool that provides an excellent electronic format for engineering computation and organization of information. This paper reports on the second year of an NSF CCLI Phase I project to implement a sequence of Excel modules for use in the Thermal Mechanical Engineering Curriculum.

A collection of Excel Add-ins has been developed for use in solving thermodynamics problems. This paper reports on development of three Add-ins to compute properties of refrigerants R134 and R22 and to compute gas dynamics relations for isentropic, Fanno, and Rayleigh flows of ideal gases. All of the Excel Add-ins developed can be downloaded at the project website [www.me.ua.edu/ExcelinME](http://www.me.ua.edu/ExcelinME).

## **Intro**

Under a National Science Foundation (NSF) Curriculum, Classroom, and Laboratory Improvement (CCLI) grant a number of software modules have been developed to facilitate engineering analysis in a computational spreadsheet. The ubiquitous spreadsheet of choice is Microsoft Excel.

In an earlier paper by Chappell, et al. (2008), a Microsoft Excel module called XSteam, developed by Magnus Holmgren (Holmgren 2007), was adapted and extended to compute thermodynamic properties of steam/water from a wide range of input properties. After the successful implementation of these expanded steam Excel modules in the classroom, attention was turned toward adding capability for other substances, in particular the refrigerants R134a and R22. This paper addresses the implementation and testing process of modules to calculate the thermodynamic properties of R134a and R22.

Another topic of interest in thermodynamics is compressible flow of ideal gases. An Excel module has been developed to compute basic functions for this area, including isentropic flow, normal shock, Fanno Flow, and Rayleigh Flow. This paper will also present a summary of this development.

## **R134a**

### Description

1,1,1,2-Tetrafluoroethane, commonly referred to as R134a, is a refrigerant primarily used in automobile air conditioners today. The R134a module developed in this work uses Excel macros to compute the thermodynamic properties of R134a.

### *Why it is needed*

The development of the R134a module is an important step in expanding the type of thermodynamic problems that can be solved in Thermal Mechanical Engineering classes within the spreadsheet environment. The addition of the R134a module allows students to tackle problems involving refrigerants directly in the Excel spreadsheet environment without the need of looking up values in published tables of properties.

### *Source for implementation*

A paper by Tillner-Roth and Baehr (1994) on a fundamental equation of state for R134a has become a standard source for computation of thermodynamic properties of R134a. Their work and the existing XSteam module, which was expanded and modified by Huguet, et al. (2008), were used as a starting point to implement the R134a module. The idea was to take an approach similar to Huguet's to expand the R134a module beyond the basic functionality provided by Tillner-Roth and Baehr.

The fundamental equation of state for the refrigerant offers several basic relationships and constants which were used to develop the primary functions for the R134a module in Visual Basic. The XSteam module was invaluable to the expansion of the R134a module since the coding used in each module was very similar. Unfortunately the R134a paper did not provide as many initial functions as Holmgren provided in his XSteam module. To compensate, iterative methods were implemented to develop a comprehensive module.

### *Naming of functions*

To remain consistent with previous modules, the same naming convention used for the XSteam module was used for the R134a module. "The name of the functions is of the form "var1\_var2var3", where var1 is the property to be computed, var2 is the first property passed in the call list, and var3 is the second argument passed in the call list (Woodbury, 2008)." To differentiate R134a functions from XSteam functions, "\_R134a" was appended to the end of each R134a function name. For example, p\_vT\_R134a returns the pressure of R134a as a function of specific volume and temperature.

### *Units*

Each R134a function computes in SI units. However, if a user desires values in the US Customary system of units, an optional third parameter is passed to the function. If the third parameter is the character string "ENG," then the input and output units will be US Customary units (Woodbury, 2008).

## Implementation

For the implementation of the R134a module, fundamental correlations from Tillner-Roth and Baehr (1994) were first used to create a set of primary functions and additional functions were added using iteration with these primary functions.

Using regression analysis, Tillner-Roth and Baehr (1994) determined equations for vapor pressure, saturated liquid density, and saturated vapor densities. Their paper presents the Helmholtz free energy equation and its derivatives in dimensionless form. From the dimensionless free energy equation and the ideal gas law, Tillner-Roth and Baehr developed equations for pressure, internal energy, enthalpy, and entropy with respect to temperature and specific volume. The functions developed from these equations are designated *primary functions* because they are computed directly using these relations from Tillner-Roth and Baehr.

Since the equations provided by Tillner-Roth and Baehr's paper only solve the primary equations when a single phase relation exists, it was important to insert logic checks into the coding of the primary equations to determine if the thermodynamic state resides in the two phase region. An example of the coding of a typical primary function,  $p_{Tv\_R134a}$ , can be seen in Figure 1. If the state resides in the saturated region the pressure would simply be the saturation pressure at the corresponding temperature; otherwise the single phase relationship from the paper would be used to solve for the pressure.

After the primary functions were coded in the R134a module, the next step was to determine what combinations of thermodynamic properties could be classified as secondary functions. To expand this module, iterative techniques, such as the bisection or secant methods, were used to manipulate the primary functions to develop a set of secondary functions that rely only on the primary functions. Because this group of functions relies only on the primary functions, additional iterations (iterations of iterations) are avoided, which help minimize the execution time.

The bisection method was used for the iterations and an example of how it is used in the coding of secondary functions can be seen in Figure 2. As in the coding of  $v_{Tu\_R134a}$ , when the thermodynamic state lies outside of the saturation region, the bisection method is implemented. The secant method was also considered, but the bisection method was determined to be the most robust and was utilized in all of the secondary modules.

After the possibilities of secondary functions were exhausted, the next step was to define a set of tertiary functions that call upon both primary and secondary functions. Since tertiary functions iterate using secondary functions that already use iterations, some of the tertiary functions required excessive execution times or were inaccurate and were therefore abandoned.

Table 1 has a listing of all the primary, secondary, and tertiary functions that were developed for R134a.

```

'*****
' *1 Property Functions
'*****
' *1.1 Pressure
' Pressure
Function p_Tv_R134a(ByVal T As Double, ByVal v As Double, Optional Units As String = "SI") As Double

Units = UCase(Units)
If (Units = "ENG") Then
    T = fromENGtoSI_T(T)
    v = fromENGtoSI_v(v)
End If

tau = tau_R134a(T)
delta = delta_R134a(v)
' check if less than critical temperature
If (T < Tcrit) Then
    ' if so, check for saturation
    vL = vL_T_R134a(T)
    vV = vV_T_R134a(T)
    If (v <= vV And v >= vL) Then
        ' if saturated return P_sat
        p_Tv_R134a = psat_T_R134a(T)
        If (Units = "ENG") Then
            p_Tv_R134a = fromSItoENG_p(p_Tv_R134a)
        End If
        Exit Function
    End If
End If
' if supercritical then use single phase relation
p_Tv_R134a = (1 + delta * dphi_residual_delta(tau, delta)) * R * (T + 273.15) * 1 / v

If (Units = "ENG") Then
    p_Tv_R134a = fromSItoENG_p(p_Tv_R134a)
End If

End Function

```

Figure 1. Primary Function for p\_Tv\_R134a

```

f_min = u - u_Tv_R134a(T, v_min)
f_max = u - u_Tv_R134a(T, v_max)
' Bisection Method
For Count = 0 To maxit ' set a max limit
    v_mid = (v_max + v_min) / 2
    f_mid = u - u_Tv_R134a(T, v_mid)
    If f_min * f_mid < 0 Then
        v_max = v_mid
        f_max = f_mid
    Else
        v_min = v_mid
        f_min = f_mid
    End If
    If (Abs((f_mid) / u) < tol_equal) Then
        Exit For
    End If
Next Count

```

Figure 2. Bisection Method Code in v\_Tu\_R134a.

<b>Table 1. R134a Functions with RMS Error Values</b>					
<b>Primary Functions</b>	<b>RMS Values</b>	<b>Secondary Functions</b>	<b>RMS Values</b>	<b>Tertiary Functions</b>	<b>RMS Values</b>
<b>h_px_R134a</b>	0.0005	<b>h_sx_R134a</b>	Abandoned	<b>h_prho_R134a</b>	0.0042
<b>h_Tv_R134a</b>	0.0034	<b>h_vx_R134a</b>	Abandoned	<b>h_ps_R134a</b>	0.0139
<b>h_Tx_R134a</b>	0.0004	<b>p_Th_R134a</b>	0.0628	<b>h_pT_R134a</b>	0.0005
<b>hL_p_R134a</b>	0.0006	<b>p_Ts_R134a</b>	0.0568	<b>p_sv_R134a</b>	Abandoned
<b>hL_T_R134a</b>	0.0001	<b>p_Tu_R134a</b>	0.0675	<b>s_hv_R134a</b>	0.0067
<b>hV_p_R134a</b>	0.0006	<b>s_vx_R134a</b>	Abandoned	<b>s_ph_R134a</b>	0.0068
<b>hV_T_R134a</b>	0.0006	<b>T_hv_R134a</b>	0.0015	<b>s_pT_R134a</b>	0.0003
<b>p_Tv_R134a</b>	1.2879	<b>T_ph_R134a</b>	0.0413	<b>s_pv_R134a</b>	0.0034
<b>psat_T_R134a</b>	0.0062	<b>T_ps_R134a</b>	0.0737	<b>s_Th_R134a</b>	0.0030
<b>s_px_R134a</b>	0.0005	<b>T_pv_R134a</b>	0.0092	<b>s_Tu_R134a</b>	0.0031
<b>s_Tv_R134a</b>	0.0032	<b>Tsat_p_R134a</b>	0.0005	<b>T_hs_R134a</b>	Abandoned
<b>s_Tx_R134a</b>	0.0004	<b>u_sx_R134a</b>	Abandoned	<b>T_sv_R134a</b>	Abandoned
<b>sL_p_R134a</b>	0.0006	<b>u_vx_R134a</b>	Abandoned	<b>T_uv_R134a</b>	Abandoned
<b>sL_T_R134a</b>	0.0001	<b>v_ps_R134a</b>	3.7992	<b>u_hv_R134a</b>	0.0069
<b>sV_p_R134a</b>	0.0006	<b>v_pT_R134a</b>	0.0017	<b>u_ph_R134a</b>	0.0069
<b>sV_T_R134a</b>	0.0006	<b>v_Th_R134a</b>	0.0956	<b>u_ps_R134a</b>	0.0131
<b>u_px_R134a</b>	0.0005	<b>v_Ts_R134a</b>	0.0135	<b>u_pT_R134a</b>	0.0005
<b>u_Tv_R134a</b>	0.0033	<b>v_Tu_R134a</b>	0.0451	<b>u_pv_R134a</b>	0.0042
<b>u_Tx_R134a</b>	0.0003	<b>v_sx_R134a</b>	Abandoned	<b>u_Th_R134a</b>	0.0006
<b>uL_p_R134a</b>	0.0006			<b>h_pv_R134a</b>	0.0042
<b>uL_T_R134a</b>	0.0001			<b>u_Ts_R134a</b>	0.0026
<b>uV_p_R134a</b>	0.0005			<b>u_hs_R134a</b>	Abandoned
<b>uV_T_R134a</b>	0.0005			<b>u_sv_R134a</b>	Abandoned
<b>v_px_R134a</b>	0.0065			<b>v_ph_R134a</b>	3.5783
<b>v_Tx_R134a</b>	0.0093			<b>h_Ts_R134a</b>	0.0026
<b>vL_p_R134a</b>	0.0002			<b>h_Tu_R134a</b>	0.0007
<b>vL_T_R134a</b>	0.0003			<b>v_hs_R134a</b>	Abandoned
<b>vV_p_R134a</b>	0.0083			<b>x_hv_R134a</b>	0.0125
<b>vV_T_R134a</b>	0.0115			<b>x_hs_R134a</b>	Abandoned
<b>x_ph_R134a</b>	0.0106			<b>x_sv_R134a</b>	Abandoned
<b>x_ps_R134a</b>	0.0113			<b>h_sv_R134a</b>	Abandoned
<b>x_pu_R134a</b>	0.0118			<b>p_hv_R134a</b>	0.0042
<b>x_pv_R134a</b>	0.0065			<b>p_hs_R134a</b>	Abandoned
<b>x_Th_R134a</b>	0.0278				
<b>x_Ts_R134a</b>	0.0062				
<b>x_Tu_R134a</b>	0.0068				
<b>x_Tv_R134a</b>	0.0082				
<b>rhoL_T_R134a</b>	0.0003				
<b>rhoV_T_R134a</b>	0.0119				

### Comment on Use of Thermodynamic Equation of State

One might ask why the thermodynamic equation of state (the Helmholtz function) was used as the basis for the computation of thermodynamic properties. Why not, instead, use a table lookup procedure such as is employed by students to solve problems? That is, why not implement the tables within VBA as a matrix of numbers and develop algorithms based on linear interpolation of these matrices?

The primary reason is that thermodynamic properties listed in tables in textbooks are computed from an equation of state. This is in fact how thermodynamic properties are determined. The most accurate method to reproduce these table values will be to use such an equation of state.

A second reason is to minimize data entry errors and reduce the need for comprehensive verification testing. If the table of numbers from a printed set of tables were to be entered, approximately 1,850 separate values would be needed. Each of these would need to be typed in and verified. Also, in the verification stage, each and every pair of possible interpolation pairs would need to be tested in order to assure accuracy. With the thermodynamic equation of state, a random sample of points in the state space will suffice to verify the accuracy of the implementation.

A final reason also relates to the accuracy of linear interpolation in the tables. Although printed thermodynamic tables are touted as reliably accurate under linear interpolation, this might not hold in the case of double or triple interpolations required for iteration of tertiary functions.

### Verification of Functions

#### *Methods for testing*

To test the accuracy of the functions of the R134a module, an array of states was chosen in the saturation, super heated, and super critical regions to generally encompass all of the thermodynamic state space. The reference values of the properties for these states were found from the ASHRAE R134a tables (ASHRAE, 2005). Using these fully defined states, the R134a functions could be checked by inputting table values into pre-developed testing tables. An example of the testing tables used to check the R134a function  $p_{hv\_R134a}$  can be seen in Figure 3 below.

Similar table formats were used for all function checks. Table values were manually input into appropriate cells in the tables. To test the functions, the function in question would call upon the input property values in the same row to compute the answer for the desired output property. To verify the functional accuracy of the module a procedure similar to the one used by Chappel et al. (2008) was used to check each function. The function output was compared to the table values to check the relative error of the function at a specific state. The relative error was calculated using Equation 1.

$$\%error = \frac{y_i - y_{data}}{y_{data_i}} \times 100\% = e_{rel_i} \quad (1)$$

The function p\_hv\_R134a worked very well inside the paper's specified temperature range of 170K (-103C) to 455K (182C) (Figure 3). When the state specified is outside of this range, the function is programmed to return #VALUE to signal that this combination is not possible. The goal for all functions was to have an accuracy of at most 0.5 percent error from the table values. After the relative error was found for each of the states, an average error (RMS error) over all the states was found for each function. Equation 2 shows how the RMS error was calculated.

$$RMS_{rel} = \sqrt{\frac{\sum_{i=1}^n e_{rel_i}^2}{n}} \quad (2)$$

## Results of Module

### *Accuracy*

A complete list of all primary, secondary, and tertiary functions and their RMS error values in the R134a module can be seen in Table 1. The RMS error value represents the root mean square error value (average relative error) for each function considering the suite of states used for evaluation. This value indicates the average accuracy of each function when compared to the ASHRAE table values.

As can be seen by the table, all of the RMS error values are very low except for some of the secondary and tertiary functions that solve for or utilize specific volume. The large RMS error values for these functions are due to table truncation errors and high sensitivity around the saturated liquid region. The steep slope of the state space in the liquid region results in high sensitivity to specific volume in the compressed liquid region, so even slight errors will become greatly magnified. The table values used to verify the module's accuracy have been truncated to an accuracy of about four significant figures. Use of these values as data to the Excel functions results in values that correspond to nearby states, but not exactly to the (truncated) table value.

Overall, the RMS error values of the R134a module were within the acceptable accuracy range and well below the targeted 0.5%. The module consists of seventy-four total functions.

### *Computational Speed*

To maintain relatively fast calculations, minimizing the number of iterations required by each function was important. When iterative functions iterate using other iterative functions, the speed of the calculations suffers. Steps were taken to reduce calculation time by eliminating unneeded iterations in functions. Although the original goal was to include all available combinations, some of the secondary and tertiary functions were abandoned due to



	A	B	C	D	E	F	G	H	I	J	K	L
9	<b>Check for p_hv_R134a</b>											
10	Data Points from ASHRAE Tables											
11	Saturation					Table		Table				
12	T	P <sub>sat</sub>	P	h <sub>f</sub>	vf	p_hv_R134a	%Error	h <sub>g</sub>	vg	p_hv_R134a	%Error	
13	(°C)	(MPa)	(kPa)	(kJ/kg)	(m <sup>3</sup> /kg)			(kJ/kg)	(m <sup>3</sup> /kg)			
14	-103.30	0.0004	0.39	71.46	0.0006	#VALUE!	#VALUE!	334.94	35.4960	0.395923103	1.5187	
15	-100.00	0.0006	0.56	75.36	0.0006	0.566905241	1.2331	336.85	25.1930	0.566905241	1.2331	
16	-50.00	0.0295	29.45	135.67	0.0007	29.42960901	-0.0692	367.65	0.60620	29.44453168	-0.0186	
17	-20.00	0.1327	132.7	173.64	0.0007	132.5654859	-0.1239	386.55	0.14739	132.6527636	-0.0582	
18	10.00	0.4146	414.6	213.58	0.0008	414.4300195	-0.0434	404.32	0.04944	414.6709655	0.0147	
19	40.00	1.0166	1017	256.41	0.0009	1016.232207	-0.0362	419.43	0.01997	1016.113632	-0.0478	
20	70.00	2.1168	2117	304.28	0.001	2116.244136	-0.0263	428.65	0.00865	2116.862869	0.0030	
21	85.00	2.9258	2926	332.22	0.0011	2925.666567	-0.0046	427.76	0.00550	2925.401016	-0.0136	
22	95.00	3.5912	3591	355.25	0.0013	3591.246867	0.0013	420.67	0.00374	3593.408864	0.0615	
23	100.00	3.9724	3972	373.30	0.0015	3972.150855	-0.0063	407.68	0.00268	3973.124354	0.0182	
24	101.06	4.0593	4059	389.64	0.002	4059.096498	-0.0050	389.64	0.00195	4065.672359	0.1570	
25												
26	Superheated					Table						
27	T	P <sub>sat</sub>	P	h	v	p_hv_R134a	%Error					
28	(°C)	(MPa)	(kPa)	(kJ/kg)	(m <sup>3</sup> /kg)							
29	-20.00	0.1013	101.3	387.68	0.1957	101.372307	0.0467					
30	40.00	1.000	1000	419.99	0.0204	999.7682101	-0.0232					
31	70.00	0.1013	101.3	464.43	0.2725	101.3843734	0.0586					
32	70.00	2.000	2000	432.22	0.0096	2000.5335	0.0267					
33	110.00	4.000	4000	446.28	0.0043	3995.959508	-0.1010					
34	120.00	1.000	1000	504.19	0.0296	1000.141684	0.0142					
35	150.00	0.1013	101.3	542.83	0.3378	101.4324841	0.1061					
36	170.00	2.000	2000	550.79	0.0163	1998.336367	-0.0832					
37	200.00	1.000	1000	591.29	0.0371	#VALUE!	#VALUE!					
38	210.00	4.000	4000	584.95	0.0085	#VALUE!	#VALUE!					
39	250.00	2.000	2000	644.3	0.0203	#VALUE!	#VALUE!					
40	300.00	4.000	4000	697.38	0.011	#VALUE!	#VALUE!					
41												
42	Supercritical											
43	T	P <sub>sat</sub>	P	h	v	p_hv_R134a	%Error					
44	(°C)	(MPa)	(kPa)	(kJ/kg)	(m <sup>3</sup> /kg)							
45	110.00	6.00	6000	375.61	0.0013	5984.030474	-0.2662					
46	150.00	6.00	6000	484.69	0.0035	5987.970548	-0.2005					
47	180.00	6.00	6000	531.45	0.0045	5978.823109	-0.3529					
48	240.00	6.00	6000	611.63	0.0059	#VALUE!	#VALUE!					
49	270.00	6.00	6000	650.43	0.0066	#VALUE!	#VALUE!					
50	300.00	6.00	6000	689.36	0.0072	#VALUE!	#VALUE!					
51												

Figure 3. Testing Table for a R134a function to find the pressure when enthalpy and specific volume are known.

inconsistencies in the data and the large amount of calculation time required. The thermodynamic combinations eliminated were uncommon combinations such as `_hv`, `_vx`, and `_sx`. These combinations do not typically appear in practice or problems so they do not subtract from the usability of the module.

A comprehensive evaluation of execution times of the functions was not conducted, but the functions all return values typically within milliseconds. A random sampling of three of the

tertiary functions was benchmarked for speed. The functions were called at least 1000 times consecutively, and the VBA timer() function was used to infer the average time for execution (ignoring overhead of the FOR..NEXT loop used in programming the test). The functions tested and their average execution times are listed in Table 2 below. Note that the slowest of these functions requires only 10 milliseconds to return a value.

<b>Function</b>	<b>Region</b>	<b>Average Execution Time (seconds)</b>
x_hv_R134a	Saturation	0.004078
s_pv_R134a	Superheated	0.000292
v_ph_R134a	superheated	0.010047
v_ph_R134a	saturated	0.00075

### R134a Example

#### *Problem*

A rigid, well-insulated tank having a volume of  $0.2 \text{ m}^3$  is filled initially with R134a vapor at a pressure of 1 MPa and a temperature of 40C. A leak develops and refrigerant slowly escapes until the pressure reaches 100 kPa. Determine the final temperature in the tank (C) and the amount of mass that leaves the tank (kg).

#### *Solution*

The Excel solution is seen in the screen shot in Figure 4. Note the reporting format implemented in the example solution in Figure 4 (Woodbury, 2008). This reporting format is very clear and allows the student and reader to easily understand the approach and solution to the problem. The report format uses the first column in the document as the “Name” column where students assign each variable a name, such as T\_1, to be used in Excel’s Name Manager. The second column is where the actual calculation or given variable is input. The third column in the document is used to denote the units of the variable. Lastly, the Excel equations used to arrive at the answer in the second column is shown by placing an apostrophe (tic-mark) in front of the equation and inserted into the fourth column. As evident by this example, the reporting format organizes the solution very concisely.

Note that the function x\_ps\_R134a makes calculation of the final quality very simple. A conventional approach to finding v\_2 has been taken here, but it can be found directly once it is recognized that the final state is fixed by P\_2 and s\_1. The final specific volume could be computed directly from the function v\_Ps\_R134a instead.

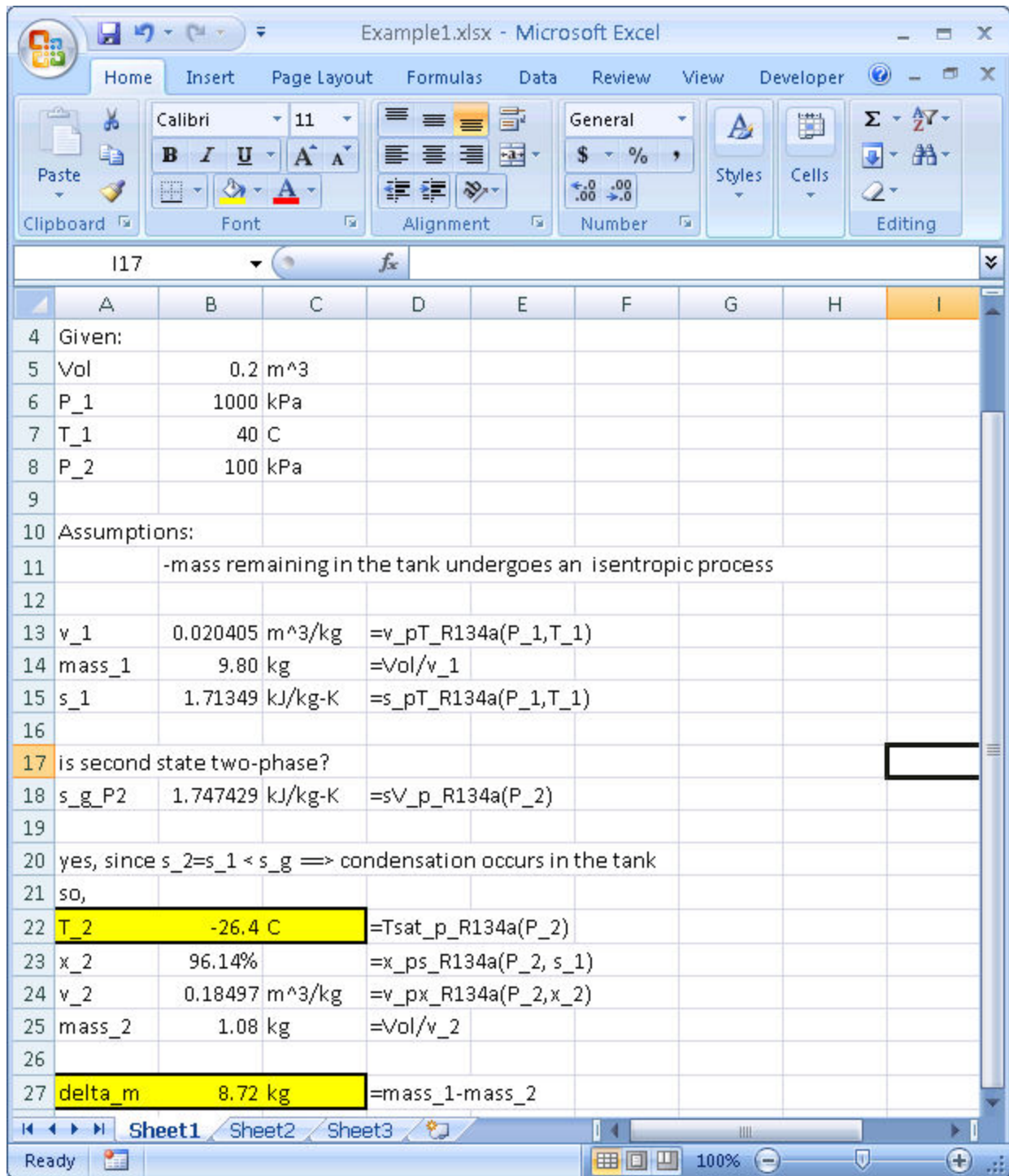


Figure 4. Solution of R134a example.

## R22

### Description

#### *Why it is needed*

Although it is being phased out because of detrimental effects on the ozone layer, R22 is still widely used as a refrigerant. As well, some thermodynamics textbooks still have problems written for R22. Thus, it is desirable to have functions for computation of R22 properties.

During the development of the R134a module, an earlier paper giving the equation of state for R22 was found (Wagner et al. (1993)). The structure of the equation of state for both refrigerants R22 and R134a is very similar, so implementation of the R22 module by modification of the R134a module was very straightforward.

### Implementation

The R134a module was used as a model to code the R22 module. Each instance of `_R134a` found in the R134a module was replaced with `_R22` to insure a consistent naming convention. Constants pertaining to R134a, such as critical temperature, critical density, and maximum temperature and pressure, were replaced with the appropriate R22 properties. The only difference in the R22 and R134a equations of state are the number of constants used and their values. At this point, the implementation process for the R22 and R134a modules are the same because the initial primary equations provided in the R134a and R22 papers were the same.

### Evaluation of Functions

#### *Methods for testing*

For the testing of the R22 module, REFPROP (NIST, 2008) was used to determine twenty-five random thermodynamic states from the saturation, superheated, and supercritical regions. REFPROP 8.0 is the NIST reference fluid thermodynamic and transport properties database which provides properties for a wide range of fluids. The process for testing the R22 module was improved due to lessons learned from the testing of the R134a module. Instead of copying and pasting the actual table values into each appropriate cell in the testing tables, a single table was made of all twenty-five thermodynamic states and all their properties. As can be seen in Figure 5, appropriate names were applied to the columns and rows of the table so that Excel's Name Manager could be used to assign a descriptive, dynamic name to each table value. For example, to call the saturated vapor entropy of the first saturation state in the table `"=(sat1 s_g)"` would be typed in the testing tables. By using dynamic names in the testing tables, instead of copying and pasting all table values into hundreds of locations in the testing tables, the possibility of data entry error was substantially minimized. This new testing technique drastically decreased the testing time of the R22 module. Also since the table values from REFPROP were only entered in one location, correcting possible data entry errors or checking additional states became a very simple process. Once again a goal of only 0.5 percent error in R22 functions was set for all functions.

Data Points from RefProps																
	Temp	Pbar	PMPa	PkPa	v_f	v_g	v_50	u_f	u_g	u_50	h_f	h_g	h_50	s_f	s_g	s_50
props	T	P <sub>sat</sub>	P <sub>sat</sub>	P	v <sub>f</sub>	v <sub>g</sub>	v <sub>50</sub>	u <sub>f</sub>	u <sub>g</sub>	u <sub>50</sub>	h <sub>f</sub>	h <sub>g</sub>	h <sub>50</sub>	s <sub>f</sub>	s <sub>g</sub>	s <sub>50</sub>
units	(°C)	(bar)	(MPa)	(kPa)	(m³/kg)	(m³/kg)	(m³/kg)	(kJ/kg)	(kJ/kg)	(kJ/kg)	(kJ/kg)	(kJ/kg)	(kJ/kg)	(kJ/(kg·K))	(kJ/(kg·K))	(kJ/(kg·K))
sat1	-150	0.000019337	1.9337E-06	0.001934	0.000588	6123.699	3061.85	37.558	324.0586	180.8083	37.558	335.9	186.729	0.14276	2.5653	1.35403
sat2	-125	0.0012237	0.00012237	0.12237	0.000611	116.3914	58.19601	64.16293	332.8672	198.5101	64.163	347.1	205.6315	0.33947	2.2492	1.294335
sat3	-100	0.020102	0.0020102	2.0102	0.000636	8.265829	4.133233	90.70372	342.354	216.5289	90.705	358.97	224.8375	0.50502	2.0543	1.27966
sat4	-75	0.14719	0.014719	14.719	0.000665	1.28289	0.641777	117.2402	352.3571	234.7987	117.25	371.24	244.245	0.64819	1.93	1.289095
sat5	-50	0.6453	0.06453	64.53	0.000697	0.323845	0.162271	143.9851	362.5223	253.2537	144.03	383.42	263.725	0.77525	1.848	1.311625
sat6	-25	2.0143	0.20143	201.43	0.000734	0.11163	0.056182	171.2921	372.4145	271.8533	171.44	394.9	283.17	0.89125	1.7918	1.341525
sat7	0	4.9799	0.49799	497.99	0.00078	0.047105	0.023943	199.6114	381.592	290.6017	200	405.05	302.525	1	1.7507	1.37535
sat8	25	10.439	1.0439	1043.9	0.00084	0.022608	0.011724	229.4133	389.4294	309.4214	230.29	413.03	321.66	1.1045	1.7174	1.41095
sat9	50	19.427	1.9427	1942.7	0.000924	0.011634	0.006279	261.455	394.8378	328.1464	263.25	417.44	340.345	1.208	1.6852	1.4466
sat10	75	33.177	3.3177	3317.7	0.00107	0.005947	0.003508	297.9092	394.7606	346.3349	301.46	414.49	357.975	1.3177	1.6424	1.48005
sat11	90	44.423	4.4423	4442.3	0.001282	0.003563	0.002423	326.3955	386.0403	356.2179	332.09	401.87	366.98	1.4001	1.5922	1.49615
sup1	10	4.5	0.45	450		0.055488			388.3405			413.31				1.7892
sup2	50	4.5	0.45	450		0.065535			412.4692			441.96				1.8839
sup3	10	5	0.5	500		0.049383			387.6386			412.33				1.7765
sup4	50	5	0.5	500		0.058613			412.0234			441.33				1.8723
sup5	-25	0.8	0.08	80		0.291698			374.5841			397.92				1.8895
sup6	0	0.8	0.08	80		0.323185			387.6152			413.47				1.9491
sup7	-25	1	0.1	100		0.232008			374.2492			397.45				1.8666
sup8	0	1	0.1	100		0.2575			387.35			413.1				1.9267
sup9	70	12	1.2	1200		0.024292			418.7497			447.9				1.8149
sup10	90	12	1.2	1200		0.026369			432.4278			464.07				1.8607
sup11	80	16	1.6	1600		0.018173			422.2134			451.29				1.8006
sup12	120	16	1.6	1600		0.021393			450.7617			484.99				1.891
scrit1	80	18	1.8	1800		0.01576			420.4117			448.78				1.7839
scrit2	70	24	2.4	2400		0.01006			405.7459			429.89				1.7077
scrit3	100	24	2.4	2400		0.012219			430.2648			459.59				1.7908
scrit4	150	24	2.4	2400		0.015095			468.2114			504.44				1.9037
scrit5	180	24	2.4	2400		0.016639			490.9652			530.9				1.9641
scrit6																

Figure 5. R22 Functions Checks Table from RefProp

During testing, attention was given to making additional changes in the R22 code to reduce the run time of the functions and to increase the uniformity of the module. Some of the reasons to have unnecessarily long run times could be due to superfluous iterations and functions calls. The first action taken to increase the speed of each calculation was to reduce the coded steps found in the derivative equations of the Helmholtz Free Energy equations that are called by almost every function in the module. Tau, defined as the critical temperature divided by the temperature of the state, and Delta, defined as the density of the state divided by the critical density, were calculated an excessive number of times within each derivative equation. Private functions were introduced to calculate Tau and Delta and were inserted before iterations began in the derivative functions to reduce the runtime of the functions and to make the module more uniform. All improvements made to the R22 module to reduce calculation time and to improve uniformity were also made in the R134a module so that consistency would exist between all modules.

### Results of Module

A complete list of all primary, secondary, and tertiary functions and their RMS error values in the R22 module can be seen in Table 3. The RMS error value represents the root mean square error value for each function. This value shows the accuracy of each function when compared to the REFPROP table values. Seventy-four functions were included in the R22 module. The other seventeen functions considered were excluded because of inconsistencies in results and excessive runtimes.

Similar to the R134a module, the results of the R22 module were very accurate. Almost all functions fell within the desired 0.5 percent error criteria for accuracy. The RMS error values were all reasonable except for very large error values for some of the specific volume functions. These extreme RMS error values are highlighted in yellow in Table 3. The large error values are due to the strong sensitivity of specific volume in the saturated liquid region.

### R22 Example

#### *Problem*

Refrigerant 22 is compressed steadily from saturated liquid at 50 psia to 200 psia with an isentropic efficiency of 88%. Determine the specific work (Btu/lbm) required. What is the temperature of the R22 leaving the compressor?

#### *Solution*

This problem is in US Customary units so the optional argument “ENG” is given to all the function calls to utilize these units. The solution is seen in the screen shot in Figure 6. Note that the four-column format is adhered to, with the actual formulas used to compute the values cut-and-pasted as visible text strings in the fourth column.

<b>Table 3. R22 Functions with RMS Error Values</b>					
<b>Primary</b>	<b>RMS</b>	<b>Secondary</b>	<b>RMS</b>	<b>Tertiary</b>	<b>RMS</b>
<b>h_px_R22</b>	0.0026	<b>h_sx_R22</b>	Abandoned	<b>h_prho_R22</b>	0.0023
<b>h_Tv_R22</b>	0.0064	<b>h_vx_R22</b>	Abandoned	<b>h_ps_R22</b>	0.0016
<b>h_Tx_R22</b>	0.0036	<b>p_Th_R22</b>	0.0115	<b>h_pT_R22</b>	0.0001
<b>hL_p_R22</b>	0.0043	<b>p_Ts_R22</b>	0.0103	<b>h_pv_R22</b>	0.0023
<b>hL_T_R22</b>	0.0061	<b>p_Tu_R22</b>	0.0124	<b>h_sv_R22</b>	Abandoned
<b>hV_p_R22</b>	0.0004	<b>s_vx_R22</b>	Abandoned	<b>h_Ts_R22</b>	0.0016
<b>hV_T_R22</b>	0.0004	<b>T_hv_R22</b>	0.0037	<b>h_Tu_R22</b>	0.0003
<b>p_Tv_R22</b>	0.0111	<b>T_ph_R22</b>	0.0010	<b>p_hs_R22</b>	Abandoned
<b>psat_T_R22</b>	0.0143	<b>T_ps_R22</b>	0.0012	<b>p_hv_R22</b>	0.0190
<b>rhoL_T_R22</b>	0.0002	<b>T_pv_R22</b>	0.0014	<b>p_sv_R22</b>	Abandoned
<b>rhoV_T_R22</b>	0.0103	<b>Tsat_p_R22</b>	0.0007	<b>s_hv_R22</b>	0.0031
<b>s_px_R22</b>	0.0014	<b>u_sx_R22</b>	Abandoned	<b>s_ph_R22</b>	0.0031
<b>s_Tv_R22</b>	0.0063	<b>u_vx_R22</b>	Abandoned	<b>s_pT_R22</b>	0.0001
<b>s_Tx_R22</b>	0.0031	<b>v_ps_R22</b>	225.7018	<b>s_pv_R22</b>	0.0013
<b>sL_p_R22</b>	0.0023	<b>v_pT_R22</b>	0.0005	<b>s_Th_R22</b>	0.0032
<b>sL_T_R22</b>	0.0053	<b>v_sx_R22</b>	Abandoned	<b>s_Tu_R22</b>	0.0030
<b>sV_p_R22</b>	0.0003	<b>v_Th_R22</b>	2971.1509	<b>T_hs_R22</b>	Abandoned
<b>sV_T_R22</b>	0.0005	<b>v_Ts_R22</b>	1437.7998	<b>T_sv_R22</b>	Abandoned
<b>u_px_R22</b>	0.0025	<b>v_Tu_R22</b>	0.2799	<b>T_uv_R22</b>	Abandoned
<b>u_Tv_R22</b>	0.0063			<b>u_hs_R22</b>	Abandoned
<b>u_Tx_R22</b>	0.0034			<b>u_hv_R22</b>	0.0002
<b>uL_p_R22</b>	0.0042			<b>u_ph_R22</b>	0.0002
<b>uL_T_R22</b>	0.0059			<b>u_ps_R22</b>	0.0014
<b>uV_p_R22</b>	0.0003			<b>u_pT_R22</b>	0.0001
<b>uV_T_R22</b>	0.0003			<b>u_pv_R22</b>	0.0022
<b>v_px_R22</b>	0.0026			<b>u_sv_R22</b>	Abandoned
<b>v_Tx_R22</b>	0.0086			<b>u_Th_R22</b>	0.0002
<b>vL_p_R22</b>	0.0003			<b>u_Ts_R22</b>	0.0015
<b>vL_T_R22</b>	0.0002			<b>v_hs_R22</b>	Abandoned
<b>vV_p_R22</b>	0.0032			<b>v_ph_R22</b>	2066.4173
<b>vV_T_R22</b>	0.0106			<b>x_hs_R22</b>	Abandoned
<b>x_ph_R22</b>	0.0775			<b>x_hv_R22</b>	0.0013
<b>x_ps_R22</b>	0.0623			<b>x_sv_R22</b>	Abandoned
<b>x_pu_R22</b>	0.0866				
<b>x_pv_R22</b>	0.0031				
<b>x_Th_R22</b>	0.0969				
<b>x_Ts_R22</b>	0.0789				
<b>x_Tu_R22</b>	0.1068				
<b>x_Tv_R22</b>	0.0156				

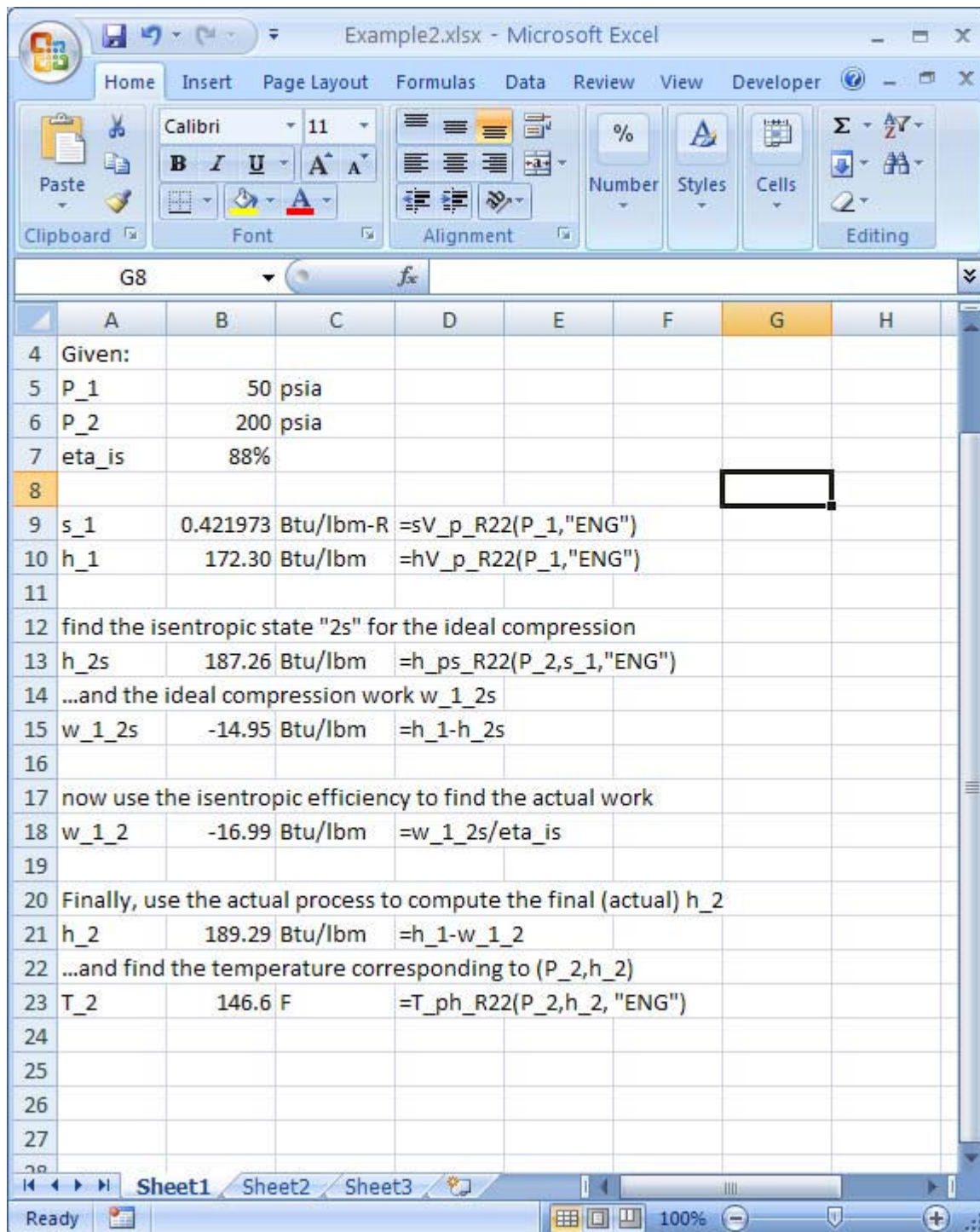


Figure 6. Example R22 Problem.



## Gas Dynamics

### Description

Algebraic equations for compressible flow of an ideal gas are quite amenable to development of an Excel library of functions. Such a collection of functions provides an excellent substitute for the tables of compressible flow functions in the appendix of many gas dynamics textbooks.

Availability of such a library enables engineers to perform compressible flow calculations in the spreadsheet environment. Functions for isentropic flow, normal shock, oblique shock, Prandtl Meyer expansion, Fanno flow, and Rayleigh flow can readily be implemented.

### Implementation

Basic equations for compressible flow are readily available in most thermodynamics textbooks.

The equations used for the present development were actually taken from John and Keith (2006).

The relations were easy to code into Visual Basic for Applications (VBA) and were wrapped into an Add-in module similar to the other suites. A list of the functions that are available in the module is seen in Table 4.

### Gas Dynamics Example

#### *Problem*

Air ( $\gamma=1.4$ ) expands from a storage tank through a converging-diverging nozzle having a throat area of  $50 \text{ cm}^2$ . The conditions in the tank are  $P=200 \text{ kPa}$  and  $T = 300 \text{ K}$ . A normal shock stands in the diverging portion of the nozzle at a location where  $A=100 \text{ cm}^2$ . The exit area of the nozzle is  $200 \text{ cm}^2$ . Find: a)  $A^*$  from the tank to the shock location, b)  $A^*$  from the shock to the exit, c) the Mach number at the exit, d) stagnation pressure at exit, e) exit plane static pressure.

#### *Solution*

The solution to the problem is seen in the screen shots in Figures 7 and 8. Note the use of the Excel “Goalseek” capability to do the “reverse lookup” to find the Mach number corresponding to a known  $A/A^*$  ratio in part b).

Table 4 - Gas Dynamics Excel Add-in functions

Iisentropic Suite

T_T0(M, $\gamma$ )	static to total temperature ratio
P_P0(M, $\gamma$ )	static to total pressure ratio
den_den0(M, $\gamma$ )	density to density at total temperature and pressure
A_Astar(M, $\gamma$ )	area ratio (Astar is the area corresponding to Mthroat = 1)
Rho_V(P0, T0, $\gamma$ , M, R)	mass flux at given mach number and stagnation conditions

Standing Normal Shock Suite

SNS_M2(M1, $\gamma$ )	mach number downstream of a normal shock.
SNS_T2_T1(M1, $\gamma$ )	static temperature ratio across a normal shock.
SNS_P2_P1(M1, $\gamma$ )	static pressure ratio across a normal shock.
SNS_den2_den1(M1, $\gamma$ )	density ratio across a normal shock.
SNS_P02_P01(M1, $\gamma$ )	stagnation pressure ratio across a normal shock.
SNS_Mexit(Pb_P01, Ae_At, $\gamma$ )	exit mach number when a normal shock stands

Oblique Shock Suite

OBS_M2(M1, $\theta$ , $\gamma$ )	mach number downstream of an oblique shock.
OBS_delta(M1, $\theta$ , $\gamma$ )	turning angle in degrees –Has Units.
OBS_deltamax(M1, $\gamma$ )	maximum turning angle in degrees

Prandtl-Meyer Function

PMF_nu(M, $\gamma$ )	Prandtl-Meyer function v in degrees
----------------------	-------------------------------------

Fanno-Flow Suite

Fan_fL_D(M1, M2, $\gamma$ )	friction relative length.
Fan_T1_T2(M1, M2, $\gamma$ )	static temperature ratio.
Fan_P1_P2(M1, M2, $\gamma$ )	static pressure ratio.
Fan_P01_P02(M1, M2, $\gamma$ )	stagnation pressure ratio.
Fan_rho1_rho2(M1, M2, $\gamma$ )	density ratio

Rayleigh-Flow Suite

Ray_T1_T2(M1, M2, $\gamma$ )	static temperature ratio.
Ray_P1_P2(M1, M2, $\gamma$ )	static pressure ratio.
Ray_P01_P02(M1, M2, $\gamma$ )	stagnation pressure ratio.
Ray_T01_T02(M1, M2, $\gamma$ )	stagnation temperature ratio.
Ray_rho1_rho2(M1, M2, $\gamma$ )	density ratio.

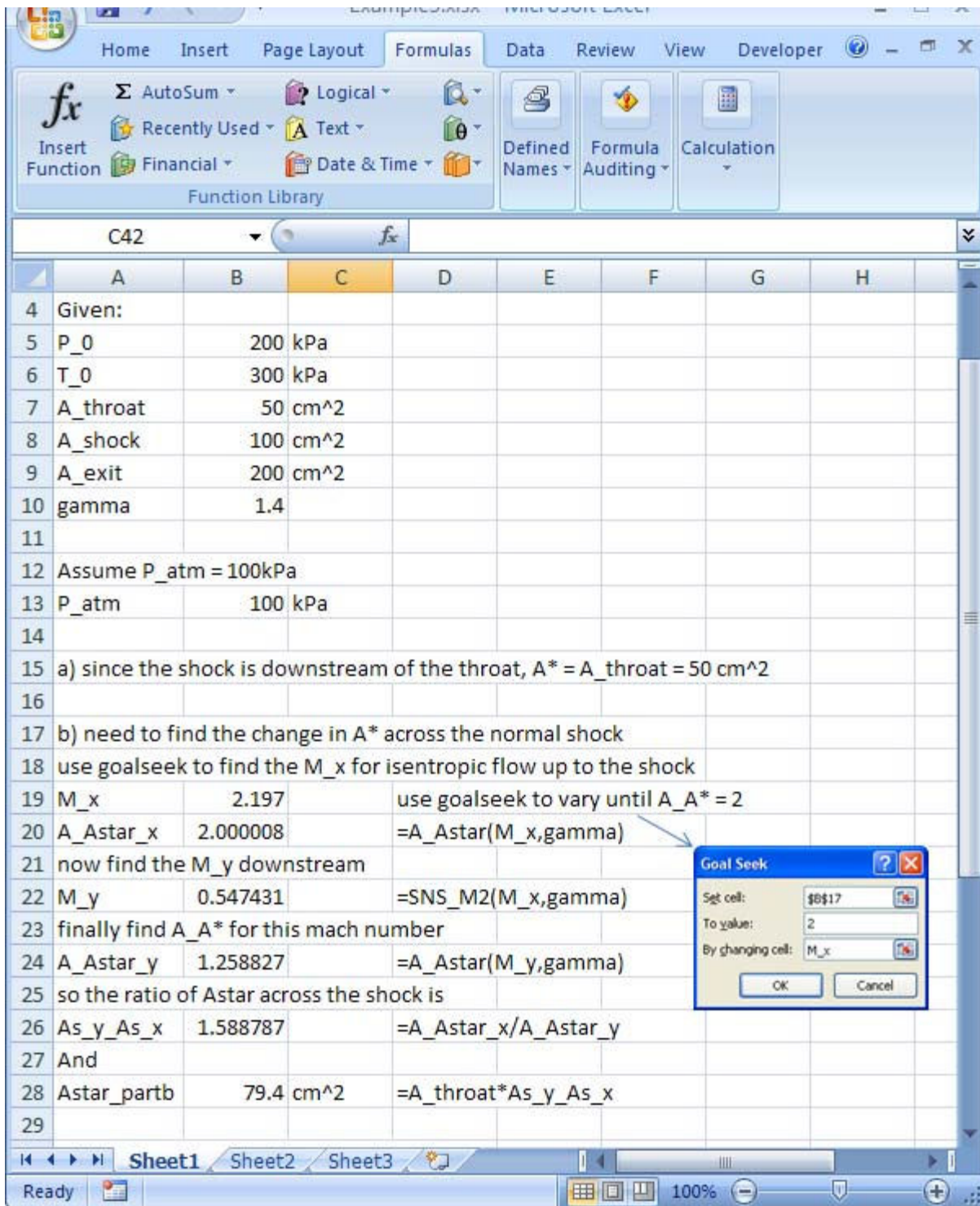


Figure 7 – Gas Dynamics example solution part a) and b)

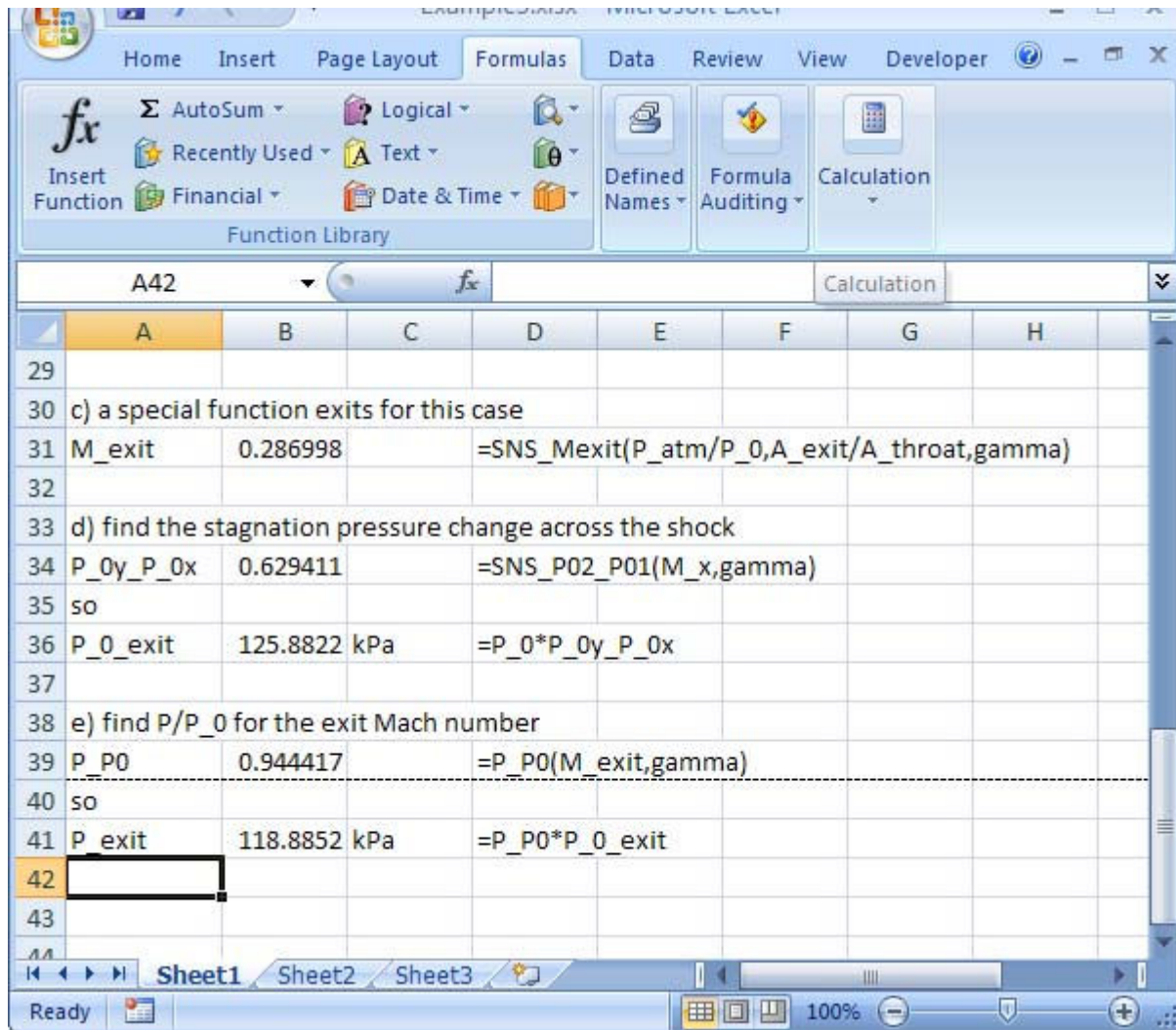


Figure 8 Gas Dynamics example solution – parts c) – e)

### Conclusions

With the addition of the R134a, R22 and Gas Dynamics modules, the potential use of the Excel add-ins developed under the NSF project “Excel in ME” has been significantly expanded. The R134a and R22 modules were developed using well-known fundamental equations of state from the literature for each refrigerant. These equations were used to develop a set of primary functions in Visual Basic which were then expanded by adding additional functions that iterate using the bisection method. The R134a and R22 modules are useful to teachers and students as they allow work in Excel without the necessity of looking up values in tables. A Gas Dynamic module was developed using well-established algebraic relations for flow of an ideal gas. These functions enable solution of compressible flow problems in the spreadsheet environment. All of the Excel Add-ins developed can be downloaded at the project website [www.me.ua.edu/ExcelinME](http://www.me.ua.edu/ExcelinME).

## Acknowledgement



This material is based upon work supported by the National Science Foundation under Grant No. DUE-0633330. The authors gratefully acknowledge support from this NSF award.

## Disclaimer

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

- ASHRAE, (2005), *Handbook of Fundamentals*, <http://www.ashrae.org/>
- Chappell, J., Taylor, R. P., and Woodbury, K. A. (2008) "Introducing Excel-based Steam Table Calculation into Thermodynamics Curriculum," 2008 ASEE Annual Conference & Exposition, June 22 - 25 - Pittsburgh, PA
- Holmgren, M., (2007) Excel-Engineering website. <http://www.x-eng.com/>
- Huguet, J., Taylor, R. P., and Woodbury, K. A. (2008) "Development of Excel Add-in Modules for Use in Thermodynamics Curriculum: Steam and Ideal Gas Properties," 2008 ASEE Annual Conference & Exposition, June 22 - 25 - Pittsburgh, PA
- John, J. E. A., and Keith, T., (2006) *Gas Dynamics*, Third Edition, Prentice-Hall
- NIST (2008), Ref Props 8, NIST Standard Reference Database 23, <http://www.nist.gov/srd/nist23.htm>
- Tillner-Roth, R., and Baehr, H. D. (1994) "An International Standard Formulation for the Thermodynamic Properties of 1,1,1,2-Tetrafluoroethane (HFC-134a) for Temperature From 170 K to 455 K and Pressures up to 70 MPa," *Journal of Physical and Chemical Reference Data*, Vol. 23, No. 5
- Wagner, W., Marx, V., and Prub, A., (1993) "A new equation of state for chlorodifluoromethane (E22) covering the entire fluid region from 116K to 550K at pressures up to 200 MPa," *Rev. Int. Froid* Vol. 16, No. 6
- Woodbury, K. A., Taylor, R. P., et al. (2008) "Vertical Integration of Excel in The Thermal Mechanical Engineering Curriculum," 2008 ASME International Mechanical Engineering Congress & Exposition, October 31- November 6, 2008, Boston, Massachusetts