Spreadsheet Calculations of Thermodynamic Properties

Larry Caretto, David McDaniel, Tom Mincer
CSU, Northridge/US Air Force Academy/CSU, Northridge

Abstract

A thermodynamic property calculation tool has been developed based on the Excel™ spreadsheet program. This tool can be used to determine individual state points by a graphical user interface (GUI) calculator. In addition, spreadsheet cell formulas and Visual Basic function calls can also be used to build models of thermodynamic systems. The resulting model can be used to show the effects of parameter changes on overall results such as cycle efficiency, power production, refrigeration load and coefficient of performance. In addition, the Excel™ GoalSeek or Solver tools can be used to solve systems of equations or to determine system parameters that give a constrained optimum condition. The software is available at no cost.

Introduction

Computer calculation of thermodynamic properties is an efficient replacement for property tables, which are still used in thermodynamics instruction. Computerized property calculations are used routinely in engineering applications. They are available from a variety of sources, such as the National Institute of Standards and Technology.[1]

Computer property calculations are a part of current thermodynamics texts. The book by Çengel and Boles[2] includes an academic version of the Engineering Equation Solver (EES) that allows students to formulate thermodynamic problems as a set of equations that may involve thermodynamic property calculations. EES is then able to solve a properly posed problem. It can also be used for finding an individual property at a single state. The texts by Sonntag *et al*.[3] and Moran and Shapiro[4] include programs that can find state properties or generate tables.

Several recent ASEE papers have discussed the use of computer property calculations in thermodynamics instruction. Dixon[5] suggested that it is time to teach thermodynamics with only computer property calculations. He noted that the U.S. Coast Guard Academy introduces EES in the first thermodynamics course, and concluded that the "the value of utilizing the program capabilities makes the learning effort well worthwhile." Ngo and Lai[6] discussed web-based tools for computing thermodynamic properties and learning how to use tables. Aung[7]described three programs, CyclePad, PsyCalc, and Gaseq, which solve thermodynamic problems with property calculations included. Hudson[8] discussed the required use of laptop computers, which include thermodynamic property calculation programs, at Mississippi State University. Somerton *et al*.[9] developed a MATLAB toolbox for thermodynamic property calculations.

Goodwin[10] developed a set of spreadsheet calculations of thermodynamic properties. The application of this package is similar to the one described here, although the functional notation is different and fewer substances are available. Goodwin's package includes templates that students can use to prepare plots for the results of their problems that show the saturation curve.

The program described here started as a mainframe thermodynamic calculation code,[11] written in Fortran, based on the data and computational procedures outlined by Reynolds.[12] Subsequent versions used text input and output on a desktop computer and then used a Windows™ interface for input and results before the development of the spreadsheet program described here.

In comparison to the programs discussed above, the program presented here was designed to provide easy-to-use thermodynamic properties calculations within the Excel™ platform, which is widely available on personal and office computers. The combination of a graphical interface and a simple transition from that interface to cell formulas provides a wide range of applications from simple calculations of a single state using the graphical interface to detailed models of thermodynamic systems using cell formulas. There is a cell formula insertion procedure that eases the transition from the graphical interface to cell formulas. The application of this package does not require the casual Excel™ user to learn any new features; however, it provides experienced users with the ability to build, solve and optimize complex thermodynamic models in both cell formulas and Visual Basic macros.

Discussion

This section begins with a discussion of the actual use of the toolbox for property calculations, followed by a discussion of the underlying computations and computer code and concluding with a discussion of the classroom use of this tool.

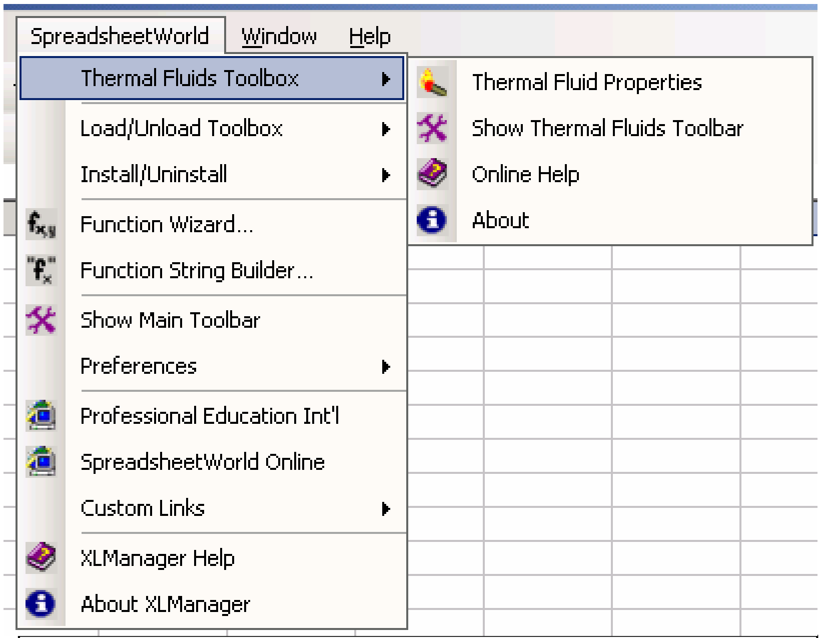
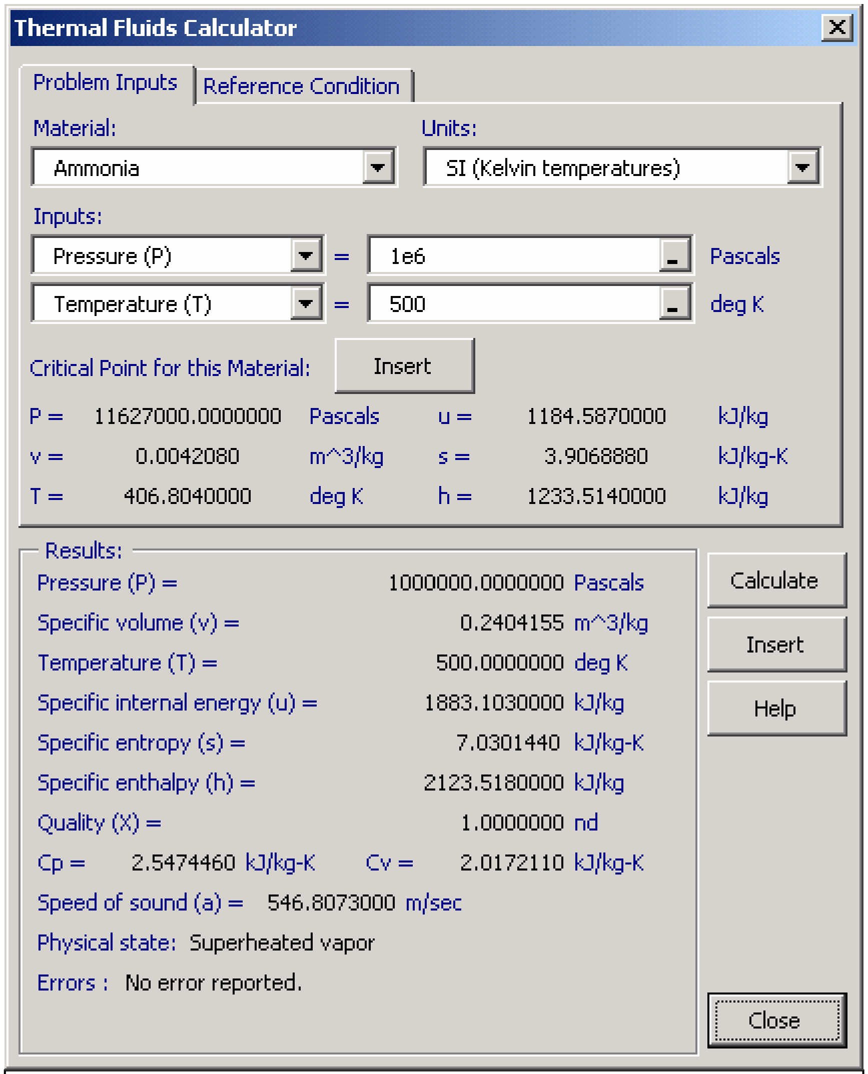


Figure 1. Menu Options for Thermal Fluids Toolbox

Using the toolbox – The Thermal Fluids Toolbox is one of a series of toolboxes available from SpreadsheetWorld, Inc. (www.spreadsheetworld.com). Before installing the toolbox, it is necessary to install a utility application called XLManager that manages the entire toolbox suite. After this manager is installed, the Thermal Fluids Toolbox can then be installed. Once the toolbox is installed, a new SpreadsheetWorld menu is available on the main Excel™ menu. Figure 1 shows a version of Excel with the Thermal Fluids Toolbox installed. A new

SpreadsheetWorld menu is available. (Adobe PDF users will recognize the similarity to the Adobe menu that appears in Excel when the full Acrobat suite is installed.) Calculations with the Thermal Fluids Tool box are started from this menu and the corresponding submenu.

Selecting "Thermal Fluid Properties" from the submenu shown in Figure 1 brings up the calculator shown in Figure 2. The top of the calculator accepts the user input, and the results are shown at the bottom. The user has pull-down menus available from which the following items may be specified: (1) the fluid, (2) the unit system, and (3) the variables used to specify the state. A list of available fluids is shown in Table 1. Units may be specified as SI units or engineering units. In SI units, the temperature can be in degrees Celsius or kelvins; in engineering units, the temperature can be in degrees Fahrenheit or Rankine.[*] The user can select any pair of properties from the following set to specify the state: pressure, volume, temperature, internal energy, entropy, enthalpy, and quality. Once the input variables have been selected by the user, the appropriate units appear to the right of the text boxes provided for the user input.



Material: Ammonia

Units: SI (Kelvin temperatures)

Inputs:

Pressure (P) = 1e6 Pascals

Temperature (T) = 500 deg K

Critical Point for this Material:

P =	11627000.0000000	Pascals	u =	1184.5870000	kJ/kg
v =	0.0042080	m^3/kg	s =	3.9068880	kJ/kg-K
T =	406.8040000	deg K	h =	1233.5140000	kJ/kg

Results:

Pressure (P) =	1000000.0000000	Pascals
Specific volume (v) =	0.2404155	m^3/kg
Temperature (T) =	500.0000000	deg K
Specific internal energy (u) =	1883.1030000	kJ/kg
Specific entropy (s) =	7.0301440	kJ/kg-K
Specific enthalpy (h) =	2123.5180000	kJ/kg
Quality (X) =	1.0000000	nd

Cp = 2.5474460 kJ/kg-K Cv = 2.0172110 kJ/kg-K

Speed of sound (a) = 546.8073000 m/sec

Physical state: Superheated vapor

Errors : No error reported.

Figure 2. Thermal Fluids Property Calculator Dialog

Figure 2 shows the results of calculations for ammonia using SI units with temperatures in kelvins. Here the user has entered P = 1 MPa (in the program input unit of pascals) and T =

[*] SI units use kJ and kg for energy and mass units and Pa for pressure units. Engineering units use Btu and lb_m for energy and mass units and psia for pressure units. One reviewer questioned the use of Pa rather than kPa for SI pressure units. This choice was made because, unlike energy units which are almost always in kJ/kg, pressure units are often in MPa rather than kPa. The decision to use Pa as the input unit for pressure was intended to force users to recognize the actual pressure units and enter values such as 10 kPa or 10 MPA as 10e3 or 10e6, respectively.

500 K. Once the input data are entered, the user clicks the calculate button (in the middle of the dialog on the right) and the results are shown at the bottom of the calculator.

The properties calculator retains information from the previous inputs for subsequent user input. The user need only enter new changed data for a new calculation. The calculator is an effective tool for finding a small number of property data that are used in a hand calculation.

Table 1 – Table of Available Substances and their Identifiers

Substance[1]	ID	Substance	ID	Substance	ID
Air[2]	1	Isopentane	14	Refrigerant 13	28
Ammonia	2	Lithium	15	Refrigerant 14	29
Argon	3	Mercury	16	Refrigerant 22	30
Butane	4	Methane	17	Refrigerant 23	31
Cesium	5	Neon	18	Refrigerant 123	32
Carbon Dioxide	6	Nitrogen	19	Refrigerant 500	33
Ethane	7	Octane	20	Refrigerant 502	34
Ethylene	8	Oxygen	21	Refrigerant 503	35
Helium	9	Potassium	22	Refrigerant C318	36
Heptane	10	Propane	23	Refrigerant 134a	37
Hexane	11	Propyl Alcohol	24	Rubidium	38
Hydrogen	12	Propylene	25	Sodium	39
Isobutane	13	Refrigerant 11	26	Water	40
		Refrigerant 12	27		

Notes:

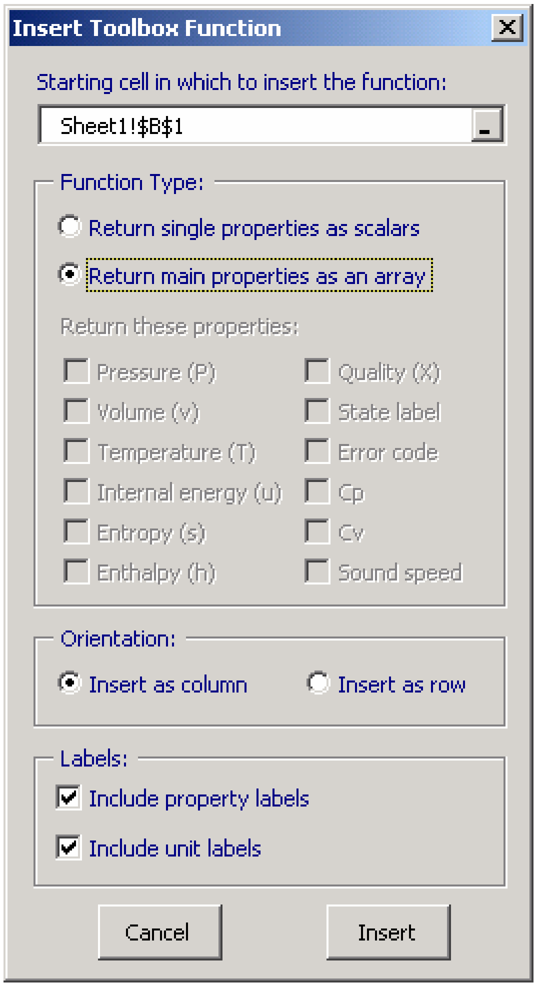
[1] The names to be used in spreadsheet formulas are the same as those shown above except that propyl alcohol is entered as "Pro-Alcohol" and refrigerants are entered as "R-" preceding the number (*e.g.,* Refrigerant 134a would be entered as "R-134a" or by the number 37).

[2] Calculations are available only for gas-phase properties of air (1).

The toolbox may also be used by entering formulas directly onto the spreadsheet. This offers maximum flexibility when analyzing complex systems. The use of formulas is enhanced by the provision in the calculator that automatically inserts cell formulas for the user. This provides a working example of a formula that users can copy and edit for their own use.

As shown in Figure 2, the calculator has two insert buttons. The first, near the top, inserts one of two formulas capable of computing the critical properties (TFCritProps and TFCritProp). The lower one, just below the calculate button, inserts the cell formula for the state results from the user-selected input conditions. Clicking this insert button brings up the insert dialog shown in Figure 3. This dialog allows the user to specify the kind of cell formula(s) desired and the location where they will be entered.

There are two basic kinds of property formulas. One kind of formula returns a single property. Multiple versions of this formula can be used to get more than one property for a given state. The other type of formula returns all the property data, including a description of the state and an error code. The second kind of formula is known as an array formula in Excel™.[*] This formula has the advantage of providing all properties with only one formula; however, it has the disadvantage of placing many unneeded results on the spreadsheet. The insertion dialog in Figure 3 has selected the option of inserting all properties as an array formula, with the property values starting in cell B1. (The labels are placed in column A and the units are placed in column C.) Figure 4 shows the resulting spreadsheet after inserting the problem from Figure 2. The entire output is contained in cells A1 to C9. The numeric results start in cell B1 as specified in the insertion dialog. Cells B1 to B9 in Figure 4 contain the following array formula:



Insert Toolbox Function

Starting cell in which to insert the function:

Sheet1!B1

Function Type:

○ Return single properties as scalars

● Return main properties as an array

Return these properties:

□ Pressure (P)	□ Quality (X)
□ Volume (v)	□ State label
□ Temperature (T)	□ Error code
□ Internal energy (u)	□ Cp
□ Entropy (s)	□ Cv
□ Enthalpy (h)	□ Sound speed

Orientation:

● Insert as column ○ Insert as row

Labels:

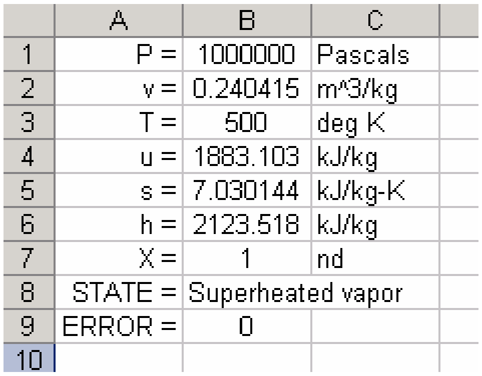
☑ Include property labels

☑ Include unit labels

Figure 3. Formula Insertion Dialog

=TFProps("Ammonia","SI_K","P"
,1000000,"T",500) [1]

The function name, TFProps, is the array function that returns all the property calculation results as an array. The arguments of this function give the following information from left to right: the name of the substance as shown in Table 1, a string identifier for the units,[†] the one-character string "P" indicating the next numeric value is the input value for pressure, the value of 100000 for a pressure of 1 MPA, the string "T" indicating that the next numeric value is the input for temperature, and the value 500 for T = 500 K.



	A	B	C
1	P =	1000000	Pascals
2	v =	0.240415	m^3/kg
3	T =	500	deg K
4	u =	1883.103	kJ/kg
5	s =	7.030144	kJ/kg-K
6	h =	2123.518	kJ/kg
7	X =	1	nd
8	STATE =	Superheated vapor	
9	ERROR =	0	
10			

Figure 4. TFProps Formula Inserted on Worksheet

[*] Such formulas provide results that occupy more than one cell. These formulas are entered by selecting the desired number of cells where the results are to appear and then the formula is typed once to appear in all cells.

[†] The possible unit identifiers are SI_C, SI_K, EE_F, and EE_R. The first two characters indicate the unit system as English engineering units or SI units. The final character specifies the units for temperature. Alternatively, a numeric code (1 for SI_C, 2 for SI_K, 3 for EE_F and 4 for EE_R) may be used. The numeric identifiers in Table 1 may also be used in place of string names for the substance whose properties are desired.

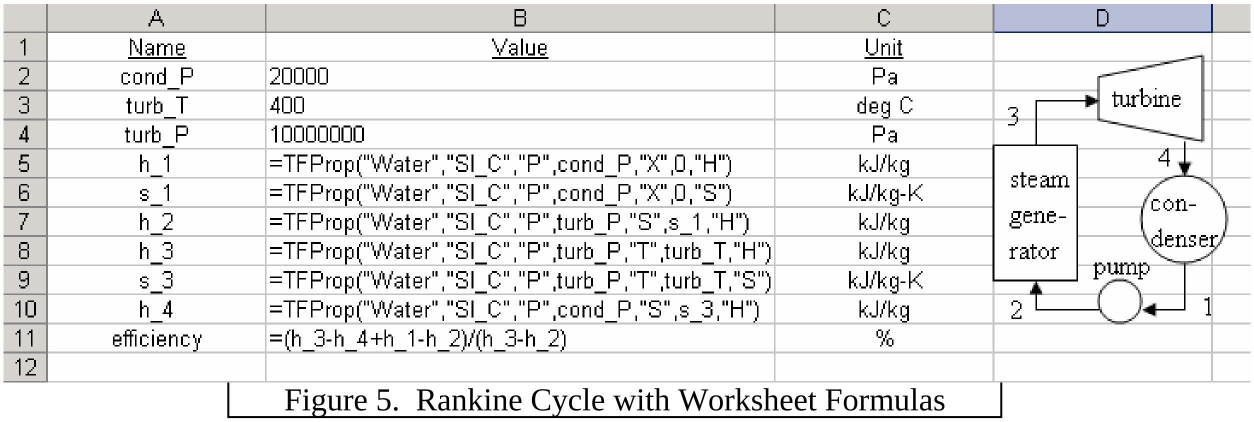
If the user had chosen to insert individual property results instead of the entire array in the insertion dialog, one or more single-cell formulas would be inserted in the spreadsheet. The single cell formula has the name TFProp (as opposed to the plural TFProps for all properties). This function has the same first arguments as the TFProps function, but a final string argument is added to specify the particular quantity to be inserted. For example, the following formula would be used to calculate the enthalpy only for the same conditions used previously:

=TFProp("Ammonia","SI_K","P",1000000,"T",500,”H”) [2]

The cell formulas shown in equations [1] and [2] are the ones that would be inserted into a spreadsheet for the example shown in Figure 2. In these formulas, which would be prepared by the input dialog in Figure 3, the values are constants. These inserted formulas can be used as the first step in developing a more complex model. The formulas can be copied, pasted and edited to replace the constant values in the example formulas by references to other spreadsheet cells that contain the input values. In this way the output from one calculation can be used as the input for other calculations. This is shown in the example below for the calculation of Rankine cycle efficiencies. This ability to use references to calculations of previous state points in the determination of new properties is the key to the use of the Toolbox for building complex models.

The calculator in Figure 2 can also be used with cell references. Instead of entering the constant values shown in that figure, a user can place a cell reference for the property data. The user can type the cell reference manually or use the following automated procedure: first click on the data entry text box to place a blinking cursor there then click on the desired cell and the reference will automatically appear in the text box.*

Although the calculator shown in Figure 2 is the simplest approach for computing properties at individual state points, the cell formulas provide a method for doing spreadsheet analysis of thermodynamic systems of several components. A simple example of such a calculation is the determination of the efficiency of a simple Rankine cycle. The diagram of such a cycle and the state points in the cycle are shown in the spreadsheet in Figure 5.



	A	B	C	D
1	Name	Value	Unit	
2	cond_P	20000	Pa	
3	turb_T	400	deg C	
4	turb_P	10000000	Pa	
5	h_1	=TFProp("Water","SI_C","P",cond_P,"X",0,"H")	kJ/kg	
6	s_1	=TFProp("Water","SI_C","P",cond_P,"X",0,"S")	kJ/kg-K	
7	h_2	=TFProp("Water","SI_C","P",turb_P,"S",s_1,"H")	kJ/kg	
8	h_3	=TFProp("Water","SI_C","P",turb_P,"T",turb_T,"H")	kJ/kg	
9	s_3	=TFProp("Water","SI_C","P",turb_P,"T",turb_T,"S")	kJ/kg-K	
10	h_4	=TFProp("Water","SI_C","P",cond_P,"S",s_3,"H")	kJ/kg	
11	efficiency	=(h_3-h_4+h_1-h_2)/(h_3-h_2)	%	
12				

Figure 5. Rankine Cycle with Worksheet Formulas

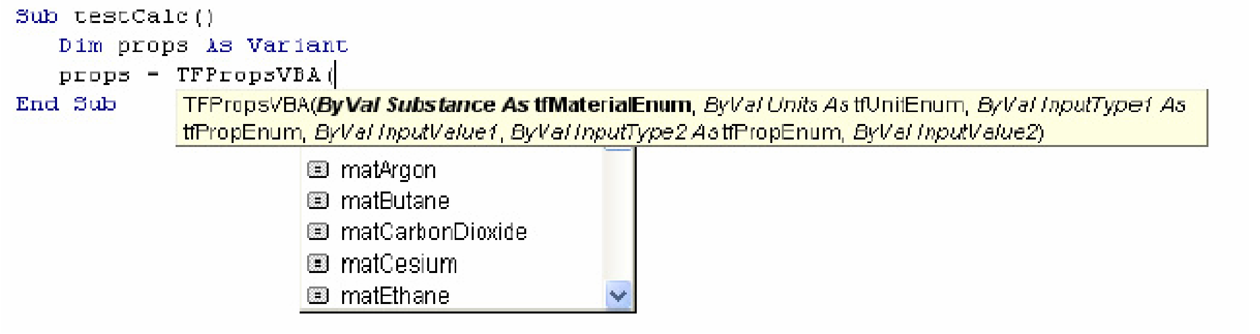
* When selecting a cell reference, users can click on the small button to the left of the data entry text box (with the minus sign at the bottom). This will shrink the calculator to reveal almost all of the spreadsheet. Users who have Excel versions prior to Office XP have obtained system crashes with the use of this icon.

The spreadsheet shows the formulas (rather than the numerical results) used for the cycle efficiency calculation. In these cell formulas, the left-hand column is used to define named cells that are used in the formulas. These named cells are used as cell references in the formulas in cells B5 to B12. Note the commonality of the cell formulas. Once a formula has been obtained from the insertion dialog, subsequent formulas can be created by copying, pasting and editing previous formulas. Figure 5 also shows that calculations for a saturated liquid state, such as h_1, are done by selecting quality as one of the input variables and choosing a value of zero for quality. A saturated vapor state can be defined in a similar manner. An isentropic process is computed by first determining the inlet entropy and then using the same entropy as one of the properties that determines the outlet state.

All property calculations, including those in Visual Basic discussed below may be performed for any combination of the seven basic input variables. For example, an input of entropy and enthalpy will give correct results for a state in a one-phase or two-phase region. The variables may be placed in any order in the pull down menus and the cell formulas. All input values and output results are checked to make sure that they are within the range for which the property equations are valid. When using the calculator from Figure 2, the maximum and minimum ranges are displayed in a window which appears when the mouse is held over the data entry window for a variable.

Thermodynamic properties in Visual Basic – The Visual Basic for Applications (VBA) programming language of Excel™ can also be used to compute thermodynamic properties using a function notation similar to the one introduced above. This allows students to program simple functions that can be plotted to show trends. Although the functions designed for the spreadsheet may also be used directly in Visual Basic code[*], special functions have been created specifically for use from VBA. These functions have the same name as spreadsheet cell functions, but they have the letters VBA appended to their names. (TFProp is replaced by TFPropVBA and TFProps is replaced by TFPropsVBA.) These VBA functions have the same arguments as the spreadsheet cell functions, but they have two main differences from the cell functions that are important for the VBA programming. The first is that the arguments can be selected from enumerated lists that appear automatically when typing the function call into a VBA module, as shown in Figure 7. In this figure, the boxed section with function arguments and the drop-down list with the substance names appear when the students types the function name. Once the student chooses a name from this list, a drop-down list for unit choices appears. This process is repeated for the choice of variables to be used as inputs to the program.
(No list appears for the actual values of the variables, but the argument names InputValue1 and InputValue2 tell the student what should be entered at a given point in the function.) The argument list and pull-down menus provide a useful guide to students learning these functions.

[*] One step is necessary before using the thermodynamic toolbox formulas in Visual Basic: the XLThermalFluids module must be selected as an available reference for VBA. This is done by selecting references from the tools menu and placing a check in the box next to XLThermalFluids.



```
Sub testCalc()
   Dim props As Variant
   props = TFPropsVBA(
End Sub
```

TFPropsVBA(**ByVal Substance As tfMaterialEnum**, *ByVal Units As* tfUnitEnum, *ByVal InputType1 As* tfPropEnum, *ByVal InputValue1*, *ByVal InputType2 As* tfPropEnum, *ByVal InputValue2*)

Figure 6. Display of Function Arguments and Substance List for VBA Function

A second difference is that the TFPropsVBA function returns its values as a one-dimensional array as opposed to the two-dimensional arrays used for the cell functions. This provides a more intuitive way to retrieve the individual state values from the array returned by TFPropsVBA. The VBA routines (like the worksheet routines) accept the string (or integer) values for substance, units, and variable identifier as used in the cell functions, in addition to the use of the enumerated constants mentioned previously, such as those shown in Figure 6. The VBA code for the simple Rankine cycle efficiency is shown in Figure 7.

```
Function rankine(Pcond As Single, PSG As Single, TSG As Single) As Single

    ‘Pcond is the condenser pressure in pascals
    ‘PSG is the steam generator output pressure in pascals
    ‘TSG is the steam generator output temperature in degrees Celsius
    Dim h1 As Double     'Condenser outlet enthalpy in kJ/kg
    Dim v1 As Double     'Condenser outlet volume in m^3/kg
    Dim h3 As Double     'Steam generator outlet enthalpy in kJ/kg
    Dim s3 As Double     'Steam generator outlet entropy in kJ/kg·K
    Dim h4 As Double     'Turbine outlet enthalpy in kJ/kg
    Dim pumpWork As Double    ‘Pump work in kJ/kg

    'Compute state points and efficiency
    h1 = TFPropVBA(matWater, unitSI_Celsius, propPressure, Pcond, _
                   propQuality, 0, retEnthalpy)
    v1 = TFPropVBA(matWater, unitSI_Celsius, propPressure, Pcond, _
                   propQuality, 0, retVolume)
    h3 = TFPropVBA(matWater, unitSI_Celsius, propPressure, PSG, _
                   propTemperature, TSG, retEnthalpy)
    s3 = TFPropVBA(matWater, unitSI_Celsius, propPressure, PSG, _
                   propTemperature, TSG, retEntropy)
    h4 = TFPropVBA(matWater, unitSI_Celsius, propPressure, Pcond, _
                   propEntropy, s3, retEnthalpy)

    pumpWork = v1 * (PSG - Pcond)  'Isentropic work as v * deltaP
    pumpWork = pumpWork / 1000#     'Convert Joules to kiloJoules
    rankine = (h3 - h4 - pumpWork) / (h3 - h1 - pumpWork)

End Function
```

Figure 7. VBA Code to Compute Efficiency of Simple Rankine Cycle

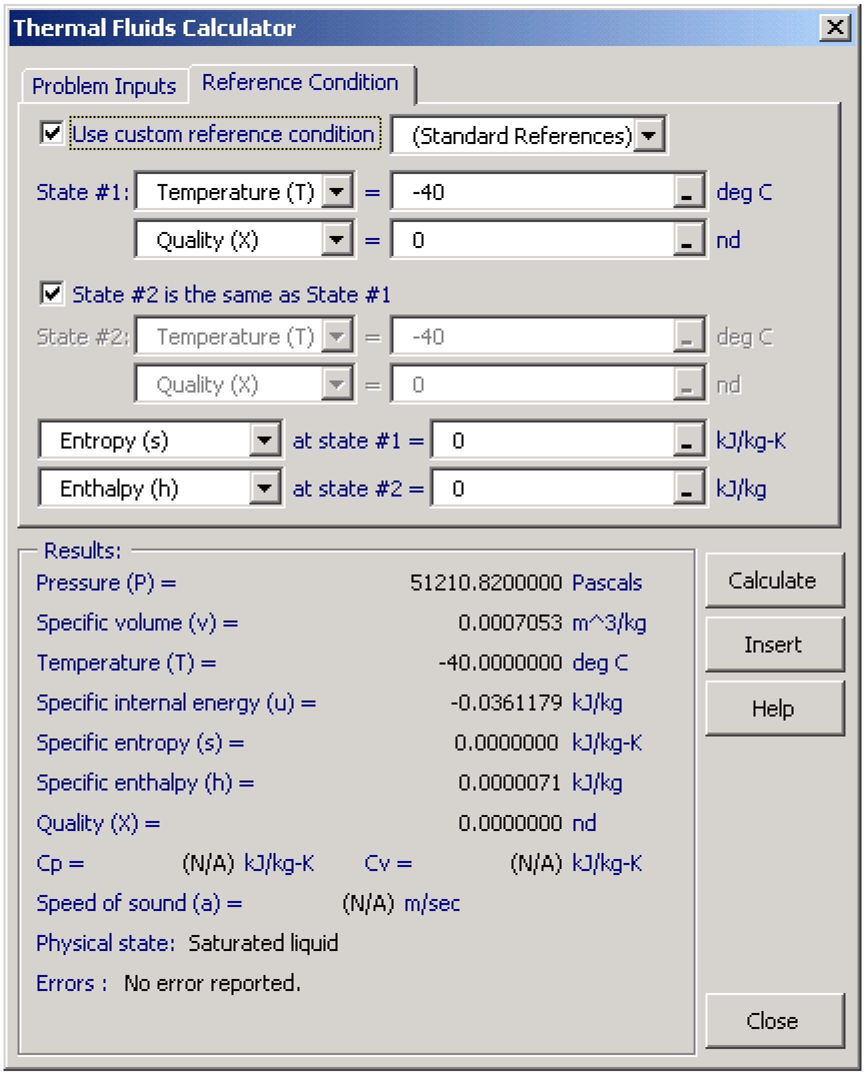
In this code, enumerated constants `matWater` and `unitSI_Celsius` specifies water as the substance water for property calculations and the unit system as SI units with temperatures in degrees Celsius. Other enumerated constants, `propPressure, propQuality,`and `retEnthalpy`, specify, respectively, the input properties of pressure and quality, and the enthalpy as the property returned by the formula. Other than the use of the enumerated constants, the function calls are seen to be similar to the ones for the cell formulas shown in Figure 5. The computation of the pump work is different in the two approaches to show the two possible calculations. The cell formulas use an isentropic calculation to determine the final outlet state of the pump. The VBA code uses the calculation of the isentropic work in a liquid as $v\Delta P$. Either approach gives about the same value for the efficiency. The use of the TFPropVBA routine to compute the outlet state may be less accurate for small pressure differences since an iteration procedure is used to determine the final density and temperature in a region where the fluid is nearly incompressible. Because of this, it is difficult to determine the exact density and temperature that give the specified pressure and entropy conditions.

The VBA code in Figure 7 provides a user defined function that can be used as a spreadsheet cell formula such as: =rankine(<cell reference for condenser pressure>, <cell reference for steam generator pressure>, <cell reference for steam generator outlet temperature>). With the appropriate choices for the input cell values, a student can examine the effects of the various cycle parameters on the cycle efficiency.

Reference states. Different publications may use different reference conditions for enthalpy and entropy. ASHRAE refrigerant tables use the saturated liquid at -40^{o}F (or -40^{o}C) as the zero point for entropy and enthalpy. The original data for many of the substances were taken from Reynolds[12] who uses the saturated liquid at 200 K as the zero point for entropy and enthalpy. Data for R-134a[13] and R-123[14] use the reference point for the International Institute of Refrigeration. This reference state is a saturated liquid at 0^{o}C where the enthalpy value is 200 kJ/kg and the entropy value is 1 kJ/kg·K. The program offers the ability to choose a different reference state as shown in Figure 8. This feature is important to maintain compatibility between the toolbox results and other tables available to the student, which may use a different zero state.

The dialogs shown in Figures 2 and 8 are actually part of the same dialog window. The dialog in Figure 2 is obtained by choosing the “Problem Inputs” tab; this is the default when the calculator is opened. Selecting the “Reference Condition” tab reveals the dialog shown in Figure 8. This dialog allows the user great flexibility in defining reference states. Different states can be defined for entropy and enthalpy and the reference state values of enthalpy and entropy can be set to any value selected by the user. Users can also select to define a reference condition on internal energy in place of enthalpy. Finally, the upper right hand corner allows the user to select commonly used reference conditions.

Figure 8 shows the results for a calculation with R-134a using a modified reference state. As noted earlier, the equations for R-134a use a reference state of the saturated liquid at 0^{o}C where the values of enthalpy and entropy are 200 kJ/kg and 1 kJ/kg·K, respectively. If this default reference state were used, the enthalpy and entropy of the saturated liquid at -40^{o}C would be 148.22 kJ/kg and 0.79594 kJ/kg·K, respectively. Figure 8 shows the data entries required if the user desires to have $h = s = 0$ for the saturated liquid at -40^{o}C. Regarding the computed outputs in Figure 8, the input state (not shown) is chosen as the reference state. The results of the calculation are seen to give zero values for enthalpy and entropy, which are expected for the reference state (within the round-off error of the calculations). It is also possible to select different reference states using cell formulas.



☑ Use custom reference condition (Standard References)

State #1: Temperature (T) = -40 deg C

Quality (X) = 0 nd

☑ State #2 is the same as State #1

State #2: Temperature (T) = -40 deg C

Quality (X) = 0 nd

Entropy (s) at state #1 = 0 kJ/kg-K

Enthalpy (h) at state #2 = 0 kJ/kg

Results:

```
Pressure (P) =                          51210.8200000  Pascals
Specific volume (v) =                       0.0007053  m^3/kg
Temperature (T) =                         -40.0000000  deg C
Specific internal energy (u) =             -0.0361179  kJ/kg
Specific entropy (s) =                      0.0000000  kJ/kg-K
Specific enthalpy (h) =                     0.0000071  kJ/kg
Quality (X) =                               0.0000000  nd
Cp =          (N/A)  kJ/kg-K       Cv =         (N/A)  kJ/kg-K
Speed of sound (a) =         (N/A)  m/sec
Physical state:  Saturated liquid
Errors :   No error reported.
```

Figure 8. Calculator with Custom Reference Condition

Equations used. The bulk of the equation-of-state data and the computational procedures were taken from Reynolds.[12] The basic approach is to write all equations of state in the following form.

$$P = \frac{RT}{v-b} + \sum_{i=1}^{N} C_i(T)H_i(v) \qquad [3]$$

In this equation P is the pressure, v is the specific volume, T is the absolute temperature, R is the engineering gas constant, and b is a coefficient that is zero in most equations. The first term in this equation, with b = 0, is simply the ideal gas law. The remaining terms mark a deviation from the ideal gas law. In the summation, $C_i(T)$ is a function of temperature only and $H_i(v)$ is a function of specific volume only. Although equation [3] gives the pressure as a function of temperature and specific volume, most equations of state use density instead of specific volume. Regardless of the equation of state, specific volume is the variable specified by and output to the user and is used in the equations here. The internal energy and entropy can be written as the sum of an ideal gas contribution and a departure term.

$$u = u^o + u^d \qquad\qquad s = s^o + s^d \qquad [4]$$

The departure terms can be found from the following equations.

$$u^d = \sum_i \left[C_i(T) - T\,\frac{dC_i(T)}{dT} \right] \int_\infty^v H(v')dv' \quad s^d = -\sum_i \frac{dC_i(T)}{dT} \int_\infty^v H_i(v')dv' \quad [5]$$

Equation [5] shows that dC_i/dT and $\int_\infty^v H(v')dv'$ are required for calculations of internal energy and entropy. In addition, the derivatives dC_i^2/dT^2 and dH/dv are required for computing derivatives used to solve the equations via Newton's method. These derivatives are also required for the computation of other properties such as heat capacities and sound speed. The property calculations are done in terms of the $C_i(T)$ and $H_i(v)$ functions and their derivatives or integral. All equations of state are programmed to calculate these terms, and the basic property calculation code is not changed by the addition of a new equation of state.

Programming considerations. The main calculation routines, programmed in Fortran, were initially developed for mainframe use. They have undergone various modifications for application to desktop computers in both a command line and Windows™ environment. The adaptation to Excel™ was accomplished to provide thermodynamic properties calculations within a computing tool that is widely available for engineering calculations. The previous Fortran code could be used as a set of subroutines for modeling thermodynamic systems, but the code structure required the user to write the system modeling code in Fortran.

The Thermal Fluids toolbox is built as a standard Excel™ add-in. This allows for direct calls to the toolbox functions to be made from the worksheet in addition to the graphic interface elements discussed above. The existing Fortran code was recompiled as a Dynamic Link Library (DLL) which is called by the various VBA routines to perform a calculation. Retaining the code in Fortran insulated the toolbox from any speed degradation due to multiple function calls in a particular spreadsheet. The XLManager utility surrounding the Thermal Fluids toolbox allows for quick loading and unloading of the toolbox so that the performance of the Excel™ application is not degraded when the toolbox routines are not needed. Finally, a hyperlinked help file in standard Windows™ format is included with the toolbox and is accessible from the main toolbox menu as well as from the property calculator.

The migration to Excel™ allows users to build models as a combination of cell formulas and Visual Basic code and to use built-in Excel™ tools such as GoalSeek or Solver for finding optimum solutions. The graphing features of Excel™ are also readily available to display trends from system calculations and to prepare property diagrams. The Thermal Fluids toolbox is specifically designed to allow for maximum flexibility in analyzing complex systems.

The underlying Fortran code has several checks to determine errors and provide error messages where users have selected inputs that lead to an invalid state. Iteration controls are set to detect potential divergence and return an "iterations not converged" message before potential program crashes. Results for each substance have been compared to tables in the references for the equations of state to ensure correct results. Users of the software who encounter errors are encouraged to report such errors to the authors.

Classroom applications. At CSUN, students are introduced to the thermodynamic toolbox early in the semester. A set of instructions and a sample spreadsheet are available for downloading. In some classes the students have one class meeting in a computer laboratory for a hands-on demonstration. Following this initial introduction, students are free to use the toolbox in their homework solutions, but they have to use tables for examinations. The use of tables is required because not all students have access to a laptop computer, which would be required to run the toolbox.[*] The requirement to learn tables for examinations limits the students' use of the toolbox during the semester.

The main application of the thermodynamic toolbox at CSUN has been for design projects. These projects are intended to provide students the experience of determining parameters in a process with several components. One problem that has been used in this regard is shown in Figure 9. Students are asked to construct a spreadsheet model of this system in Excel™ and use the Solver tool to determine the optimum values of the four intermediate turbine withdrawal pressures: P_{12}, P_{13}, P_{14}, and P_{15}. Students find problems like this quite intimidating when the problem is assigned near the end of their first course in thermodynamics. However, with help in formulating the problem, students generally find that the toolbox in Excel™ gives them a useful tool for solving complex thermodynamics problems.

Another application that can be used in advanced courses is to link the toolbox with a simple routine for the numerical integration of ordinary differential equations to solve transient first law problems. Typical textbook approaches to these problems assume that the states of the inlet and exit streams remain constant during the process. Using the toolbox it is straightforward to integrate the first order equations for the first law and mass balance in a transient open system in a manner that accounts for the property variations of the flowing streams.

The toolbox is a useful tool for instructors to obtain solutions to problems that they create for lectures or examinations. It can also provide accurate property graphs for lecture presentations. One somewhat ironic use of the toolbox is the development of simple one-page property tables that can be used for closed-book quizzes and examinations.

[*] The authors believe that it should be possible to run the toolbox on any handheld computer or personal digital assistant (PDA) running Windows PocketPC and plan to create a stand-alone application of the toolbox (separate from Excel) this year.

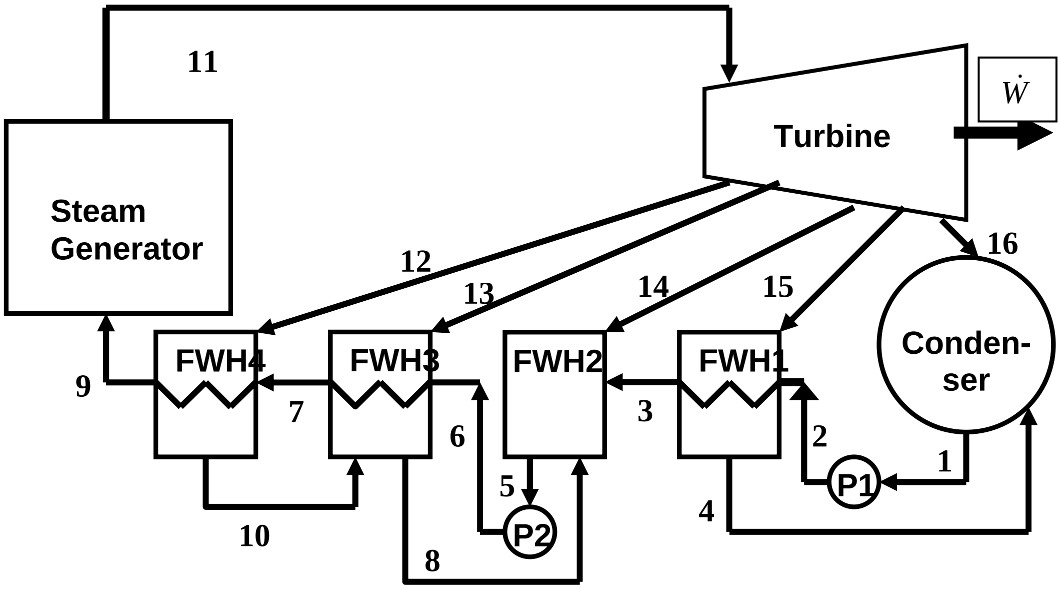


Figure 9. Steam cycle Diagram from Çengel and Boles[2] Used for Design Project

In a separate article accompanying a poster session at this conference, Maixner[15] discusses the use of the Thermal Fluids toolbox in various thermal sciences courses at the U.S. Air Force Academy. According to Maixner, cadets "greeted [the toolbox] with a great deal of enthusiasm" and asked why they did not use it when first introduced to property tables. Maxiner found that the use of the toolbox in more advanced courses allows them to focus more on cycle analysis and optimization without being burdened with tedious table lookups.

Conclusions

The Thermal Fluids Toolbox provides calculations of thermodynamic properties in an Excel™ spreadsheet. Calculations may be done for a variety of substances using a flexible input format. A graphical interface calculator is available for computing individual state points and complex models may be constructed from cell formulas, Visual Basic code, or a combination of the two. The software may be downloaded at no cost from www.spreadsheetworld.com.

Disclaimer

The views expressed are those of the authors and do not reflect the official policy or position of the US Air Force, Department of Defense, or the US Government.

Bibliography

[1] NIST Reference Fluid Thermodynamic and Transport Properties Database (REFPROP): Version 7.0, NIST Standard Reference Database 23, web page http://www.nist.gov/srd/nist23.htm viewed on December 20, 2004
[2] Yunus A. Çengel and Michael A. Boles, *Thermodynamics, an Engineering Approach* (4th edition), McGraw-Hill, 2002.

[3] Richard E. Sonntag, Claus Borgnakke, and Gordon J. Van Wylen, *Fundamentals of Thermodynamics* (6th edition), Wiley, 2003.

[4] Michael J. Moran and Howard N. Shapiro, *Fundamentals of Engineering Thermodynamics*, (4th edition), Wiley, 2000.

[5] Gregg W. Dixon, "Teaching Thermodynamics without Tables – Isn't it Time?**"** Proceedings of the 2001 American Society for Engineering Education Annual Conference & Exposition, 2001.

[6] C. C. Ngo and F. C. Lai, "Web-Based Thermodynamics Tables Wizard," Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition, 2002.

[7] Kyaw Aung, "Integration of Computational Tools in Engineering Thermodynamics Curriculum," Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition, 2002.

[8] Susan T. Hudson , "Laptop Computer Integration in a Lower Level Mechanical Engineering Course," Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition, 2002.

[9] Craig W. Somerton, Chiew-Ping Bong, Laura J. Genik, "A MATLAB Toolbox for Thermodynamic Property Evaluation", Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition, 2003.

[10] David G. Goodwin, "TPX: Thermodynamic Properties for Excel," http://www.its.caltech.edu/~emg/me18/tpx.pdf, downloaded December 20, 2004.

[11] Ashok K. Khanna, *Development of Thermodynamic Computerized Properties Calculations*, MS Project Report in Engineering, California State University, Northridge, 1982.

[12] W. C. Reynolds, *Thermodynamic Properties in SI*, Department of Mechanical Engineering, Stanford University, 1979.

[13] Reiner Tillner-Roth and Hans Dieter Baehr, "An International Standard Formulation for the Thermodynamic Properties of 1,1,1,2-Tetrafluoroethane (HFC-134a) for Temperatures from 170 K to 455 K and Pressures up to 70 MPa" *J. Phys. Chem. Ref. Data*, **23**, 657-729 (1994).

[14] Ben A. Younglove, and Mark O. McLinden, "An International Standard Equation of State for the Thermodynamic Properties of Refrigerant 123 (2,2-Dicholoro-1,1,1-Trifluoroethane), *J. Phys. Chem. Ref. Data*, **23**, 731-779 (1994).

[15] Michael R. Maixner, "Excel™ Analysis of Combined Cycle Power Plant," Session 2005-1022, Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition, 2005.

Biographical Information

LARRY CARETTO
Larry Caretto is a Professor of Mechanical Engineering at California State University, Northridge (CSUN). He received his BS, MS, and PhD in engineering from UCLA. His interests are in combustion, air pollution, energy, computer simulation and public policy.

DAVID MCDANIEL
David McDaniel is an Assistant Professor of Aeronautics at the U.S. Air Force Academy. He received his BS in Aeronautical Engineering from the Academy, his MS in Mechanical Engineering from The George Washington University, and is currently working on his PhD at the University of Colorado at Colorado Springs in computational aerodynamics. His interests include flight dynamics, computational fluid dynamics, and structural dynamics.

TOM MINCER
Tom Mincer is an Associate Professor of Mechanical Engineering at CSUN. He received his MS and PhD in Aerospace Engineering from USC. Tom is president of SpreadsheetWorld, Inc., a company that provides training services in integrated systems design teams using basic computer tools to link system models for simulation, configuration trade studies, optimization, and uncertainty analysis and Excel™/VBA engineering toolboxes.