A VISUAL RANKINE CYCLE SIMULATION USING LABVIEW

Patrick A. Tebbe
Department of Engineering
The College of New Jersey
Ewing, NJ 08628
tebbe@tcnj.edu

Stephen Lombardo
Department of Chemical Engineering
William Miller
Nuclear Sciences and Engineering Institute
Christa Weisbrook
College of Engineering
University of Missouri – Columbia
Columbia, MO 65211

ABSTRACT

This paper describes a non-traditional use of the LabVIEW software package to create course software for thermodynamics. The Rankine steam cycle has been modeled and an appropriate graphical user interface has been produced to achieve a visual solution environment. Thermodynamic properties are computed based on internal data arrays using interpolation techniques. Based on user input values the remaining systems parameters such as efficiency and heat rates can be calculated. A temperature versus entropy diagram is also produced for each cycle solution. To facilitate trend analysis and exploration of system parameters data from successive solutions can be written to a user file for later analysis. Use of this software at both institutions has found it to be very easy to use and understand. The program was found to be an excellent way to add interest to this subject.

INTRODUCTION

A difficulty in engineering thermodynamics courses is providing students the time and freedom to explore more advanced topics and design problems. This is particularly true when thermodynamic cycles are involved due to the coupling between variables. Students can only work a limited number of problems, even with the aid of modern calculators. Solutions become prohibitive for large problems, such as realistic plant cycles, because they are computationally intensive. These problems can be partially addressed with the use of computer modeling. The use of simulation for training, particular for thermodynamic power cycles, is not new. For years the commercial power industry has made use of simulators for training and plant scenario analysis. The nuclear power industry in particular is very active in the application of simulators. However, there are arguments against the use of simulators in the classroom, namely the time they may take away from other topics and the fear that students will not learn to understand the underlying principles for themselves. The authors will agree that these are legitimate concerns, however; used in conjunction with traditional teaching methods and with the right objectives in mind simulation can provide several advantages. Recent work by Benson[1] presented several approaches to the use of computers in engineering education along with their benefits. In the authors' view the most important benefit is that the students can take greater control over the learning process.

The goal of this work has been to create educational material for engineering that will aid in instruction of thermodynamics. Specifically we have explored a non-traditional use of the LabVIEW package to develop several program modules. These programs are intended to supplement traditional lectures and homework assignments by allowing individual exploration and facilitating numerical calculation of system variables for design problems. The original implementation simulated a steam turbine and included a conditional structure that allowed any combination of known turbine values (cycle input, output, or both) to be used as user input.[2] The current program is more constrained with regard to user input but is structured for more appropriate classroom use. It is based on the simple four-component Rankine cycle. Specific design criteria were that the program be extremely user friendly, able to be used for trend or design analysis, and be visually stimulating.

PROGRAM STRUCTURE

The programs have been designed around the graphical programming language G and the software environment of LabVIEW, produced by National Instruments. LabVIEW was originally developed for data acquisition purposes but has since expanded into a wide range of fields. Since LabVIEW has so many applications in engineering it serves as a cost effective software package for academic institutions that may already use it for other courses and programs. The visual programming method of LabVIEW is simple yet powerful and deserves a brief explanation. LabVIEW programs consist of two windows, the wiring diagram that represents the programming structure and the front panel that takes the place of the graphical user interface. The front panel can contain a variety of different input and output objects. Creation of the front panel is done using a point and click method that places controls and indictors on the front panel. The second window, or wiring diagram, provides all the programming functionality. Small graphical icons represent connections to existing front panel controls and indicators as well as basic programming functions. These icons are connected with "wires" which serve as a path for dataflow. Programs in LabVIEW are referred to as "Virtual Instruments" or VIs. Every VI can be represented by its own icon with defined connection points for input and output. In this way, a VI can become a sub-VI much like a subroutine in FORTRAN or C++. The visual programming method of LabVIEW allows the user to avoid the overhead associated with text-based languages since most "programming is done with the mouse rather than the keyboard" .[3]

Correct calculation of the fluid properties at a given state is vital for any modeling effort in thermodynamics. For these programs the properties of enthalpy, entropy, and specific volume as functions of pressure and temperature are needed. However, the thermodynamic state can be specified by any two independent properties and the capability must exist to calculate properties in a variety of ways. The first option pursued was simple analytical equations for the properties given by Ganathapy.[4] These proved easily implemented but exhibited a greater degree of error than desired, especially in regions of high and low pressure. A second option was then implemented by inputting the property tables in their entirety into LabVIEW arrays. Property calculation was then simply a matter of table lookup or interpolation. The speed and error associated with this method was found to be quite acceptable. This method did produce some problems in areas near the vapor dome, however; and care had to be taken in the number and location of table points used.

The most important thermodynamic property VI was the one used to calculate the saturation properties of water, "Saturation_Table." This VI was used throughout the program as a sub-VI. The saturation properties themselves are provided in an eight component array holding saturation pressure and temperature, as well as saturated liquid and saturated vapor values for enthalpy, entropy, and specific volume. Since a point in the saturation table can be specified by either a pressure or a temperature, a "Case Structure" is used with two options (T or P). Within each the LabVIEW VI "Threshold 1D Array" is used to bound the desired point within the property array. The result of this calculation is then input to the LabVIEW provided "Interpolate 1D Array" VI. This interpolation is performed inside a "For Loop" which repeats in order to interpolate all values from the table. The results are output as a property array.

A VI called "Superheated_Tables_ Interpolation" was also created to output the superheated properties given both pressure and temperature as inputs. Just as for the saturation VI, the thermodynamic property tables are input as an array. The threshold VI is then used to bound first pressure then temperature. These values can then be used to perform a double interpolation to get the desired thermodynamic properties. If the requested point is beyond the range of the property array a linear extrapolation is used to determine the properties. An informational message is output to the user explaining this.

Often, however, both superheated pressure and temperature are not known. The thermodynamic state must be fixed by a different combination of values, for instance pressure and enthalpy. A separate "Super_props" VI was created for these cases. Either pressure or temperature and either enthalpy or entropy are used as the inputs. Consider the case where pressure and enthalpy are input. A "For Loop" is used to scan the available temperatures in the property array by comparing them to the saturation temperature at the desired pressure. This determines which temperatures are valid for the superheated property region (i.e., they must be larger than the saturation temperature at this pressure). Then each of these temperatures is used to call the "Superheated_Tables_Interpolation" VI to obtain the enthalpy. From these values the bounding temperatures, based on comparison to the input enthalpy, are identified. This procedure is repeated using temperature increments of 1, 0.1, 0.01, and 0.001 degrees Celsius. The resulting temperature is then "assembled" down to three decimal places.

RANKINE CYCLE SOLUTION

The Rankine cycle is the universal steam power cycle taught in thermodynamics courses. The basic system consists of a boiler, steam turbine, condenser, and pump in which the working fluid, in this case water, continually recirculates (Figures 1 and 2). Students are generally given a set of component data, such as pressures or temperatures and asked to solve for work, heat flux, or cycle efficiency. Alternatively, cycle parameters can be given and a specific component value can be solved for. Following the same basic procedures a student would use, solution of the cycle problem proceeds in a sequential manner, using a "Sequence Structure" with six frames. The equations behind these steps can be found in any standard engineering thermodynamics textbook and will not be repeated here. The basic steps in each frame are:

Frame 0:
There are several aspects of physical significance in the Rankine cycle. If the water flowing through the steam turbine begins to condense too much (i.e. part of the water converts to liquid form) the turbine blades can be damaged by impact. Therefore, the Rankine program first checks the turbine inlet condition (1) to determine if the thermodynamic state is saturated or superheated. If the state is saturated a warning message is output to the user explaining the problem.

Frame 1:
The exit state of the turbine (2) is determined. Assuming isentropic behavior the exiting entropy can be calculated using the inlet pressure and temperature. The saturated vapor entropy is then calculated and used to determine if the quality is less than, equal to, or greater than 1. Depending

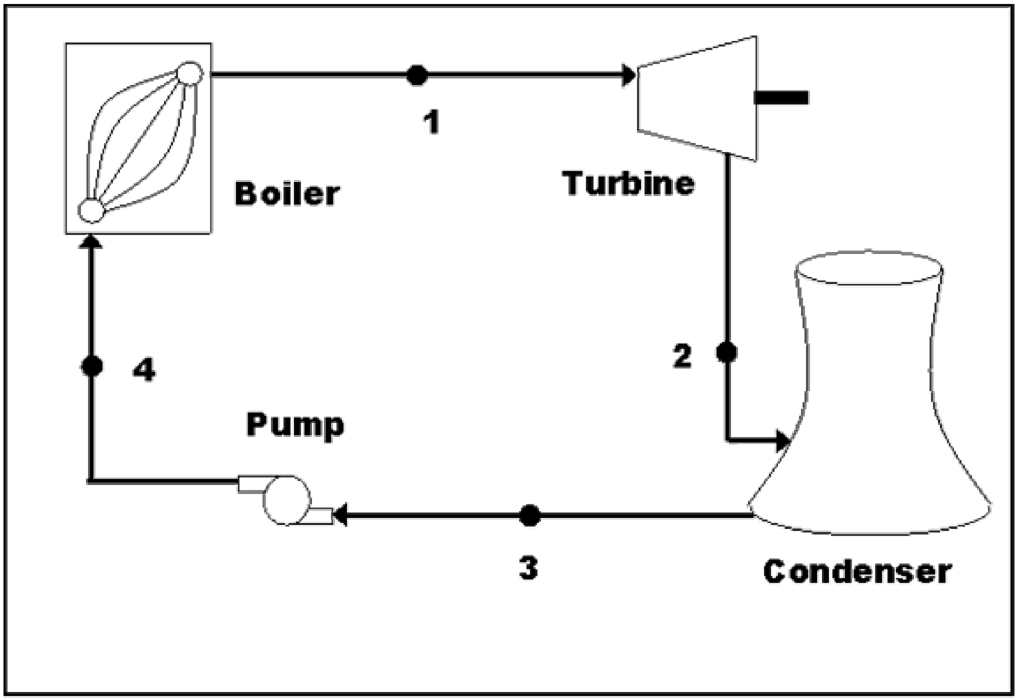


Figure 1: Schematic of the basic Rankine steam power cycle.

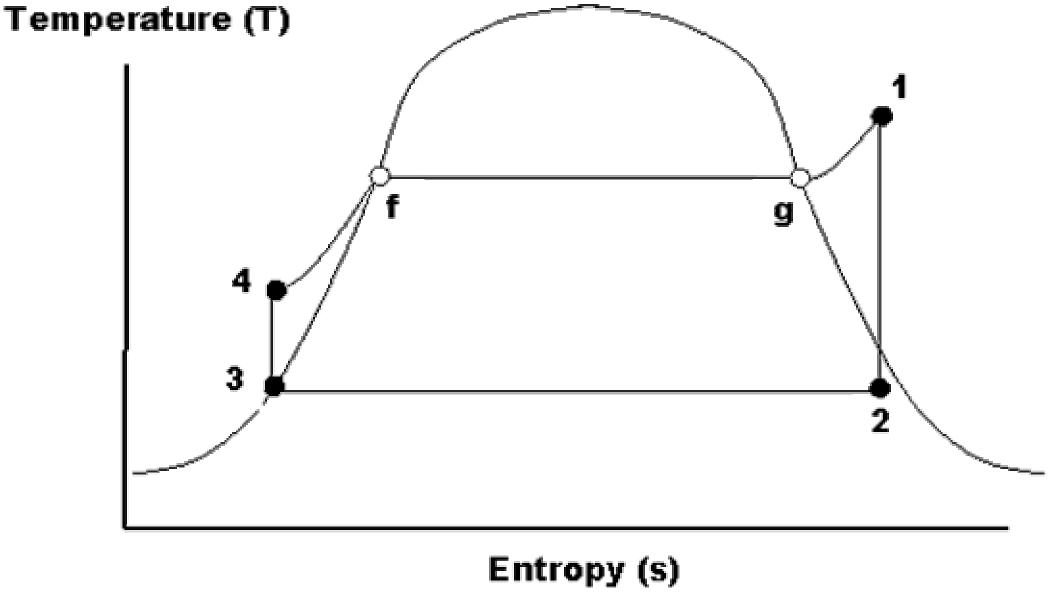


Figure 2: Typical T-s diagram for a Rankine cycle. The numbered points refer to Figure 1. The (f) and (g) points refer to the saturated liquid and vapor states, respectively, at that pressure.

on the result, the saturated or superheated property VI is then called to determine the value of enthalpy. To correct for the isentropic assumption allowance is provided for the input of the turbine isentropic efficiency and the corrected enthalpy is then calculated. Using this enthalpy and the condenser pressure as inputs, the "Super_props" VI is then called to calculate all other properties at this point.

Frame 2:
With the assumption of saturated liquid at the exit of the condenser (3), the exit properties are computed from the "Saturation_Table" VI using the provided condenser pressure.

Frame 3:
With an isentropic pump assumption, the inlet (3) and exit (4) pressures are used to calculate the isentropic work. This value is corrected using an input pump efficiency. The resulting value can then be used to calculate the actual exit enthalpy based on conservation of energy. By approximating the enthalpy of liquid water using the specific heat an approximate value of temperature can also be calculated. All other properties at the exit state can then be approximated as the saturated liquid properties from the "Saturated_Table" VI.

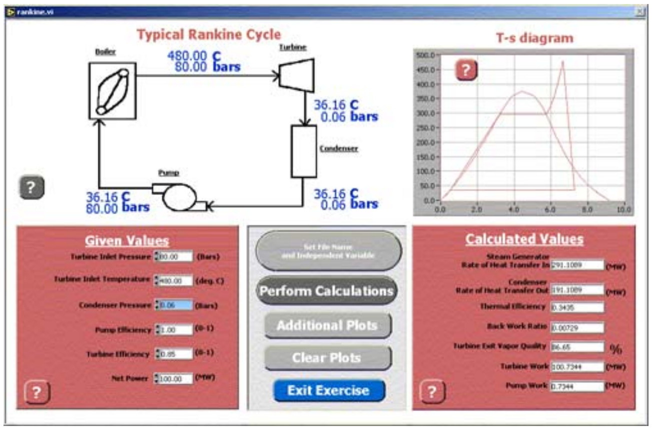
Frame 4:
All interface indicators are updated with the newly calculated values. System parameters, such as overall efficiency, are then calculated.

Frame 5:
The arrays necessary to produce the T-s diagram are computed and plotted.

USER INTERFACE

The user interface (Figure 3) was designed using LabVIEW's graphical interface features. The lower left box allows students to input values for six parameters; turbine inlet pressure and temperature, condenser pressure, pump and turbine efficiency, and net cycle work. Only the first three values are always necessary. The component efficiencies will default to values of one, signifying isentropic operation. Specifying net power is equivalent to specifying the mass flow rate. If this value is set to a default of 1, the resulting calculations represent specific values, or per mass units. Since the program was designed with trend analysis in mind the student must first specify an output file and independent variable using the top middle control button. Once the appropriate input values are set, the "Perform Calculation" button will cause the cycle to be solved. All values on the top left cycle diagram will then be updated. In the top right box an appropriate T-s diagram will be produced and in the lower right box the system values will be displayed. Each time the "Perform Calculation" button is pressed system values are written to the output file as a function of the specified independent variable. Pressing the "Additional Plots" button opens a new window (Figure 4) with plots of heat rate into the boiler, cycle efficiency, back work ratio, and steam quality out of the turbine versus this variable. Should the user make a mistake and wish to start over, the "Clear Plots" command will erase all current data.

As already mentioned, the user interface was constructed with a temperature versus entropy (T-s) display for the entire cycle (upper right of Figure 3). This allowed students to see changes in the cycle in a more qualitative and visual fashion. In order to create this diagram an array of data is assembled containing points for all lines and input to a "XY Graph." In this type of graph, straight lines are drawn between successive coordinate pairs. On the actual T-s diagram (Figure 2) the lines from point 4 to the left side of the vapor dome (f) and from the right side of the dome (g) to point 1 are actually portions of the constant pressure line. These lines are not technically linear, however, on the scale of the diagram they can easily be approximated by straight lines. The necessary endpoints can easily be calculated using the input and computed properties at each point as well as the saturation data at the boiler pressure. These points are assembled in the proper order in an array. The dome itself is composed of a series of linear segments. The coordinates for this array are hard coded in the program as an array constant. Both arrays are bundled together and input to the "XY Graph".



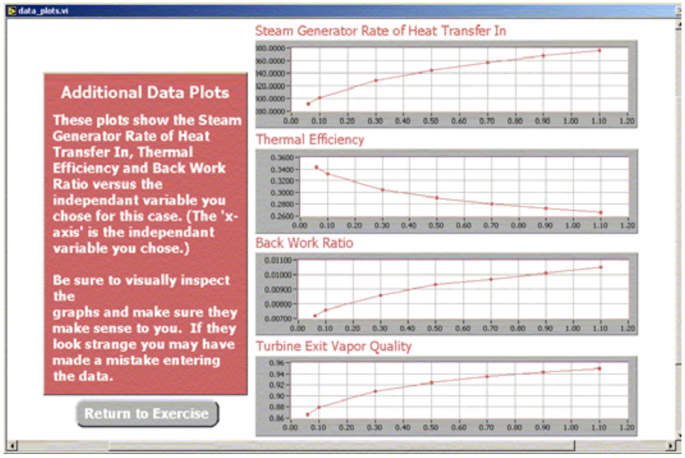
Given Values

Turbine Inlet Pressure 80.00 (Bars)
Turbine Inlet Temperature 480.00 (deg. C)
Condenser Pressure 0.06 (Bars)
Pump Efficiency 1.00 (0-1)
Turbine Efficiency 0.85 (0-1)
Net Power 100.00 (MW)

Calculated Values

Steam Generator Rate of Heat Transfer In 291.1089 (MW)
Condenser Rate of Heat Transfer Out 191.1089 (MW)
Thermal Efficiency 0.3435
Back Work Ratio 0.00729
Turbine Exit Vapor Quality 86.65 %
Turbine Work 100.7344 (MW)
Pump Work 0.7344 (MW)

Figure 3: User interface with typical problem values being displayed.



Additional Data Plots

These plots show the Steam Generator Rate of Heat Transfer In, Thermal Efficiency and Back Work Ratio versus the independant variable you chose for this case. (The 'x-axis' is the independant variable you chose.)

Be sure to visually inspect the graphs and make sure they make sense to you. If they look strange you may have made a mistake entering the data.

Figure 4: Additional plots available from the program.

OUTCOMES AND CONCLUSIONS

The described software has been used in thermodynamic courses at both the University of Missouri (MU) and The College of New Jersey (TCNJ). At MU the material was employed in both the basic engineering thermodynamics course (ENGR 322), taken by the majority of majors, and the chemical engineering thermodynamics course (ChE 261). In the ChE 261 course, the software was mainly used as a homework solution tool. After first solving similar problems by hand, the software was used to solve several additional homework problems. In the ENGR 322 course the software was used as part of a design project where the students observed the effect of changing several variables. As part of the assignment, they then were required to select a variable of their choosing for trend analysis. Using the data files from the LabVIEW program, plots were produced which the students then had to explain and discuss, in relation to system performance.

Based on student surveys and input from the student software assistant the material was extremely easy to use. Students were able to use and understand the graphical interface with minimal instruction and without prior demonstration. In fact, the teaching assistant commented that she spent more time assisting with the related use of Excel than the LabVIEW software. Student feedback also indicated that student interest in the topic increased while using the software. The addition of the T-s diagram, in particular, was well received. However, evaluation data indicated that the educational benefit was largely determined by how the software use was integrated with the class and assignments. Further details on this aspect will be presented at a later date.

While simulation teaching aides and thermodynamics are not subjects traditionally associated with the LabVIEW package, we have shown that there is great potential for educational uses of LabVIEW beyond data acquisition. This project has demonstrated a simple yet effective method of thermodynamic property calculation, as well as solution of a Rankine cycle problem. In addition, use of graphical interface items addresses the visual needs of many students. As previously mentioned, this software is not

intended to replace lecture or student solution of problems. However, the authors found it an excellent way to add interest and a change of pace to thermodynamics courses and facilitated more in-depth analysis of the subject.

REFERENCES

1. **Benson, T., 1998, "Using Computers in Fluids Engineering Education," Proceedings of FEDSM'98, paper no. FEDSM98-5135, American Society of Mechanical Engineers.**

2. **Tebbe, P. and Rodman, T., "Development of a Graphical Turbine Simulator," Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference, American Society of Mechanical Engineers (1999)**

3. **Burnett, M., Goldberg, A., and Lewis, T., Visual Object-Oriented Programming, Manning Publications (1995).**

4. **Ganapathy, V., 1986, BASIC Programs for Steam Plant Engineers, Marcel Dekker, Inc., New York City.**

ACKNOWLEDGEMENTS

Credit is due to the undergraduate and graduate students working on this project at the University of Missouri and at The College of New Jersey. The assistance of the entire project team at the University of Missouri is also appreciated, including the work of Dr. James Laffey, Scott Halverson and Gayla Neumeyer. This research is supported by the National Science Foundation through grant DUE-9950869.

BIOGRAPHICAL INFORMATION

Patrick Tebbe is an Assistant Professor of Mechanical Engineering at The College of New Jersey. Patrick teaches undergraduate thermodynamics and thermal sciences laboratories. He received his Ph.D. in Mechanical Engineering from the University of Missouri-Columbia in 1997. Before moving to New Jersey he served as an Adjunct Assistant Professor at Missouri and also pursued a M.S. in Nuclear Engineering.

Stephen Lombardo is an Associate Professor of Chemical Engineering at the University of Missouri – Columbia. He teaches undergraduate thermodynamics, unit operations laboratories, ceramic materials, and process controls. He received his Ph.D. from the University of California – Berkeley.

William Miller is a Professor in the Nuclear Science and Engineering Institute at the University of Missouri – Columbia as well as the Director for the Energy Systems and Resources Center. He received his M.S. from Kansas State University and his Ph.D. from the University of Missouri.

Christa Weisbrook is an Assistant Dean for Academic Programs for the College of Engineering at the University of Missouri – Columbia. She received her M.S. from the University of Wisconsin – Madison and her Ph.D. in Mechanical Engineering from the University of Missouri.