

Development of Engineering Analysis Software For Education and Research

S.A. Klein

Solar Energy Laboratory

Department of Mechanical Engineering

University of Wisconsin - Madison

Three energy analysis programs are described. The programs are: TRNSYS, a general sequential modular transient system simulation program originally developed for simulation of solar energy systems; CP/THERMO, a tutorial program designed to assist student learning of classical thermodynamics; and EES, a general non-linear simultaneous equation solving program incorporating thermodynamic property data. TRNSYS is a research tool, although there have been some attempts to use it for education. CP/THERMO and EES were originally developed for use in engineering education, although EES is now seeing increasing applications in research. This paper briefly describes each program indicating the motivation for its development, underlying principles, interface design, strengths, weaknesses and user-response. Although computer programs developed for education and research have quite different requirements, they also have a number of similarities from a developmental standpoint. A generalization of the experiences encountered in developing and using these programs is provided.

TRNSYS – Transient System Simulation Program

It is apparent that computer simulations can be a powerful tool for system design. Experience has shown, however, that the development of a versatile simulation program is a complex task requiring significant programming, validation, and documentation efforts. The incentive for development of the TRNSYS program^{1,2} was to simplify the task of formulating computer simulations of energy systems.

An obvious characteristic of many energy systems is their modularity. A refrigeration system, for example, consists of one or more compressors, heat exchangers, valves, controllers, and other equipment. Because of the modularity, it is tempting to design a simulation that calculates the performance of a system by collectively simulating the performance of each of its components. There are several advantages to this approach. First it reduces the complexity of the simulation problem by reducing a large problem into a number of smaller more manageable efforts. Second, many components are common to different systems so if they are formulated in a general manner, the component models can be reused in other studies. This latter capability provides an incentive for generalizing and documenting the component models.

Once all of the components of a system have been identified and a mathematical model has been formulated for each, it is next helpful to construct an information flow diagram for the system. An information flow diagram is a schematic representation of the flow of information into and out of each of the system components. Several types of information are evident in the information flow diagram. An obvious distinction is between information flowing into and out of a component. The input information, represented by arrows directed into the box in Figure 1, can be of three types. *Parameters* are used for information that does not vary with time, such as the physical size or surface area of a system. *Inputs* and *integrated quantities* are used to provide

information that does vary with time, such as temperatures and flowrates. The *integrated quantities* are calculated by the central integration solver built-into TRNSYS based upon *time derivatives* of these quantities output by the component model. In an transient simulation, time itself is a necessary input. *Outputs* represent information calculated by the model. The outputs of one model often become *inputs* to another, as seen in the information flow diagram for a simple solar energy system shown in Figure 2. The TRNSYS program allows the component models to be interconnected, analogous to the manner in which pipes and wires connect the actual equipment. TRNSYS must then solve the systems of algebraic and differential equations that result from component interconnection and provide other services such as printing, plotting, and reports.

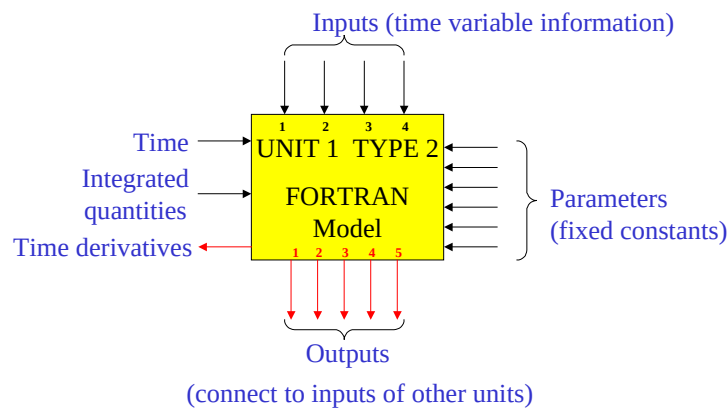


Figure 1: Information flow diagram for a component model

A system information flow diagram may exhibit recyclic flow of information. Recyclic flow occurs whenever there is a path in the information flow diagram formed by the output arrows that leads from a component to one or more other system components and then back to the starting component. The existence of recyclic flow indicates that iterative calculations are needed to solve the system of equations that represent the system model. TRNSYS is designed to recognize this situation and apply iterative techniques, as needed. In the original design of the TRNSYS, successive substitution was used in which each component model is called, if needed, until all of the output variables in the model converge to within a user-specified tolerance. Thus TRNSYS is classified as a sequential-modular solver. Later versions of TRNSYS provide a simultaneous solver option for the system inputs and outputs, but the advantage of a simultaneous solver is often confounded by logic statements used in the modules for controls and other purposes.

TRNSYS was designed to simulate transient systems which are represented by differential equations with time as the independent variable. The original design of TRNSYS provided a single integration algorithm to solve all differential equations in the system simultaneously. A second-order predictor-corrector algorithm was chosen for this purpose. The advantage of this algorithm for solving the simultaneous algebraic and differential equations that normally arise in a system model is that the iterative calculations occurring during a single time step are performed at a constant value of time. (This is not the case for Runge-Kutta algorithms). As a result, the solutions to the algebraic equation of the system converge by successive substitution as the

iteration required to simulate the differential equations progresses. Later versions of TRNSYS provide alternative integration methods.

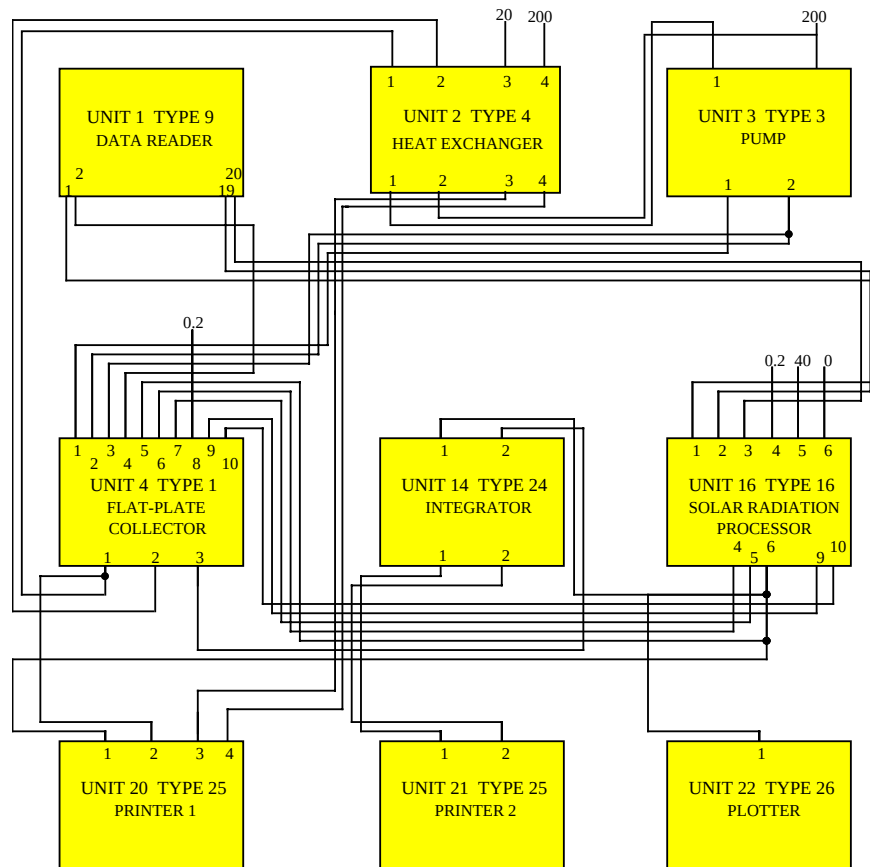


Figure 2: Information flow diagram for a simple solar energy system

TRNSYS has been extensively used for simulation of solar energy systems for which it was initially developed. A large library of validated component models, written in a general manner, is provided in the TRNSYS library. It is remarkable that TRNSYS, first written over 25 years ago, is still in use today. The program has been internationalized with major contributors from Germany, France, Belgium, and Sweden. The program has survived because its general nature allows it to simulate any modular transient system and because it has been continuously supported. The standard documentation required of all component models has also contributed to the overall usefulness of the program. The general principles around which TRNSYS is designed are of use in any general simulation program.

For all its success, TRNSYS has a number of disadvantages. The program is limited to component models written in FORTRAN. The development of the models in FORTRAN and subsequent compiling and linking can be tedious when repeated changes are required to the simulation model (the usual case). Graphical input and output capabilities have been developed for TRNSYS, but they are not fully integrated. The solution methods used by TRNSYS fail for some non-linear problems, particularly when there are no differential equations, as in steady-state simulations. Indeed, it may not be possible for a sequential modular solver as used in

TRNSYS to be computationally robust since specific details of the models and controls are not known to the solver, but rather coded within the component models. Finally, the program was found to be unsuitable for use in undergraduate education due to the complexity required to develop and modify component models.

CP/Thermo – Computerized Problems in Thermodynamics

The limitations of TRNSYS as an educational tool provided an incentive for the development of the CP/Thermo program³. This program was developed to assist students in their first or second mechanical engineering thermodynamics course.

Working problems contributes to student learning of thermodynamics. It is for this reason, most textbooks include a selection of problems after each chapter from which instructors commonly assign homework. CP/Thermo is a tutorial program that includes over 100 problems covering most of the subjects covered in traditional mechanical engineering thermodynamics textbooks. The program offers an alternative to conventional homework.

A brief description of CP/Thermo is necessary to understand its capabilities. When started, the program displays list of subjects from which the student can select a topic, unit system, and associated textbook. A second menu provides a list of problems for the chosen topic. The program randomly chooses among the available data sets for the chosen problem and displays the problem statement in a window. Separate windows are provided for a figure, nomenclature, status (which summarizes known information), equation summary (which lists all equations that have been identified), and a score/time window, as shown in Figure 3. An additional window, not shown in Figure 3, is provided for property input.

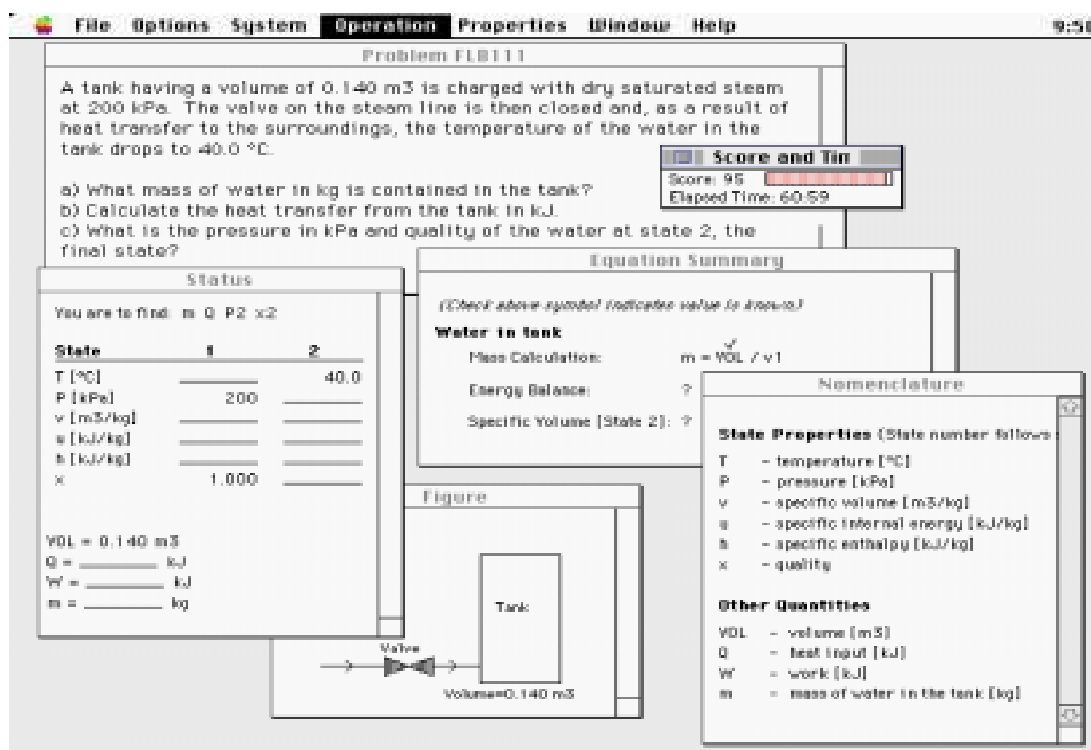


Figure 3: Windows for the CP/Thermo program

The student can proceed to select a system and apply mass, energy, or entropy balances in any order. The status window shows all known quantities and provides blanks for unknown quantities. The student can enter values for unknown quantities directly in this window, if the values of the quantities can somehow be determined. In a more organized approach, the student would choose an operation, such as an energy balance. In some cases, CP/Thermo provides the equation whereas in others, the student is required to enter the equation. In this case, a keypad with symbols that likely appear in the equation is provided to facilitate equation input, as shown in Figure 4. Any algebraic equivalent of the correct equation is accepted.

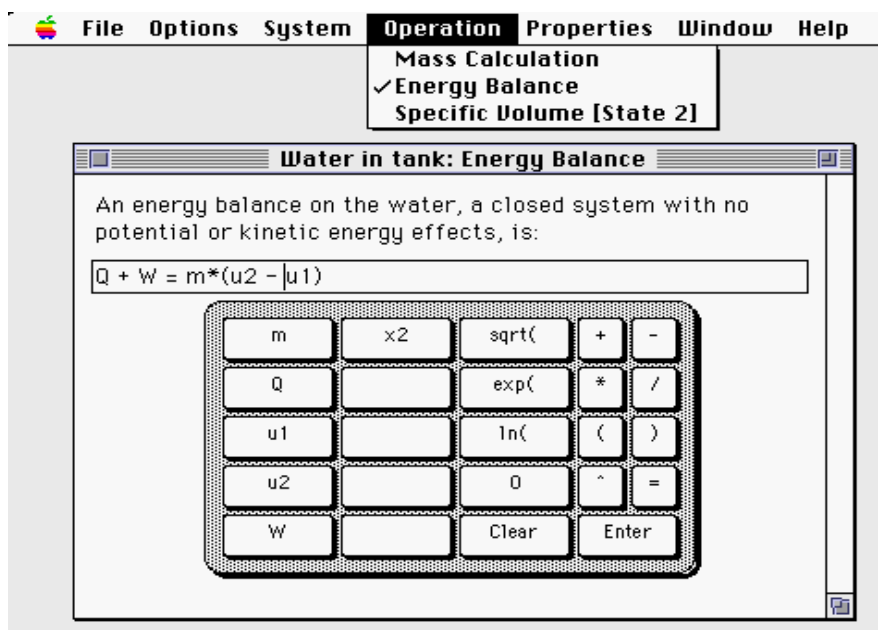


Figure 4: Equation input screen in the CP/Thermo program

The Help feature is, by far, the most important and complex part of the program. There are often many equally correct paths to a solution of the program. The Help processor keeps track of the student's progress and provides information upon request relating to the student's specific solution path. The first request for help provides a simple hint. Subsequent requests provide increasing more detail until finally, the Help processor reveals the answer to the sub-problem for which help was requested. Each request for help reduces the score by an amount related to the amount of help provided. The scoring feature was intended to keep students from frivolous use of the help.

A major point is that, with the help, a student can always complete a problem and obtain the correct solution. This is not the case with conventional homework. If a student cannot work a conventional problem at home, he/she must obtain help from a fellow student or the instructor or more likely, stop working on the problem. In some cases, the students do not realize that they have worked a problem incorrectly. In addition, conventional homework requires algebraic and problem lookups that are reduced when using CP/Thermo.

CP/Thermo was an innovative program, particularly considering the Apple II computing equipment on which it was initially developed. The program required years of programming

effort. However, this program was a failure for a number of reasons. First, by design, the program attempted to provide problems that were very similar to those included in the textbook. In effect, the computer was being used to do what students had become used to doing without a computer. The computational advantages of the computer were not fully utilized to allow students to do problems that they could otherwise not do. Second, the difficulty required to compose the problems and the associated help restricted this activity to the program author. Other instructors could not easily contribute problems. Third, the program did not attempt to eliminate algebra or property table lookups, but rather incorporated these requirements into the problems, thereby confusing computational and conceptual aspects of the problem. Fourth, students tended to not use the Help processor, the centerpiece of the CP/Thermo program, because they vainly did not want to have their score reduced, even when told the score would not be considered. Finally, the program structure was restricted such that students could not use it for design studies and ‘what-if’ problems. The program is no longer in use.

EES – Engineering Equation Solver

Most engineering problems can be separated into conceptual and mathematical parts. The conceptual part is the translation of the problem into a model, i.e., a set of mathematical relations. This part requires problem interpretation, justified assumptions, basic understanding of the underlying physics, and some creativity. It is unlikely that this part of the problem can be relegated to a computer. The mathematical part seeks an answer to the problem by solving the equations identified in the conceptual part, possibly with the addition of property data from handbooks. In practice, the mathematical part can be very time-consuming, although the algebraic manipulations and property table lookups add no additional insight or understanding after one knows how to do these chores. The mathematical details may in fact detract from the problem solving experience by introduction of algebra errors and by the confusion of mathematical and conceptual complexity. Furthermore, the optimization process associated with design ordinarily requires repeated calculations for a range of values of the design variables. Additional computations are needed to determine the sensitivity of the results to the uncertainties in the parameters. Such calculations require use of a computer tool.

One possibility that has been used extensively is to have the engineer directly code the mathematical parts of the problem into FORTRAN or another computer language. This is the approach used with the TRNSYS program. However, the coding process itself can itself be a major effort since the engineer may not be a proficient programmer. In fact, it appears that the programming skills of engineers have diminished in recent years as other computer tools have become available. While preparing the program, the problem can become diffused with the engineer focused more on the programming details than on the original engineering problem. Although the best approach for some engineering problems is to write a dedicated computer program, there are now other options such as spreadsheet programs, symbolic equation solvers and numerical equation solvers. Spreadsheet programs provide automated calculations and plotting capability and they have the advantages of being inexpensive, easy to use, widely available and familiar to many engineers. Spreadsheet programs are appropriate for some types of problems. However, it is difficult to apply spreadsheet programs to the task of solving sets of non-linear equations. In addition, the spreadsheet notation, useful for accounting applications, is cryptic and cumbersome when applied to most engineering problems.

Equation-solving programs can be viewed as high-level computer languages designed to help solve particular types of computational problems. Symbolic equation solvers, e.g., Mathematica⁴, and Maple⁵, provide computerized algebra and calculus and return symbolic equations. Numerical equation solvers, e.g., TK-Solver⁶ and MathCad⁷ and EES^{8,9} use numerical methods to solve sets of equations returning numerical values. The capabilities of these programs overlap since some provide both symbolic and numerical capabilities and some problems can be solved symbolically or numerically. The computational advantages offered by commercially available equation-solving programs are impressive but they are still lacking capabilities for some problem types such as those which occur in thermal systems. Hundreds of equations may be involved in a thermodynamic system analysis. These equations often have non-linearities introduced by heat transfer or fluid dynamics relations. A numerical solution of a large set of non-linear equations is, at best, painful to obtain with most existing commercial software. Further, few equation-solving programs provide the necessary thermodynamic and transport property data, although some allow the user to incorporate such information.

The need for a specialized tool to help solve realistic thermoscience problems motivated the development of EES. The basic function provided by the program is the numerical solution of a set of non-linear algebraic equations. Consider the six equations shown in the Equations Window in Figure 5. The equations are entered in a manner similar to that used in FORTRAN or other computer languages with the major difference being that the unknowns in the equation need not appear on the left side of an equal sign. Further, the equations can be entered in any order. A Formatted Equations window is provided to display the equations in more recognizable mathematical format. Selecting Solve from a menu produces the Solution window with calculated values for all variables. The order of the calculations and the accuracy to which the equations were solved is displayed in the Residuals window.

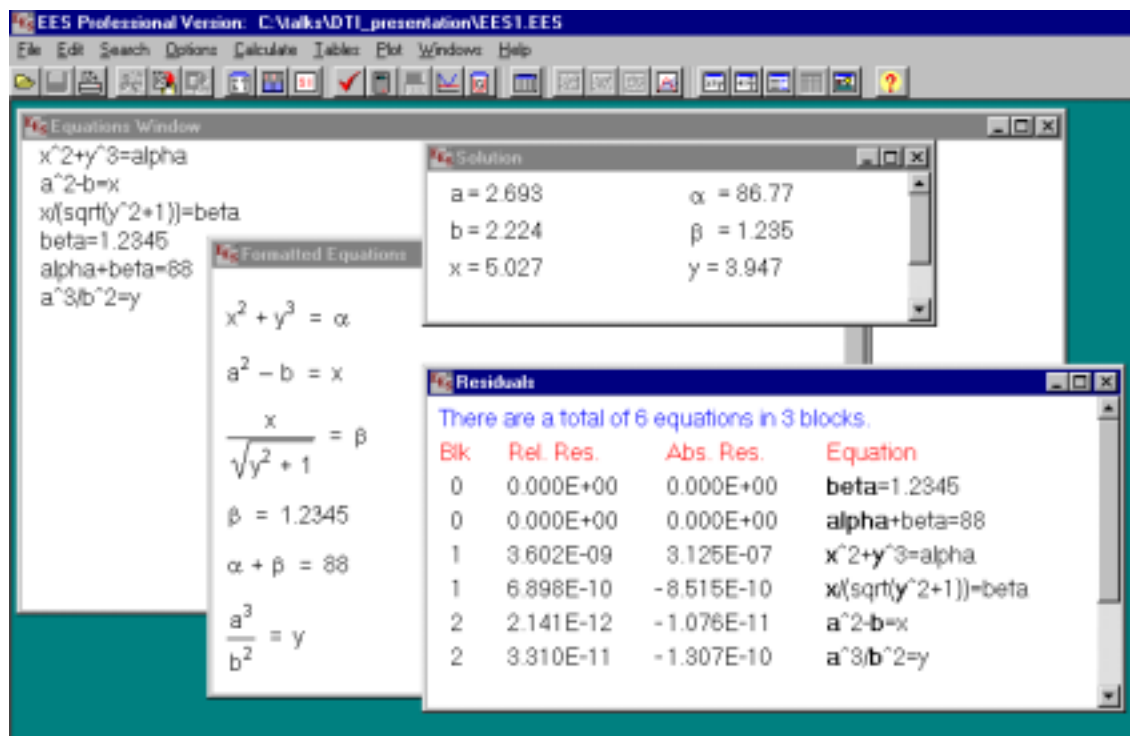


Figure 5: EES windows showing a simple problem and solution

EES is functionally similar to some existing programs, but it differs from them in that it has been designed from the start to incorporate the property information needed for solving problems in the thermal sciences area. For example, the steam tables are implemented such that any thermodynamic property can be obtained from a built-in function call in terms of any two other independent properties. Similar capability is provided for most refrigerants including R134a and R407C, R410A, ammonia, methane, propane, butane, isobutane, carbon dioxide, oxygen and many other fluids. Air tables are built-in, as are psychrometric functions and JANAF table data for many common gases¹⁰. Transport properties (viscosity, thermal conductivity, and surface tension) are also provided. Figure 6 shows a very simple program in which a Parametric table (which operates like a spreadsheet) is used to display calculated values of the pressure and density of saturated vapor for refrigerant R22 as a function of temperature. Shown in the background is a pressure-enthalpy figure for R22 generated by the program.

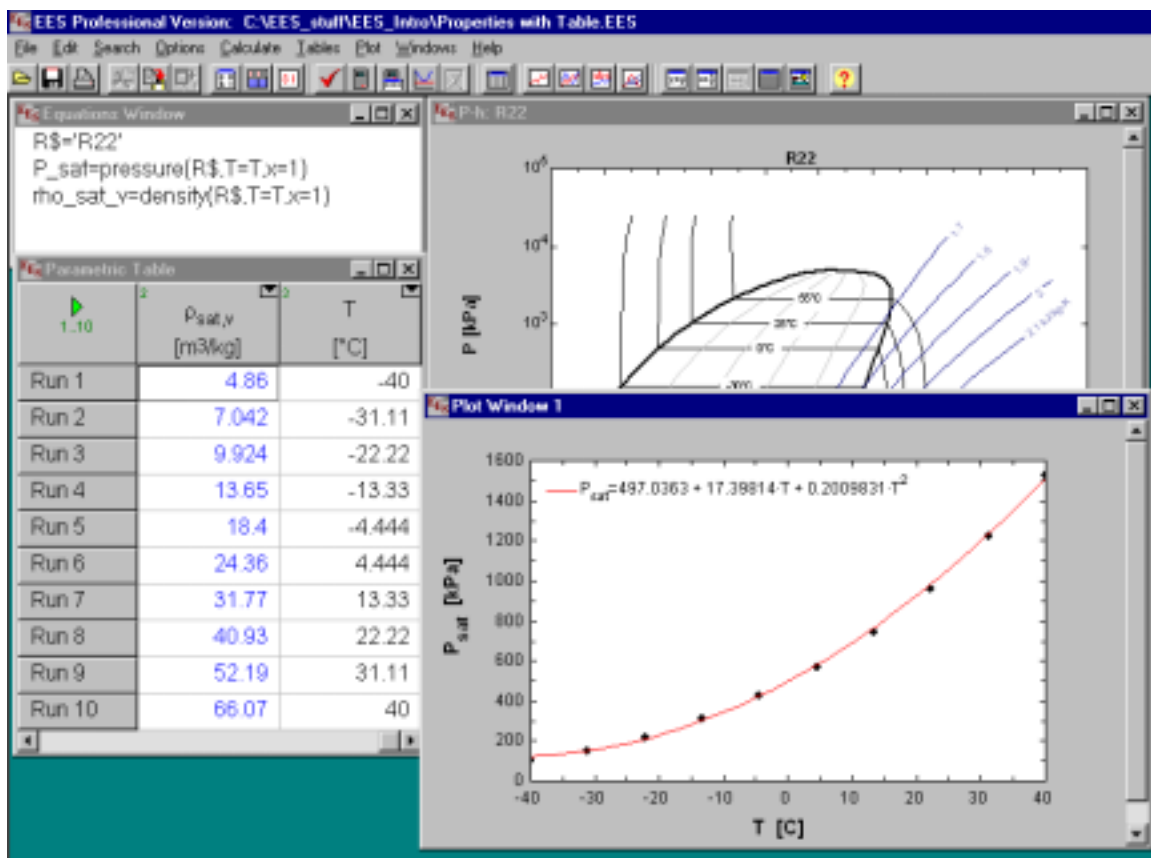


Figure 6: EES windows showing calculation of the density of vapor pressure of R22

Non-linear equations can have multiple solutions. The solution that results often depends upon the guess values. EES allows the guess value and the lower and upper bound of each variable to be specified, defaulting to a guess value of 1.0 with no bounds. A variant of Newton's method is used to solve systems of non-linear algebraic equations. The Jacobian matrix needed in Newton's method is evaluated numerically at each iteration. Sparse matrix techniques are employed to improve calculation efficiency and permit rather large problems to be solved efficiently. The efficiency and convergence properties of the solution method are further improved by implementation of an equation blocking algorithm which breaks the problem into a

number of smaller problems which are more easily solved¹¹. For example, EES would recognize that the equations entered in Figure 5 are not a coupled-set of six equations with six unknowns, but rather two single variable equations and two sets of two equations each with two unknowns. This blocking capability, which can be seen in the Residuals window of Figure 5, makes the solving algorithms more robust and increases the computational speed.

EES has many other capabilities. For example, it can solve differential and integral equations, do optimization and uncertainty analyses, provide linear and non-linear regression, convert units and check unit consistency, and most importantly, generate high-quality engineering plots. In addition, EES provides Diagram windows that can display graphics, text, user inputs, and calculated variables. Controls are provided to initiate calculations, show plots, and link to other applications. The Diagram window capabilities allow EES to appear as a stand-alone program developed for a particular task. Indeed EES allows the program to be saved as a stand-alone executable that can be run independently. These stand-alone programs have been used for both educational and commercial applications.

EES was originally developed for use in engineering thermodynamics classes but it is increasingly being used outside of the university. The program addresses limitations encountered in TRNSYS and CP/Thermo. Unlike TRNSYS, EES is very easy to use, requiring little training. It does away with the need to program in the traditional sense, although admittedly, the skill associated with writing code is helpful in understanding how EES works and optimizing its capabilities. Unlike CP/Thermo, the scope of problems that can be solved with EES is unlimited.

Conclusions

Some general observations can be made based on the author's experience in developing and using these three programs. The requirements for programs designed for education and research differ but they have a number of similarities. Today's users are sophisticated and demand many features, such as a graphical interface and compatibility with other software. One might think that students would be more tolerant than practicing engineers of a program that occasionally 'crashes', but that has not been found to be the case. All users demand high reliability. Programs having graphical interfaces and other related capabilities require significant development effort and require constant support to remain viable. Computing tools that provide capability to do tasks that can be done without a computer will not be readily adopted. Finally, programs that restrict the users' ability to do 'what-if' studies will ultimately not be useful. The capabilities of computers have developed much faster than the software needed to make them useful. There is a need for continued development of computer tools and integration of these tools into engineering curricula and industry.

References

1. Klein, S.A., Beckman, W.A. and Duffie, J.A., "TRNSYS – A Transient Simulation Program," ASHRAE Transactions, Vol. 82, part 1, pp. 623-633, (1976).
2. TRNSYS - A Transient System Simulation Program, Version 15, Solar Energy Laboratory, University of Wisconsin – Madison, (2000)

3. Klein, S.A. and Beckman, W.A. "Teaching Thermodynamics by the Book on a MicroComputer", *Computers in Mechanical Engineering*, pp.24-28, May, (1985)
4. Mathematica, Wolfram Research Inc, Addison-Wesley Publishing Co., 1988
5. Redfern, Darren, Maple V, Waterloo Maple Software, Brooks/Cole Publishing Co, Pacific Grove, CA, 1992
6. TK Solver Plus, Universal Technical Systems, Inc, 1220 Rock Street, Rockford, IL 611015.
7. MathCad, MathSoft Inc., 1 Kendall Sq., Cambridge, MA 02139
8. Klein, S.A., "Development and Integration of an Equation-Solving Program for Engineering Thermodynamics Courses," *Computer Applications in Engineering Education*, Vol. 1(3), pp. 265-265, (1973)
9. Klein, S.A. and Alvarado, F.L., EES – Engineering Equation Solver, F-Chart Software, 4406 Fox Bluff Rd, Middleton, WI, Version 6, (2000)
10. Stull, D.R., and Prophet, H., **JANAF Thermochemical Tables**, Second Edition, U.S. National Bureau of Standards, Washington, (1971)
11. Tarjan, R. "Depth-First Search and Linear Graph Algorithms," *SIAM J. Comput.* 1, 146-160, (1972)