

EESy Solutions

Engineering Equation Solver Newsletter

Inside this issue:

Welcome	1
The Component Library	1
<i>Accessing the Component Library</i>	2
<i>Component Model Information</i>	3
<i>Index of Component Models</i>	3
Example: A Simple GTE	5
Component Library, some last words	9
EES Training Opportunities	9
Additional Changes to EES	10
Instant Update Service	10

Welcome

This is the 28th issue of EESy Solutions, a newsletter that provides news, tips, and other updates for users of the Engineering Equation Solver software. This issue is dedicated to introducing the new component library in EES which provides a suite of models of commonly used thermal-system components (e.g., turbines, heat exchangers, valves, etc.)

EES has been a commercial software for more than two decades. If you have missed any of the previous issues of EESy Solutions, they can be downloaded from www.fchart.com.

The Component Library

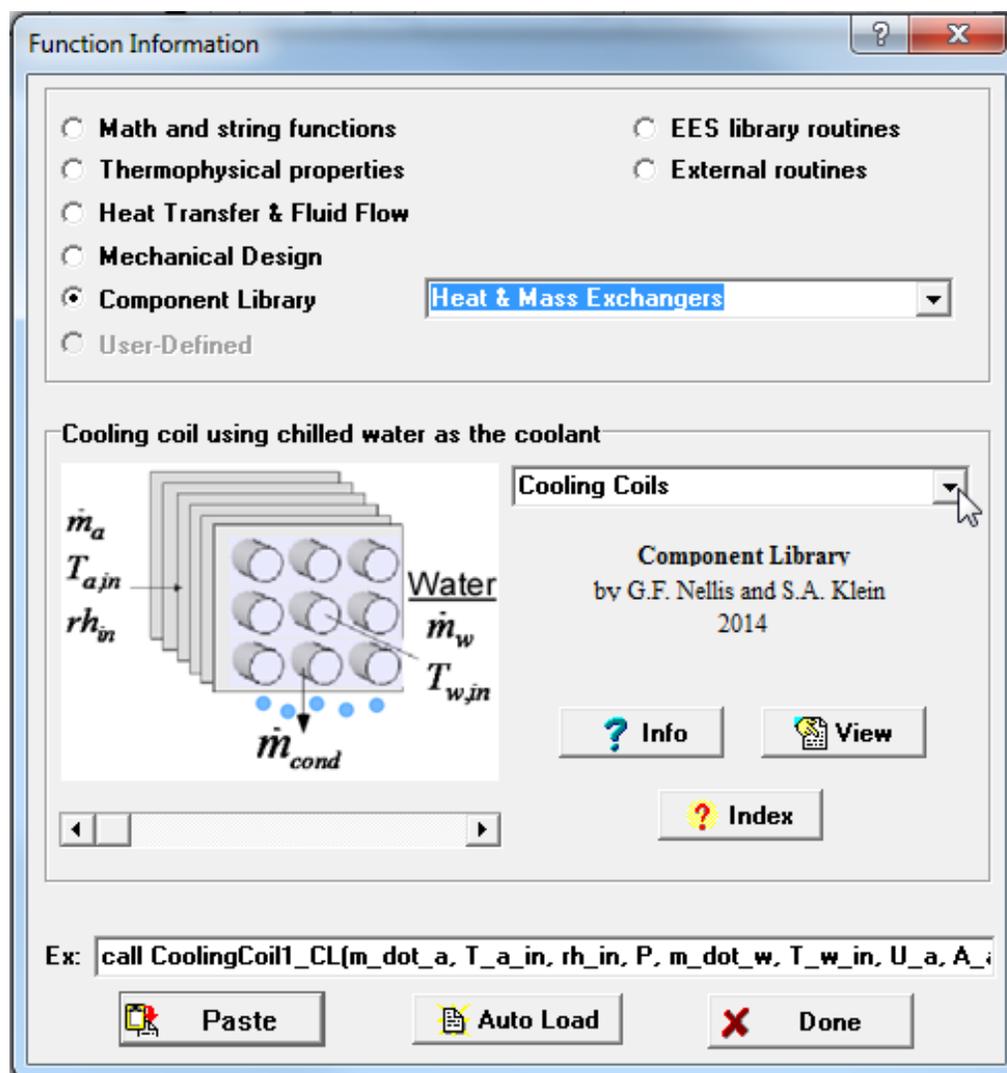
The Component Library is the newest application library provided with the Professional version of EES, joining the Heat Transfer and Mechanical Design libraries. The Component Library provides access to a set of EES functions and procedures that simulate components within thermal-fluid systems. Currently there are menus containing component models for Combustors, Compressors, Turbines, Fans & Blowers, Pumps, Heat Exchangers, Heat & Mass Exchangers, Nozzles & Diffusers, and Flow Control Devices.

The component models vary in complexity and functionality. Some are useful for simulation of a fixed piece of hardware whereas others are more useful for designing a piece of hardware to carry out a certain job. The tedium of programming and verifying the component simulation is removed when using the component library, allowing the user to focus on the complexity of the system simulation where multiple components are connected together to simulate systems.

Preparing a detailed simulation of a system requires that many component models to be linked together so that they provide a consistent set of inputs and outputs. This is often difficult to achieve in practice and will usually require sophisticated strategies for obtaining convergence. The use of the component library enables the user to quickly try various methods for obtaining system level convergence.

Accessing the Component Library

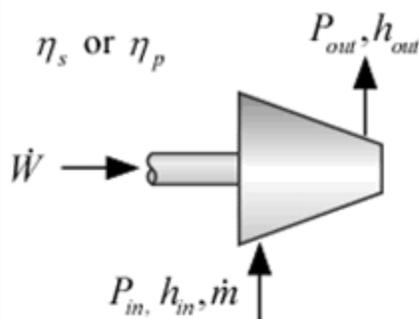
The Component Library is accessed from the Function Information selection in the Options menu. Click the Component Library radio button and select the category of interest from the dropdown menu. A second dropdown menu will appear in some cases, allowing the category to be further refined. The scrollbar allows the user to select from the various component models that are available.



Component Model Information

Select the Info button to obtain information about the origin and correct use of the model. The information page includes a description of the model and how it is intended to be used as well as references. An example of a call to the function or procedure and the associated outputs is included.

Compressor2_CL



Procedure Compressor2_CL provides a model of a compressor with a specified isentropic or polytropic efficiency. The model operates both within and outside of vapor dome.

Inputs:

h_{in} : inlet enthalpy (J/kg, kJ/kg, or Btu/lbm)
 P_{in} : inlet pressure (bar, atm, Pa, kPa, MPa)
 P_{out} : outlet pressure (bar, atm, Pa, kPa, MPa)
 \dot{m} : mass flow rate (kg/s or lb_m/hr)
 $F\$$: fluid string identifier
 η : if $\eta > 0$ then it is interpreted as the isentropic efficiency
 if $\eta < 0$ then $abs(\eta)$ is interpreted as the polytropic efficiency

Outputs:

h_{out} : outlet enthalpy (J/kg, kJ/kg, or Btu/lbm)
 \dot{W} : power (W, kW or Btu/hr)
 η_s : isentropic efficiency (this is useful if polytropic efficiency is an input)

Example:

```

$UnitSystem SI Mass J K Pa
F$='He'
P_in=1e5 [Pa]
h_in=Enthalpy(F$,T=300 [K])
P_out=1e6 [Pa]
m_dot=0.1 [kg/s]
eta= 0.9 [-]
Call Compressor2_CL(h_in, P_in, P_out, m_dot, F$, eta: h_out, W_dot, eta_s)
  
```

{Solution:

```

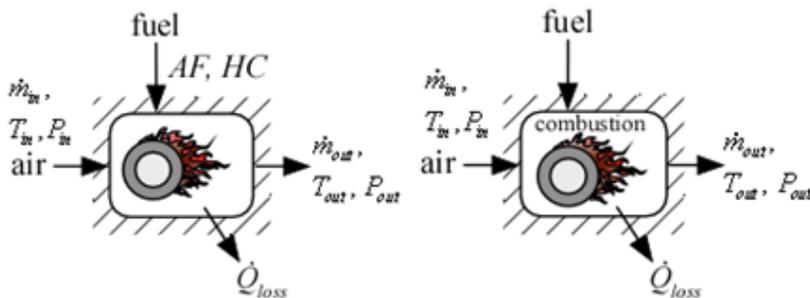
h_out = 2.626e6 J/kg
W_dot = 261683 W
eta_s = 0.9}
  
```

Index of Component Models

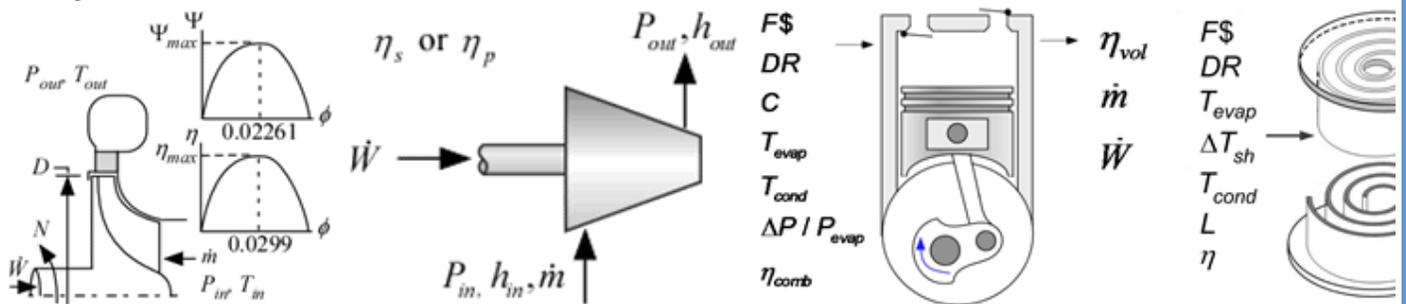
Click the Index button in the Function Information dialog in order to see an index of the available component models. The index is live. Clicking on one of the component pictures will bring up the Information page for that component model.

Index

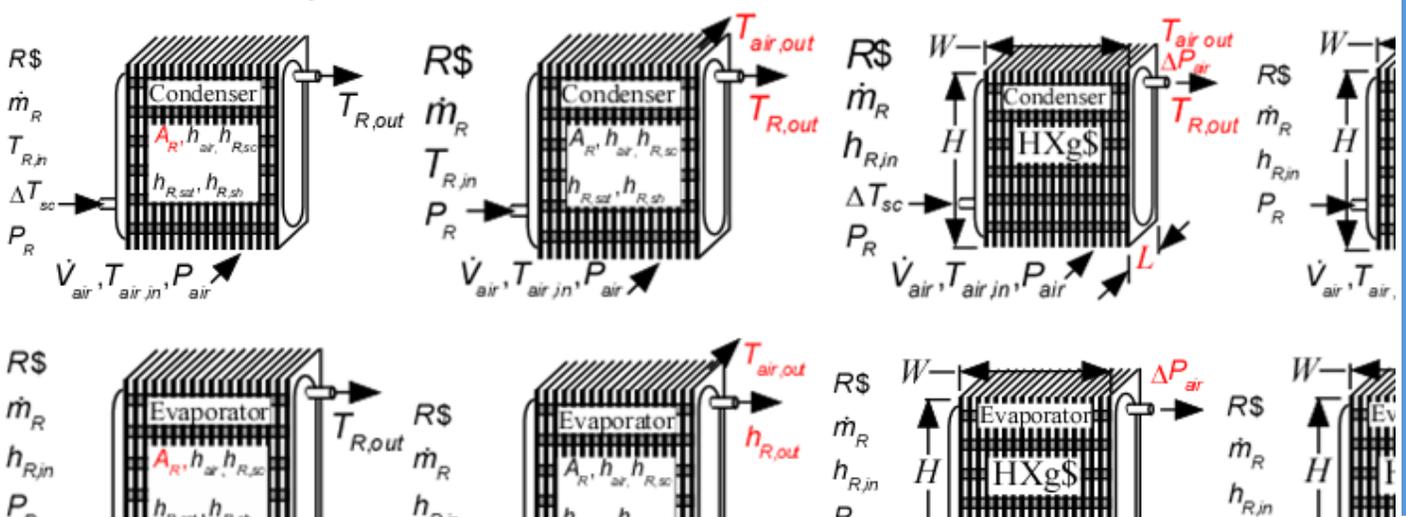
Combustors



Compressors

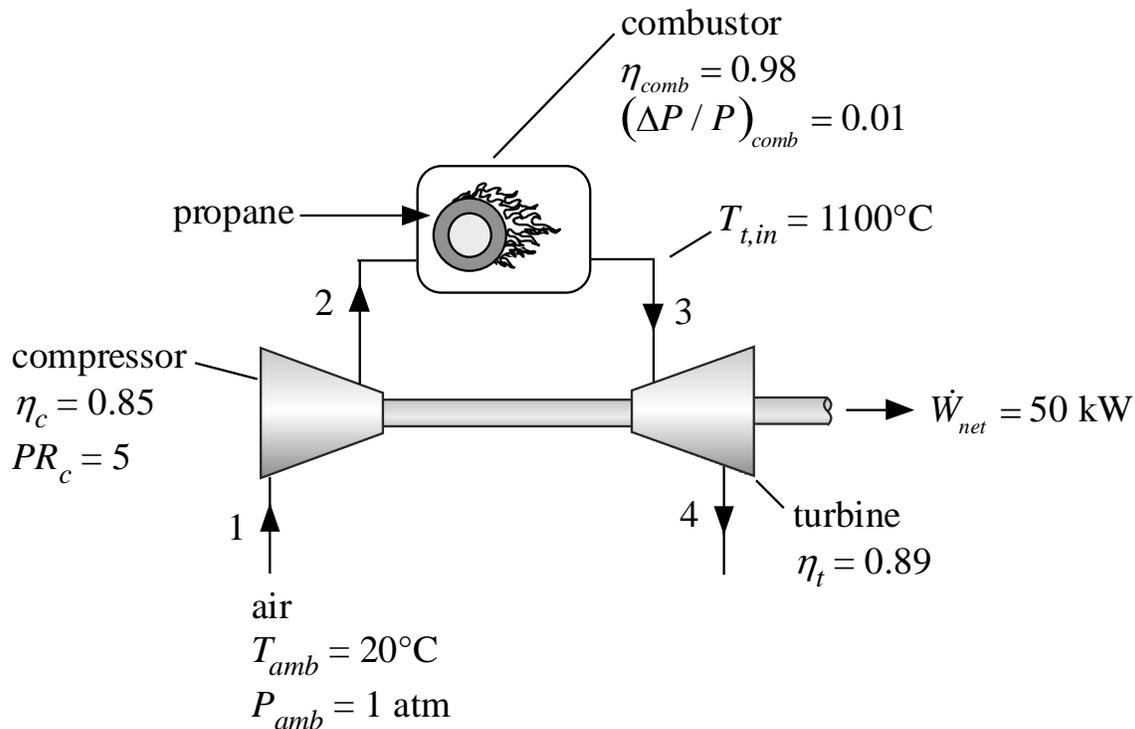


Condensers and Evaporators



Example: A Simple Gas Turbine Engine

In order to illustrate the use of the Component Library, a simulation model will be developed for the air-breathing gas turbine engine shown below.



Atmospheric air is drawn into the compressor at $T_{amb} = 20^\circ\text{C}$ and $P_{amb} = 1 \text{ atm}$. The compressor has an isentropic efficiency of $\eta_c = 0.85$ and provides a pressure ratio of $PR_c = 5$. The compressed air is combusted with propane in a combustor with efficiency $\eta_{comb} = 0.98$ and pressure loss $(DP/P)_{comb} = 0.01$. The maximum allowable temperature leaving the combustor and entering the turbine is $T_{t,in} = 1100^\circ\text{C}$. The combustion products pass through a turbine with isentropic efficiency $\eta_t = 0.89$. The gas turbine engine must produce a net power of 50 kW. These operating conditions and performance parameters are entered in EES:

```
$UnitSystem SI Mass J K Pa
```

```
"Operating conditions"
```

```
T_amb=ConvertTemp(C,K,20[C])
P_amb=1 [atm]*convert(atm, Pa)
F$='Air'
Fuel$='C3H8'
PR_c=5
W_dot_net=50 [kW]*convert(kW,W)
T_t_in=ConvertTemp(C,K,1100 [C])
```

```
"ambient temperature"
```

```
"ambient pressure"
```

```
"fluid"
```

```
"fuel"
```

```
"compressor pressure ratio"
```

```
"required net power"
```

```
"required turbine inlet temperature"
```

```
"Component efficiencies"
```

```
eta_c=0.85
```

```
eta_comb=0.98
```

```
DPOverP_comb=0.01
```

```
eta_t=0.89
```

```
"isentropic efficiency of compressor"
```

```
"combustor efficiency"
```

```
"relative pressure drop in combustor"
```

```
"isentropic efficiency of turbine"
```

Gas Turbine Engine Example

The cycle is solved starting at the compressor inlet and working in the flow direction. The compressor suction mass flow rate is assumed; it will eventually be adjusted to achieve the required power. The conditions at the compressor inlet are used to specify state 1. The pressure at the compressor exit is specified based on the compressor pressure ratio.

$m_dot_c=0.1$ [kg/s] "mass flow rate entering compressor"

"State 1, entering compressor"

$T[1]=T_amb$

"temperature"

$P[1]=P_amb$

"pressure"

$h[1]=Enthalpy(F\$,T=T[1])$

"enthalpy"

$m_dot[1]=m_dot_c$

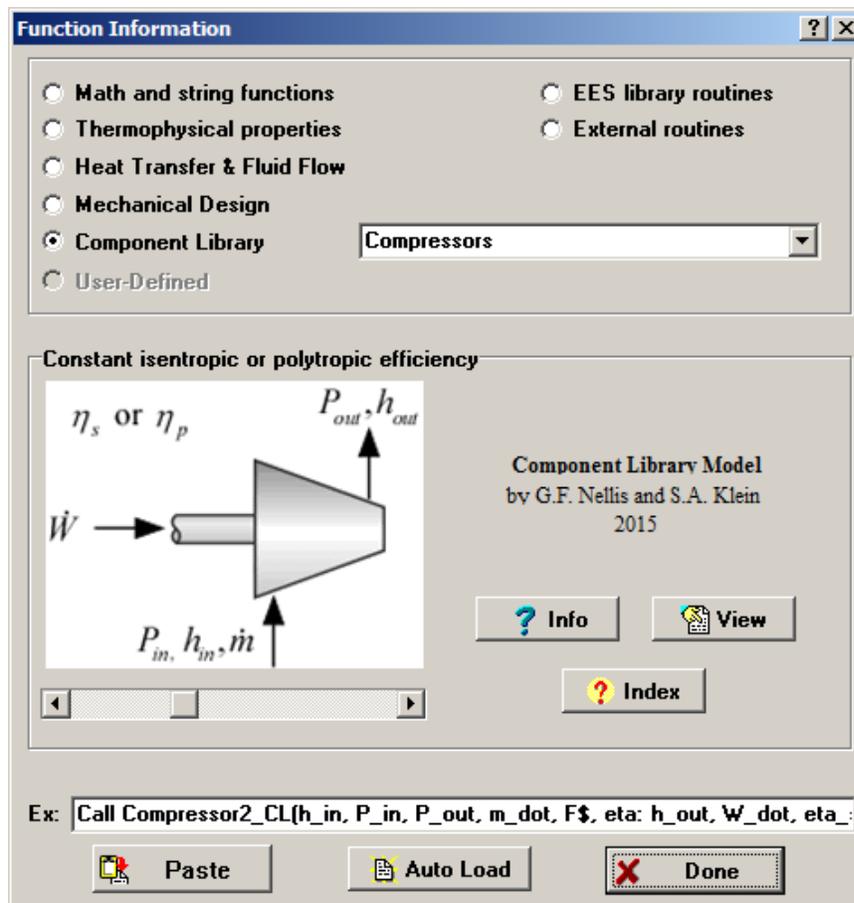
"mass flow rate"

"State 2, leaving compressor"

$P[2]=P[1]*PR_c$

"pressure"

The procedure Compressor2_CL is accessed from the Component Library by selecting Function Information from the Options menu and then clicking on the Component Library radio button. Select the Compressors category from the pull down menu and scroll through the available compressor models to display Compressor2_CL, which is a simple model of a compressor that assumes a constant efficiency (isentropic or polytropic). Select Paste to automatically place the procedure call in the Equations Window.



Gas Turbine Engine Example

The Compressor2_CL procedure returns the outlet enthalpy and compressor power. An optional third output provides the polytropic efficiency. In the call below, the third output is not assigned to a variable as no variable is placed in the third spot in the output list. The temperature and mass flow rate leaving the compressor are determined.

```
Call Compressor2_CL(h[1], P[1], P[2], m_dot[1], F$, eta_c: h[2], W_dot_c,)
                                     "call compressor component"
T[2]=Temperature(F$, h=h[2])         "temperature at state 2"
m_dot[2]=m_dot[1]                    "mass flow rate"
```

The model Combustor2_CL is used to simulate the combustor. The model requires that the amount of excess air and chemical structure of the fuel be specified and returns the combustor outlet temperature as well as other performance parameters. Initially the % excess air is assumed and it will eventually be adjusted to achieve the specified turbine inlet temperature.

```
Excess%=200 [%]                       "% excess air in combustor"

"State 3, leaving combustor"
Call Combustor2_CL(T[2], P[2], m_dot[2], Fuel$, Excess%, eta_comb, DPoverP_comb: m_dot[3], &
  T[3], P[3], Q_dot_f, Q_dot_loss)     "call combustor component"
h[3]=Enthalpy(F$, T=T[3])             "enthalpy at state 3"
```

Finally, the turbine is simulated using the procedure Turbine2_CL.

```
"State 4, leaving turbine"
P[4]=P_amb                             "pressure"
Call Turbine2_CL(h[3], P[3], P[4], m_dot[3], F$, eta_t: h[4], W_dot_t, )
                                     "call turbine component"
T[4]=Temperature(F$, h=h[4])         "temperature at state 4"
m_dot[4]=m_dot[3]                    "mass flow rate"
```

The net power and efficiency are both computed.

```
"Cycle Performance"
W_dot_GTE_net=W_dot_t-W_dot_c         "net power produced by GTE"
eta=W_dot_net/Q_dot_f                "efficiency"
```

Gas Turbine Engine Example

The guess values are updated in order to provide a good starting point for the iterative calculations that will be required to obtain the correct power and turbine inlet temperature. The assumed values of mass flow rate and excess air are commented out and the net power and turbine inlet temperature are specified.

```
$UnitSystem SI Mass J K Pa
```

```
"Operating conditions"
```

```
T_amb=ConvertTemp(C,K,20[C])           "ambient temperature"
P_amb=1 [atm]*convert(atm, Pa)         "ambient pressure"
F$='Air'                                "fluid"
Fuel$='C3H8'                             "fuel"
PR_c=5                                   "compressor pressure ratio"
W_dot_net=50 [kW]*convert(kW,W)         "required net power"
T_t_in=ConvertTemp(C,K,1100 [C])       "required turbine inlet temperature"
```

```
"Component efficiencies"
```

```
eta_c=0.85                               "isentropic efficiency of compressor"
eta_comb=0.98                             "combustor efficiency"
DPOverP_comb=0.01                         "relative pressure drop in combustor"
eta_t=0.89                               "isentropic efficiency of turbine"
```

```
{m_dot_c=0.1 [kg/s]}                    "mass flow rate entering compressor"
```

```
"State 1, entering compressor"
```

```
T[1]=T_amb                               "temperature"
P[1]=P_amb                               "pressure"
h[1]=Enthalpy(F$,T=T[1])                 "enthalpy"
m_dot[1]=m_dot_c                          "mass flow rate"
```

```
"State 2, leaving compressor"
```

```
P[2]=P[1]*PR_c                           "pressure"
Call Compressor2_CL(h[1], P[1], P[2], m_dot[1], F$, eta_c: h[2], W_dot_c,)
                                         "call compressor component"
T[2]=Temperature(F$, h=h[2])              "temperature at state 2"
m_dot[2]=m_dot[1]                         "mass flow rate"
```

```
{Excess%=200 [%]}                       "% excess air in combustor"
```

```
"State 3, leaving combustor"
```

```
Call Combustor2_CL(T[2], P[2], m_dot[2], Fuel$, Excess%, eta_comb, DPOverP_comb: m_dot[3], &
  T[3], P[3], Q_dot_f, Q_dot_loss)         "call combustor component"
h[3]=Enthalpy(F$,T=T[3])                 "enthalpy at state 3"
```

```
"State 4, leaving turbine"
```

```
P[4]=P_amb                               "pressure"
Call Turbine2_CL(h[3], P[3], P[4], m_dot[3], F$, eta_t: h[4], W_dot_t, )
                                         "call turbine component"
T[4]=Temperature(F$,h=h[4])              "temperature at state 4"
m_dot[4]=m_dot[3]                         "mass flow rate"
```

```
"Cycle Performance"
```

```
W_dot_GTE_net=W_dot_t-W_dot_c             "net power produced by GTE"
eta=W_dot_net/Q_dot_f                     "efficiency"
```

```
T[4]=T_t_in                               "required turbine inlet temperature"
W_dot_GTE_net=W_dot_net                   "required net power"
```

The Component Library, some last words...

The simple example presented here illustrates the process of accessing the Component Library and using the models. This was a very simple cycle and the models were also quite simple. More complex models are available in the library and these will be continuously augmented and improved.



The ability to create component models has been part of EES since its inception. The Component Library represents the next logical step in the development of EES. It provides the ability to carry out detailed design and analysis of complex systems by connecting existing models of components together. These component models rely on the existing thermo-physical property library as well as the heat transfer library within EES. Since the majority of the component models are written in EES it is possible to copy the EES code and modify it to meet your specific needs.



The process of getting a system of components to converge to a self-consistent solution is not trivial, even when using EES. The user will typically need to devise an iterative solution strategy involving initial guesses that are eventually removed to force convergence. The choice of the starting point for the cycle analysis and evolution of the solution strategy is something of an art and it will often require multiple attempts to obtain a robust cycle model. The Component Library allows the user to focus on the system-level analysis without being distracted by the details of the component models themselves.



EES Training Opportunities

EES is used in many companies, organizations, and academic institutions. Most users are aware of the basic features—solving equations, Parametric Tables, plotting, and the powerful property routines. However, fewer users are aware of the more powerful features of EES such as optimization, integration, uncertainty propagation, complex algebra and subprograms. Even more advanced features available in the Professional version, such as the Diagram Window, animation, executables, web-based executables, directives, and macros, can enhance the capabilities of the program and open the door to a wide range of applications.

Our staff are available to provide a 1 day EES short course tailored to new users or experienced EES users. Contact Greg Nellis by email to arrange a training session (info@fchart.com).



F-Chart Software

PO Box 44042
Madison, WI, 53744

Phone/FAX: 608-274-4262
Internet: <http://fchart.com>
E-mail: info@fchart.com

<http://fchart.com>

Some Additional Changes to EES

- The functions Enthalpy_formation, LowerHeatingValue, and HigherHeatingValue provide these quantities for ideal gases in EES, the NASA library, and incompressible organic liquids
- The StdDev function provides the standard deviation of an array
- Improved thermodynamic property data have been implemented for cyclohexane and DeuteriumOxide (i.e., Heavy Water)
- \$If and \$IfNot directives will now accept Boolean expressions using the OR and the AND operators
- The \$RunMacroAfter directive allows macros to be embedded in the Equations Window and executed after calculations are complete
- Functions, procedures, and subprograms can now operate using their own, separate and independent unit system by placing a \$UnitSystem directive within the program unit
- The behavior of the plot crosshairs tool has been modified. The crosshairs can now be used to identify the location of a plotted point within the table containing the data by pressing the right mouse button with the crosshair centered on the point in question.
- The ability of EES to debug a program in which the number of equations and variables are not equal has been greatly improved. EES will now provide a Solve button within the Debug Window that causes it to solve equations up to the point where EES has detected that a missing equation should appear.

Instant Update Service

EES uses a different model for updating than most other programs. Each time that there is a change in the EES program, either to correct a problem or to add a new feature, the version number is incremented by 0.001 and the latest version of EES is placed on our website. Although the program has become very robust and stable, there have been many new versions of EES released since the last EESy Solutions was distributed.

Any user who has a current subscription to our Instant Update Service can download the latest version. All new licenses of EES are provided with one year of Instant Update Service. The fee to continue Instant Update Service after the first year is 20% of the current cost of the program per year if renewed within 12 months after expiration of the Service. Contact F-Chart Software if you wish to re-subscribe to the Instant Update Service.

