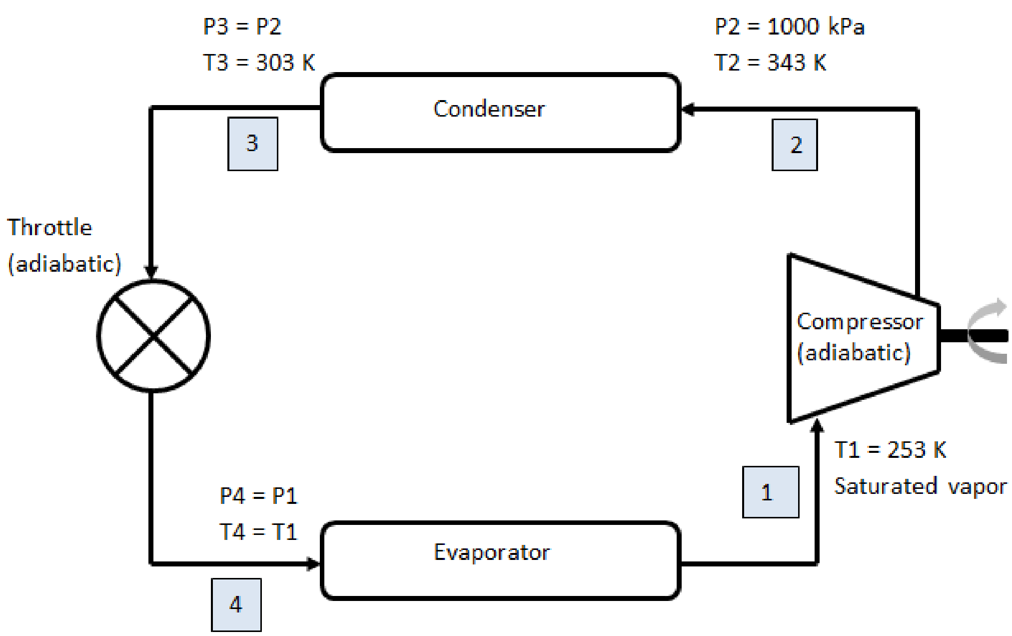
Analysis of a Vapor-Compression Refrigeration Cycle

Introduction

This application analyzes this refrigeration cycle. Heat losses and gains over the compressor, condensor, throttle and evaporator will be calculated, together with the coefficient of performance.



Additionally, a P-h-T chart with the refrigeration cycle will be plotted.

The compressor, condenser, throttle and evaporator are analyzed separately from each other. Calculations are performed by analyzing the specific energy change aross each component with this equation (a statement of the conservation of energy),

$$q - w = \Delta h + \Delta KE + \Delta PE$$

where

- w is the work done by the component
- ΔKE and ΔPE are the changes in kinetic and potential energy
- Δh is the change in specific enthalpy
- q is the heat transferred to the system

Thermophysical properties are provided by the open source C++ CoolProp library (coolprop.org). Once compiled and linked to Maple, CoolProp lets you access the properties of pure fluids, pseudo-pure fluids, and humid air with a function call.

This application comes with a CoolProp DLL for 64-bit Windows. You may need to compile CoolProp on your own computing environment for a compatible library.

While this application uses R134a as the refrigerant, you can investigate the effect of other fluids by replacing references to R134a with your preferred fluid (such as R744).

Load the CoolProp DLL

```
> restart
```

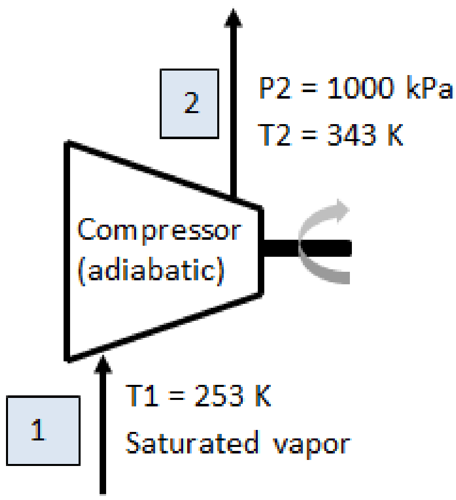
Ensure the path to the DLL is correctly referenced by define_external()

```
> Props := define_external( 'Props', LIB = "c:\\CoolProp_x64.dll", 'Output'::string, 'In1'::char, 'Val1'::(float[8]), 'In2'::char, 'Val2'::(float[8]), 'Fluid'::string, RETURN::(float[8]) ) :
```

Valid calling sequences for Props() are documented at coolprop.org

Compressor

Consider the energy flows in the compressor. For an adiabatic process, q = 0. Also ΔKE = 0 and ΔPE = 0. Hence -w = Δh



Therefore the enthalpy at points 1 and 2 are (in kJ kg^{-1})

```
> h1 := Props("H", "T", 253, "Q", 1, "R134a")
  h2 := Props("H", "T", 343, "P", 1000, "R134a")
```

h1 := 386.461535847473840

h2 := 451.844244097570993 **(3.1)**

The work done by the compressor (w) is

```
> workCompressor := h1 − h2
```

workCompressor := −65.3827083 **(3.2)**

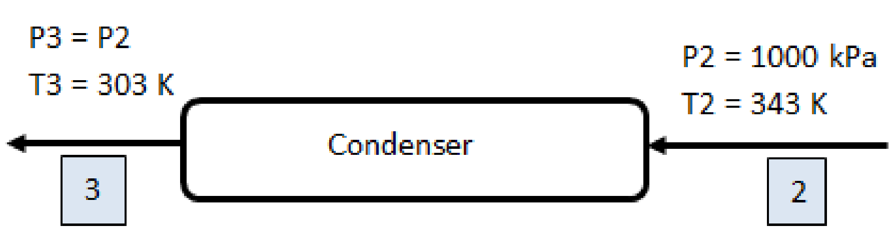
```
> press1 := Props("P", "T", 253, "Q", 1, "R134a")
```

press1 := 131.876928437669761 **(3.3)**

```
> press2 := 1000 :
```

Condenser

For the condenser, w = 0, ΔKE = 0 and ΔPE = 0. Hence q = Δh




```
> h3 := Props("H", "T", 303, "P", 1000, "R134a")
```

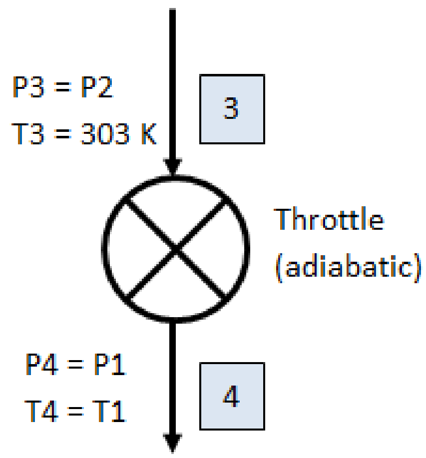
h3 := 241.499519016077613 **(4.1)**

> h3 − h2

-210.3447251 **(4.2)**

Throttle

For the throttle, q = 0, w = 0, ΔKE = 0 and ΔPE = 0. Hence Δh = 0



Enthalpy at h4

> h4 := h3

h4 := 241.499519016077613 **(5.1)**

Saturation pressure of R134a at 253 K

> press4 := press1

press4 := 131.876928437669761 **(5.2)**

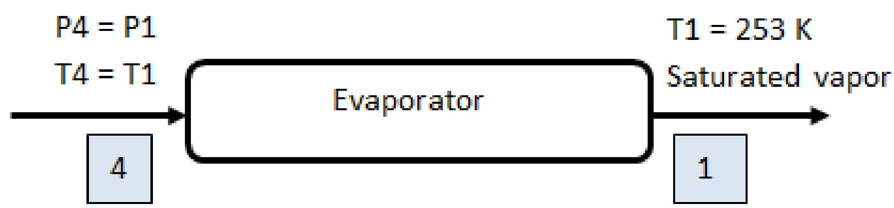
Quality at P = press4 and H = h4

> Props("Q", "P", press4, "H", h4, "R134a")

0.319491031236869172 **(5.3)**

Evaporator

For the evaporator, w = 0, ΔKE = 0 and ΔPE = 0. Hence q = Δh



Heat extracted by evaporator

```
> heatEvaporator := h4 − h1
                    heatEvaporator := −144.9620168                    (6.1)
```

Coefficient of Performance

```
>  heatEvaporator
   ---------------
   workCompressor
                          2.217130807                                  (7.1)
```

Plot the Refrigeration Cycle on a P-h-T Chart

```
> with(plots) :
> isotherm := contourplot('Props'("T", "H", h, "P", p, "R134a"), h = 150..460, p = 60..10000,
     axis[1] = [gridlines = [color = "LightSteelBlue"]], axis[2] = [mode = log, gridlines = [8, subticks
     = 4, color = "LightSteelBlue"]], contours = [seq(260 + i, i = 0..200, 20)], thickness = 0,
     color = "DarkGray", grid = [150, 150]) :
> isothermText := textplot([seq([Props("H", "T", 260 + i, "P", 4000, "R134a"), 4000,
     sprintf("%a K", 260 + i)], i = 0..200, 20)], font = [Helvetica, Normal]) :
> twoPhaseEnvLeft := pointplot([seq([Props("H", "P", i, "Q", 1, "R134a"), i], i = 0..4030)],
     connect = true, thickness = 0) :
> twoPhaseEnvRight := pointplot([seq([Props("H", "P", i, "Q", 0, "R134a"), i], i = 0..4040)],
     connect = true, thickness = 0) :
> quality := seq(pointplot([seq([Props("H", "P", p, "Q", q, "R134a"), p], p = 60..4000, 100)],
     connect = true, color = "LightGreen", thickness = 0), q = 0.1..0.9, 0.1) :
> qualityText := textplot([seq([Props("H", "P", 150, "Q", 0.01·q, "R134a"), 150,
     sprintf("%a%%", q)], q = 10..90, 10)], font = [Helvetica, Normal]) :
> cycle := pointplot([[h1, press1], [h2, press2], [h3, press2], [h3, press4], [h1, press1]],
     connect = true, color = "DarkRed", thickness = 2) :
> display(isotherm, isothermText, twoPhaseEnvLeft, twoPhaseEnvRight, cycle, quality,
     qualityText, view = [150..460, 100..5000], title = "P-h-T Chart for R134a", titlefont
     = [Helvetica, 14], labels = ["Enthalpy (kJ/kg)", "Pressure (kPa)"], labeldirections
     = [horizontal, vertical], labelfont = [Helvetica, 12])
```

P-h-T Chart for R134a