



Prof. C. Naaktgeboren, PhD  
<https://github.com/CNThermSci/AplThermSci>  
 This example is licensed under a [Creative Commons BY-NC-SA License](#).



## B04 – Ciclos de Refrigeração

### 01 – Ciclos de Refrigeração por Compressão de Vapor de Simples Estágio

#### Exemplo B0401-02 – Ciclo de Refrigeração por Compressão de Vapor de Simples Estágio

Original.

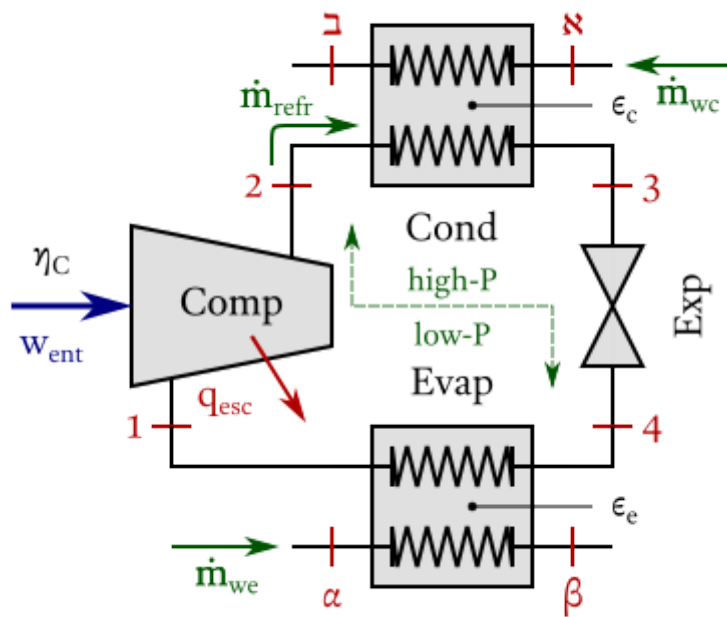
```
• prob = Dict(
•   :PR => 1.5: 0.50: 4.5, # Potência de Refrigeração (tons)
•   :Tα => 15.0: 2.50: 25.0, # Temperatura da água a resfriar (°C)
•   :Tβ => 2.0: 1.00: 4.0, # Temperatura de água fria (°C)
•   :ec => 50.0: 5.00: 65.0, # Efetividade do condensador (%)
•   :ee => 70.0: 5.00: 85.0, # Efetividade do evaporador (%)
•   :T4 => -10.0: 10.0: 0.0, # Temperatura do evaporador (°C)
•   :T3 => 40.0: 10.0: 60.0, # Temperatura do condensador (°C)
•   :Tκ => 10.0: 2.50: 20.0, # Temperatura da água a aquecer (°C)
•   :ηC => 75.0: 5.00: 90.0, # Eficiência isentrópica, %
•   :IC => 20.0: 5.00: 45.0, # Irrev. perda no compressor, %
• );
```

Recompute

```
▼Dict(
  :T3 => 50.0
  :PR => 4.5
  :Tκ => 12.5
  :IC => 30.0
  :ec => 60.0
  :Tα => 15.0
  :ηC => 75.0
  :ee => 80.0
  :T4 => 0.0
  :Tβ => 2.0
)
```

## Enunciado:

Deseja-se obter **4.5 ton** de refrigeração na produção de água gelada—corrente “ $\alpha$ - $\beta$ ”, na qual água entra em contra-corrente no evaporador a pressão atmosférica e **15.0°C**, devendo sair a **2.0°C**. Água quente a pressão atmosférica é produzida na contra-corrente “ $\beta$ - $\gamma$ ” do condensador, no qual água entra a **12.5°C** e as correntes possuem capacidade balanceadas. As efetividades do condensador e evaporador são, respectivamente, de **60.0%** e **80.0%**. O sistema de refrigeração opera com fluido refrigerante **R22**, temperatura de saída da válvula de expansão de **0.0°C** e temperatura de condensação do lado do refrigerante de **50.0°C**, eficiência isentrópica de compressão de **75.0%** com perda de **30.0%** da taxa de irreversibilidade na forma de calor para o meio, conforme indicado. Determine:



- A vazão mássica de refrigerante, em kg/s
- A vazão mássica de água gelada produzida em “ $\beta$ ”, em kg/s
- A vazão mássica de água quente produzida em “ $\gamma$ ”, em kg/s
- A temperatura da água produzida em “ $\gamma$ ”, em °C
- O COP do refrigerador, em %
- O número de unidades de transferência, NTU, do condensador
- O número de unidades de transferência, NTU, do evaporador
- O coeficiente global de transferência de calor, UA, do evaporador

## Dados:

O número de unidades de transferência,  $NTU$ , é definido como

$$NTU \equiv \frac{UA}{\dot{C}_{min}},$$

onde  $UA$  é o coeficiente global de transferência de calor do trocador de calor, e  $\dot{C}_{min}$  a menor das taxas de capacidade entre as correntes quente (q) e fria (f):

$$\dot{C}_{min} \equiv \min(\dot{C}_q, \dot{C}_f),$$

e  $\epsilon$  sendo a *efetividade* do trocador de calor, definida como:

$$\epsilon \equiv \frac{\dot{Q}}{\dot{Q}_{max}} = \frac{\pm(\dot{m}\Delta h)_{q,f}}{\dot{C}_{min}(\Delta T_{max})} = \frac{\pm(\dot{m}\Delta h)_{q,f}}{\dot{C}_{min}(T_{q,ent} - T_{f,ent})}.$$

Para trocadores em contra-corrente, tem-se:

$$NTU = \frac{\ln\left(\frac{1 - \epsilon\delta}{1 - \epsilon}\right)}{1 - \delta},$$

com

$$\delta \equiv \frac{\dot{C}_{min}}{\dot{C}_{max}}.$$

A relação inversa,  $\epsilon(NTU)$ , assume formas especiais para (i) trocadores *balanceados* ( $\delta = 1$ ):

$$\epsilon = \frac{NTU}{1 + NTU}, \quad (\delta = 1),$$

e para (ii)  $\dot{C}_{max} \rightarrow \infty$ , a saber, quando uma das correntes não experimenta variação de temperatura, a exemplo de substância pura em troca de fase à pressão constante:

$$\epsilon = 1 - e^{-NTU}, \quad (\delta = 0).$$

## Resolução

Escreve-se uma função que resolve o ciclo, utilizando **CoolProp** via **Pycall.jl** para propriedades termodinâmicas.

solve (generic function with 1 method)

```
• function solve(PR, Tα, Tβ, εc, εe, T4, T3, Tκ, ηC, IC; FL="R134a")
•     # Implement-me...
•     # Cycle States
•     St1 = CP.State(FL, Dict{"T" => Te, "Q" => 1}) # All T's in K
•     St3 = CP.State(FL, Dict{"T" => Tc, "Q" => 0})
•     S2s = CP.State(FL, Dict{"P" => St3.p, "S" => St1.s})
•     wCs = S2s.h - St1.h # Isentropic compressor work
```

```

•   wCr = wCs / ηC      # ηC normalized
•   IrC = wCr - wCs     # Irreversibility, normalized
•   qCs = IC * IrC      # Compressor heat loss
•   h_2 = St1.h + wCr - qCs # Energy balance
•   St2 = CP.State(FL, Dict{"P" => St3.p, "H" => h_2})
•   St4 = CP.State(FL, Dict{"P" => St1.p, "H" => St3.h})
•   # Quantities of interest
•   md = WC / wCr
•   q23 = St2.h - St3.h
•   q41 = St1.h - St4.h
•   COP = q41 / wCr
•   return (md, md * q23, md * q41, COP * 1.0e+2)
• end

```

KeyError: key :WC not found

```

1. getindex(::Dict{Symbol,Float64}, ::Symbol) @ dict.jl:467
2. top-level scope @ Local: 2 [inlined]
3. top-level scope @ none:0

```

```

• begin
•   A, B, C, D, E, F, G, H = solve(
•       the[:WC],
•       the[:ηC] / 1.0e+2,
•       the[:IC] / 1.0e+2,
•       the[:Tc] + 273.15,
•       the[:Te] + 273.15,
•       FL = "R22"
•   )
•   Markdown.parse(
•       @sprintf """
•   **(a)** A vazão mássica de refrigerante é de **%.4g kg/s**
•
•   **(b)** A taxa de rejeição de calor (no condensador) é de **%.4g kW**
•
•   **(c)** A capacidade de refrigeração é de **%.4g ton** (= %.4g kW)
•
•   **(d)** O COP do refrigerador é de **%.4g%**
•   """ A B C/3.517 C D
•   )
• end

```

## Bibliotecas e Demais Recursos

### Bibliotecas

```

• begin
•   using PlutoUI
•   using PyCall
•   CP = pyimport("CoolProp.CoolProp")
•   using Printf
• end

```