



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)
•   :εc => 50.0: 5.00: 65.0, # Efetividade do condensador (%)
•   :εe => 60.0: 5.00: 75.0, # Efetividade do evaporador (%)
•   :T4 => -5.0: 5.00: 5.0, # 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, %
• );
```

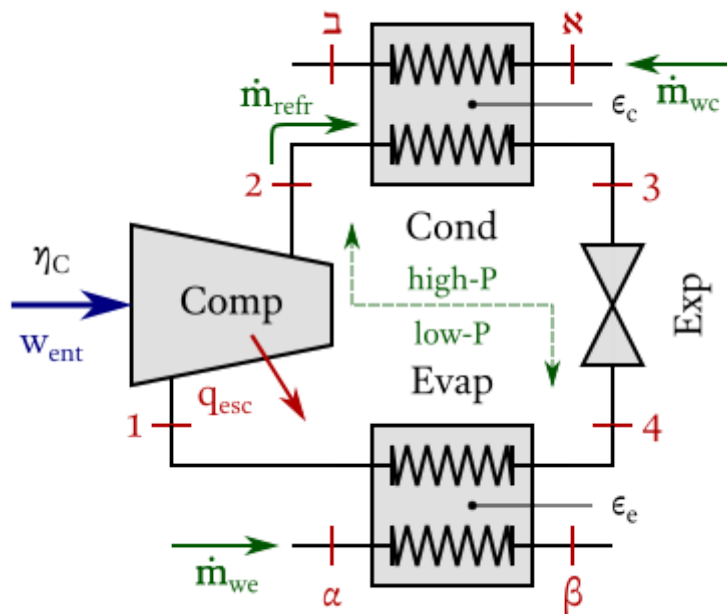
Recompute

```
▼Dict(
  :PR => 4.5
  :Tκ => 10.0
  :IC => 25.0
  :εc => 55.0
  :Tα => 15.0
  :εe => 60.0
  :T4 => 5.0
  :T3 => 60.0
  :ηC => 90.0
)
```

```
• if go == "Recompute"
•   the = Dict{ K => rand(prob[K]) for K in keys(prob) }
• end
```

Enunciado:

Deseja-se obter **4.5 ton** de refrigeração na produção de água gelada—corrente “ α - β ”, na qual água entra em contra-corrente no evaporador a pressão atmosférica e **15.0°C**. Água quente a pressão atmosférica é produzida na contra-corrente “ γ - δ ” do condensador, no qual água entra a **10.0°C**. As efetividades do condensador e evaporador são, respectivamente, de **55.0%** e **60.0%**. O sistema de refrigeração opera com fluido refrigerante **R22**, temperatura de saída da válvula de expansão de **5.0°C** e temperatura de condensação do lado do refrigerante de **60.0°C**, eficiência isentrópica de compressão de **90.0%** com perda de **25.0%** 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 “ β ”, em kg/s
- A temperatura da água produzida em “ β ”, em °C
- A vazão mássica de água quente produzida em “ δ ”, em kg/s
- A temperatura da água produzida em “ δ ”, 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, em kW/K

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 ϵ 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.

```
NTU (generic function with 1 method)
```

```
• NTU(ϵ, δ) = δ ≈ 1.0 ? ϵ/(1-ϵ) : log((1-ϵ*δ)/(1-ϵ))/(1-δ)
```

```
solve (generic function with 1 method)
```

```

function solve(
    PR,          # Potência de refrigeração (kW)
    Tα,          # Temperatura α (K)
    εc,          # Efetividade condensador (norm)
    εe,          # Efetividade evaporador (norm)
    T4,          # Temperatura 4 (K)
    T3,          # Temperatura 3 (K)
    Tκ,          # Temperatura κ (K)
    ηC,          # Eficiência isentrópica compr. (norm)
    IC;          # Saída de irreversibilidade compr. (norm)
    FL="R134a"   # Fluido de trabalho (nome CoolProp)
)
    # Cycle States
    St1 = CP.State(FL, Dict("T" => T4, "Q" => 1))
    St3 = CP.State(FL, Dict("T" => T3, "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
    Q41 = PR            # Energy balance, adiabatic Evap
    ṁ_r = Q41 / (St1.h - St4.h)
    # ---x---
    Qxe = Q41 / εe      # Max Evap Q from definition of ε = ṁwe * cpa * ΔTmax
    Stα = CP.State("water", Dict("T" => Tα, "P" => 101.35))
    cpa = Stα.cp
    ṁwe = Qxe / (cpa * (Tα - T4)) # Ignore solidification for T4 < 0°C
    # ---x---
    h_β = Stα.h - Q41 / ṁwe
    Stβ = CP.State("water", Dict("H" => h_β, "P" => 101.35))
    T_β = Stβ.T
    # ---x---
    Q23 = ṁ_r * (St2.h - St3.h)
    Qxc = Q23 / εc      # Max Cond Q from definition of ε = ṁwc * cpκ * ΔTmax
    Stκ = CP.State("water", Dict("T" => Tκ, "P" => 101.35))
    cpκ = Stκ.cp
    ṁwc = Qxc / (cpκ * (St2.T - Tκ))
    # ---x---
    h_γ = Stκ.h + Q23 / ṁwc
    Stγ = CP.State("water", Dict("H" => h_γ, "P" => 101.35))
    T_γ = Stγ.T
    # ---x---
    q41 = Q41 / ṁ_r
    COP = q41 / wCr
    # ---x---
    Ċrc = ṁ_r * (St2.h - St3.h) / (St2.T - St3.T)
    Ċwc = ṁwc * cpκ
    δc = min(Ċrc, Ċwc) / max(Ċrc, Ċwc)
    Ntc = NTU(εc, δc)
    # ---x---
    # δ=0
    Nte = NTU(εe, 0.0)
    # ---x---
    Ċwe = ṁwe * cpa # Cmin
    UAe = Nte * Ċwe # By the definition of NTU
    return (ṁ_r, ṁwe, T_β, ṁwc, T_γ, COP, Ntc, Nte, UAe)
end

```

(a) A vazão mássica de refrigerante, em kg/s, é de **0.1225**

(b) A vazão mássica de água gelada produzida em “β”, em kg/s, é de **0.6298**

- (c) A temperatura da água produzida em “β”, em °C, é de **9.006**
- (d) A vazão mássica de água quente produzida em “␣”, em kg/s, é de **0.1177**
- (e) A temperatura da água produzida em “␣”, em °C, é de **51.81**
- (f) O COP do refrigerador, em %, é de **324.3**
- (g) O número de unidades de transferência, NTU, do condensador, é de **1.003**
- (h) O número de unidades de transferência, NTU, do evaporador, é de **0.9163**
- (i) O coeficiente global de transferência de calor, UA, do evaporador, em kW/K, é de **2.417**.

```

begin
    A, B, C, D, E, F, G, H, I = solve(
        the[:PR] * 3.517,      # ton --> kW
        the[:Tα] + 273.15,    # °C --> K
        the[:εc] / 1.0e+2,    # % --> norm
        the[:εe] / 1.0e+2,    # % --> norm
        the[:T4] + 273.15,    # °C --> K
        the[:T3] + 273.15,    # °C --> K
        the[:Tκ] + 273.15,    # °C --> K
        the[:ηC] / 1.0e+2,    # % --> norm
        the[:IC] / 1.0e+2,    # % --> norm
        FL = "R22"
    )
    Markdown.parse(
        @sprintf """
        **(a)** A vazão mássica de refrigerante, em kg/s, é de **%.4g**
        **(b)** A vazão mássica de água gelada produzida em “β”, em kg/s, é de **%.4g**
        **(c)** A temperatura da água produzida em “β”, em °C, é de **%.4g**
        **(d)** A vazão mássica de água quente produzida em “␣”, em kg/s, é de **%.4g**
        **(e)** A temperatura da água produzida em “␣”, em °C, é de **%.4g**
        **(f)** O COP do refrigerador, em %, é de **%.4g**
        **(g)** O número de unidades de transferência, NTU, do condensador, é de **%.4g**
        **(h)** O número de unidades de transferência, NTU, do evaporador, é de **%.4g**
        **(i)** O coeficiente global de transferência de calor, UA, do evaporador, em kW/K,
        é de **%.4g**.
        """ A B C-273.15 D E-273.15 F*1.0e+2 G H I
    )
end

```

Bibliotecas e Demais Recursos

Bibliotecas

```

begin
    using PlutoUI

```

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