

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



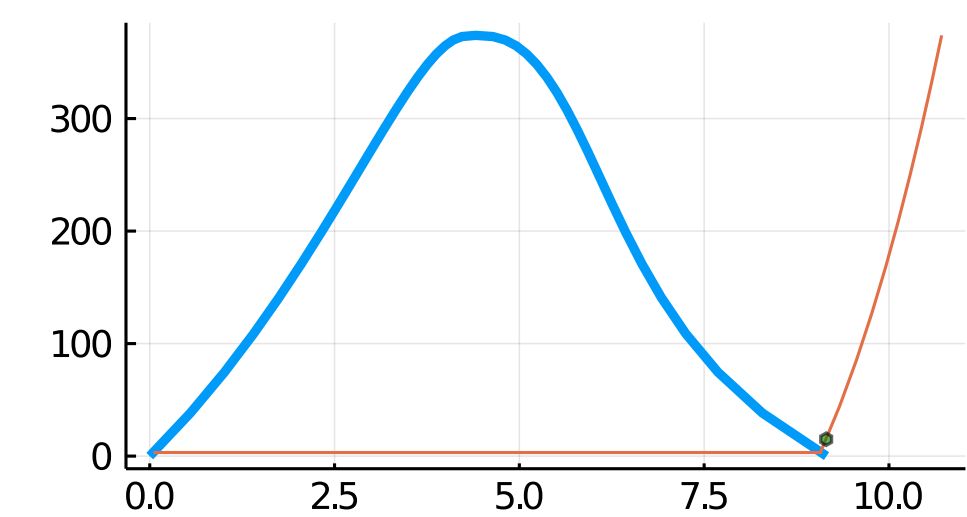
A08 – Misturas Gás-Vapor e Condicionamento de Ar

01 – Ar Seco e Atmosférico e Medidas de Umidade

Exemplo A0801-01 – Massa de vapor d'água em um galpão fechado

Adaptado de: Exemplo 14-1 (ÇENGEL, Y. A., 7a Ed., 2013, pp. 734-735)

► Dict(:T => 15.0:1.0:60.0, :P => 80.0:5.0:120.0, :φ => 15.0:5.0:100.0)



Live docs

Temperatura T, em °C

15.0

Pressão P, em kPa

100.0

Umidade relativa φ, em %

45.0

Enunciado:

Um galpão de 11.5m × 11.5m × 6.0m contém ar a 15.0°C e 100.0kPa a uma umidade relativa de 45.0%. Determine: **(a)** a pressão parcial do ar seco; **(b)** a umidade específica (absoluta); **(c)** a entalpia por unidade de massa de ar seco; **(d)** as massas de ar seco e de vapor d'água no galpão.

Solução:

Letra	Propriedade	Valor	Unid.
(a)	P_a	99.23	kPa
(b)	ω	0.0048	kg/kg
(c)	h	27.24	kJ/kg
(d)	m_a	951.97	kg
(d)	m_v	4.58	kg

Resolução

(a) pressão parcial do ar seco

A pressão parcial do ar seco é determinada via

$$P_a = P - P_v,$$

onde

$$P_v = \phi P_g = \phi P_{sat@T},$$

e $P_{sat@T}$ é obtida por meio da biblioteca **CoolProp** via **Pycall.jl**.

```
P_g, P_v, P_a = (1.706, 0.768, 99.232) kPa
begin
    P_g = CP.PropsSI("P", "T", the_T + 273.15, "Q", 1.0, "water") * 1.0e-3 # kPa
    P_v = P_g * the_φ / 100.0 # kPa
    P_a = the_P - P_v # kPa
    md""" $P_g$, $P_v$, $P_a$ = (
        $(@sprintf("%.3f", P_g)),
        $(@sprintf("%.3f", P_v)),
        $(@sprintf("%.3f", P_a))) kPa
    """
end
```

(b) umidade específica do ar

A umidade específica (absoluta) é determinada via

$$\omega = \frac{0.622 P_v}{P - P_v} = \frac{0.622 P_v}{P_a},$$

com P_v e P_a obtidos no item anterior. Note que a constante psicrométrica

$$C_\psi = \frac{R_a}{R_v} \approx 0.621967$$

obtida por meio da biblioteca **CoolProp** via **Pycall.jl** é utilizada nos cálculos ao invés do valor arredondado 0,622.

```
ω = 0.0048 kg/kg
begin
    ω = Cψ * P_v / P_a # kg/kg
    md""" $ω$ =
        $(@sprintf("%.4f", ω)) kg/kg
    """
end
```

(c) entalpia do ar por unidade de massa de ar seco

A entalpia do ar por unidade de massa de ar seco é determinada via

$$h = h_a + \omega h_v \approx c_p T + \omega h_g,$$

Na qual T é a temperatura em °C.

```
h_a, h_v, h_g, h = (15.07, 2528.33, 2528.33, (27.24, 27.24)) kJ/kg
begin
    cp = 1.005 # kJ/kg·°C
    minP = CP.PropsSI("PMIN", "water") * 1.00001e-3 # kPa
    ha = cp * the_T # kJ/kg
    stg = CP.State("water", Dict("Q"=>1.0, "T"=>the_T+273.15))
    if the_φ < 1
        stv = CP.State("water", Dict("P"=>(Pv<minP ? minP : Pv), "T"=>the_T+273.15))
    else
        stv = stg
    end
    hv = stv.h
    hg = stg.h
    h = (ha + ω*hv, ha + ω*hg)
    md""" $h_a$, $h_v$, $h_g$, $h$ = (
        $(@sprintf("%.2f", ha)), $(@sprintf("%.2f", hv)), $(@sprintf("%.2f", hg))),
        $(@sprintf("%.2f", h[1])), $(@sprintf("%.2f", h[2])) ) kJ/kg"""
    end
```

(d) massas de ar seco e de vapor no galpão

Tais massas podem ser calculadas pela equação de estado (de gás ideal):

$$m = \frac{PV}{RT}$$

```
m_a, m_v, ω * m_a = (951.975, 4.58014, 4.58014) kg
begin
    V = the_x * the_y * the_z
    T = the_T + 273.15
    ma = Pa * V / (Ra * T)
    mv = Pv * V / (Rv * T)
    md""" $m_a$, $m_v$, $ω$ = (
        $(@sprintf("%.3f", ma)),
        $(@sprintf("%.5f", mv)),
        $(@sprintf("%.5f", ω*ma))) kg"""
    end
```

Bibliotecas e Demais Recursos

Bibliotecas

```
begin
    using PlutoUI
    using PyCall
    using Printf
    using Plots
end
```

Constantes

```
Cψ = 0.621967
begin
    CP = pyimport("CoolProp.CoolProp")
    Ra = CP.PropsSI("GAS_CONSTANT", "air") / CP.PropsSI("M", "air") * 1.0e-3
    Rv = CP.PropsSI("GAS_CONSTANT", "water") / CP.PropsSI("M", "water") * 1.0e-3
    Cψ = Ra/Rv # Psychrometric constant (usually rounded to 0.622)
    md"Cψ = $(@sprintf("%.6f", Cψ))"
end
```

Gráfico - Domo de Saturação da Água

```
begin
    # Basic constants
    FL = "water"
    TT = CP.PropsSI("Ttriple", FL)
    TC = CP.PropsSI("Tcrit", FL)
    TSTL = (TT, CP.PropsSI("S", "T", TT, "Q", 0.0, FL))
    TSTV = (TT, CP.PropsSI("S", "T", TT, "Q", 1.0, FL))
    TSCR = (TC, CP.PropsSI("S", "T", TC, "Q", 0.5, FL))
    # Saturation dome
    N = range(0.0, stop=1.0, length=20)
    P = 1.0 ./ (1.0 .- N).^2
    tLiq = TT .* (TC - TT) .* P
    tVap = reverse(tLiq)[2:end]
    sLiq = [CP.PropsSI("S", "T", _t, "Q", 0.0, FL) for _t in tLiq] .* 1.0e-3
    sVap = [CP.PropsSI("S", "T", _t, "Q", 1.0, FL) for _t in tVap] .* 1.0e-3
    DOME = (cat(tLiq, tVap, dims=1).-273.15, cat(sLiq, sVap, dims=1))
end;
```