HW-01

February 8, 2024

1 Homework Assignment 2

Complete this notebook, export to pdf (`File>Download as> PDF (via LaTeX)`) and submit the pdf to BB

1.1 Pb 1: Air thermodynamic properties class

The concept of class is central to most libraries. It consists of the definition of an object, which has specific properties, data and function.

Zografos et al. (*Compt. Methods Appl. Mech Eng.*, 61:177-187, 1987) proposed polynomials to compute density, dynamic viscosity, thermal conductivity, and specific heat at constant temperature of air at atmospheric pressure as a function of the static temperature T on the Kelvin scale as follows: * Density ρ in units of kg/m^3:

$$\rho = \begin{cases} -2.440\times10^{-2}T+5.9958\,, & 100\,\mathrm{K}\leq T<150\,\mathrm{K}\\ 345.57(T-2.6884)^{-1}\,, & 150\,\mathrm{K}\leq T\leq 3000\,\mathrm{K}\end{cases}$$

* Dynamic viscosity μ in units of N.s/m^2 = kg/(m.s) = Pa.s:

$$\mu = 4.1130\times10^{-6}+5.0523\times10^{-8}T-1.4346\times10^{-11}T^2+2.5914\times10^{-15}T^3$$

- Thermal conductivity k in units of W/(m.K):

$$k=-7.488\times10^{-3}+1.7082\times10^{-4}T-2.3758\times10^{-10}T^2+2.2012\times10^{-10}T^3+9.4600\times10^{-14}T^4+1.5797\times10^{-17}T^5$$

- Specific heat at constant pressure c_p in unites of J/(kg.K):

$$c_p = 1061.332-0.432819T+1.02344\times10^{-3}T^2-6.47474\times10^{-7}T^3+1.3864\times10^{-10}T^4$$

Air density ρ of units kg/m^3 can be computed from

The Prandtl number Pr can be computed as

$$Pr=\frac{\mu c_p}{k}$$

and the thermal conductivity α of units m^2/s as

$$\alpha = \frac{k}{\rho c_p}$$

Write a class storing all these variables for input temperature (float) and unit used for temperature (string, it can only be “K”, “C”, “F”). The class should convert the temperature to Kelvin if unit is either “C” or “F”). The class should give a warning if * the temperature is outside of the range, * the unit is not “K”, “C”, “F”, e.g. “c”

1.1.1 Tasks

- Complete the manual part of the class (text between “ “ ” xxxx “ “ “), i.e. write appropriate units.
- Complete the class to compute all quantities defined in the manual.
- Apply your class to the examples below.
- Define units for examples.
- Plot density as function of temperature in the range $10^2 \leq T < 3000$. Use log-scale for temperature. Axes must be labeled with their variables and units.

[135]:

```python
import numpy as np
import math
import matplotlib.pyplot as plt
class Air:

    """
    Returns an object containing the thermodynamic properties of air at atmospheric pressure
    Input:
            T: temperature (float)
            Tunit: unit of temperature (string). Can only be "K", "C", "F"
    output:
            self.rho: density [kg/m^3]
            self.mu: dynamic viscosity [Pa-s]
            self.k: thermal conductivity [W/m-K]
            self.cp: Specific heat at constant pressure [J*kg^-1*K^-1]
            self.Pr: Prandtl number
            self.nu: kinematic viscosity [m^2/s]
            self.alpha: thermal diffusivity [m^2/s]
    """
    def __init__(self,T,Tunit):
        if Tunit == "C":
            T += 273.15
        elif Tunit == "F":
            T = (T - 32)*5/9 +273.15
        elif Tunit != "K":
            print("Wrong input for unit of temperature")
            print("Input should be 'K', 'C', or 'F'")
        self.T = T
        k = 0
        mu = 0
        cp = 0
        rho = 0

        if 100 <= T < 150:
            self.rho = -2.440E-2 * T + 5.9958
        elif 150 <= T <= 3000:
            self.rho = 345.57 * (T - 2.6884) ** -1
        else:
            return print("ERROR: Temperature input is outside the range of 100K␣
↪to 3000K")

        self.mu = 4.1130E-6 + 5.0523E-8 * T - 1.4346E-11*T**2 + 2.5914E-15*T**3
        self.k = -7.488E-3 + 1.7082E-4 * T - 2.3758E-10 * T ** 2 + 2.2012E-10 *␣
↪T ** 3 + 9.46E-14 * T ** 4 + 1.5797E-17 * T ** 5
        self.cp = 1061.332 - 0.432819 * T + 1.02344E-3 * T ** 2 - 6.47474E-7 *␣
↪T ** 3 + 1.3864E-10 * T ** 4
        k = self.k
        mu = self.mu
        cp = self.cp
        rho = self.rho
        self.pr = (mu * cp) / k
        self.nu = mu / rho
        self.alpha = k / (rho * cp)
```

[151]:

```
myair = Air(32,"F")
```

[146]:

```
myair = Air(40,"F")
print("Temperature in K: %.2f" %myair.T)
print("Kinematic viscosity: %.2e m^2/s, Thermal conductivity: %.4f W/m-K"␣
↪%(myair.nu,myair.k))
```

```
Temperature in K: 277.59
Kinematic viscosity: 1.36e-05 m^2/s, Thermal conductivity: 0.0452 W/m-K
```

[138]:

```
myair = Air(-200,"C")
```

```
ERROR: Temperature input is outside the range of 100K to 3000K
```

[139]:

```
myair = Air(2000,"k")
```

```
Wrong input for unit of temperature
Input should be 'K', 'C', or 'F'
```

[140]:

```
myair = Air(2000,"K")
```

```
print("Dynamic viscosity: %.2e Pa-s, Thermal diffusivity: %.2e m^2/s" %(myair.
 ↪mu,myair.alpha))
```

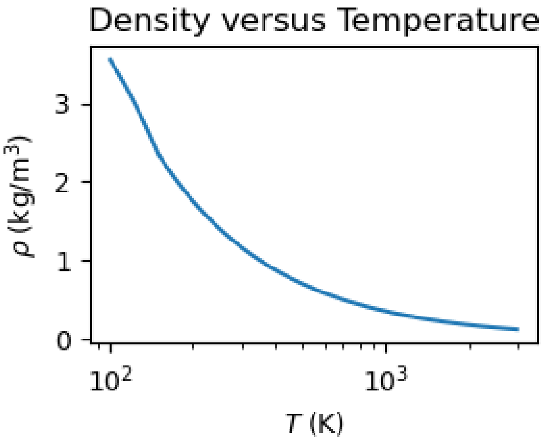
```
Dynamic viscosity: 6.85e-05 Pa-s, Thermal diffusivity: 1.79e-02 m^2/s
```

[142]:

```
T = np.logspace(2,np.log10(2999.),1000)
rho = np.zeros_like(T)
for i in range(T.shape[0]):
    # Your code here
    myair = Air(T[i], 'K')
    rho[i] = myair.rho
```

[152]:

```
# your line plot here
plt.figure(figsize =(3, 2), dpi=100)
plt.plot(T,rho)
plt.xscale('log')
plt.xlabel("$T\;(\mathrm{K})$")
plt.ylabel(r"$\rho\;(\mathrm{kg/m^3})$")
plt.title("Density versus Temperature")
plt.show()
```



1.2 Pb 2 Window

Set up equations and apply realistic numerical values to them to discuss heat losses of a single pane window, a single pane window with storm window and a double paned window with air trapped at a vacuum of 10^{-3}torr ina gap of 5mm. Do not consider the effects of radiation for any of the window.

Assumptions:

- Heat transfer is steady and 1D
- Radiation is ignored

- Convection in the gap of the storm window is neglected
- The temperature on the inside and outside surfaces of any window are isothermal and equal to $T_{si} = 10^\circ\mathrm{C}$ and $T_{so} = -10^\circ\mathrm{C}$, respectively.
- For the storm window, and the double pane window, the average temperature in the gap between the two glass pane is assumed to the average of T_{si} and T_{so} or $T_{mean} = 0^\circ\mathrm{C}$.

Setup In this cell you are expected to: * Insert your sketches of the single pane and double pane windows showing heat fluxes, temperatures at boundaries and dimensions * Store in folder `MyFigures`. Change the names `your-single-pane-sketch` and `your-double-pane-sketch` in the HTML command below to include your figures to this cell. * Complete a **worded** write-up of the solutions (for both windows) including equations written in LaTeX

This is the type of introduction of your work that is expected in this class for any deliverable

The goal is to compute the heat loss from a model window under controlled conditions. We choose to fix the inside surface temperature to $10^\circ\mathrm{C}$ and the outside surface temperature to $-10^\circ\mathrm{C}$ for a window of height and width $h = 1.2$ m and $w = 0.7$ m with a surface aera $A = h \times w$. The thickness of a residential window pane is $t_{glass} = 3/32\,\mathrm{in}$ (https://ringerwindows.com/thick-glass-windows/) with a thermal conductivity $k = 0.96$ W/m.K. The storm window has a 2.5 in.

The heat rate for a single window pane is... *Your derivation of the solution here*

[ ]:

```
import numpy as np
from Libraries import thermodynamics as thermo
T_si = 10. #C
T_so = -10. #C
h = 1.2 #m
w = 0.7 #m
t_glass = 3./32.*0.0254 #m
k_glass = 0.96
A = h*w
t_gap_storm = 2*0.0254 #m
# getting thermodynamic properties of air for the storm window
air_gap_storm = thermo.Fluid('air',thermo.C2K(0.))
t_gap_vacuum = 0.005#m
k_vacuum = xxxx #W/m.K (from graph)
R_glass =
R_gap_storm =
R_gap_vacuum =
R_total_single =
R_total_storm =
R_total_double =

print("Thermal conductivity of air at 0 C: %.4f" %air_gap_storm.k)
print("Thermal resistance for the single window: %.4f K/W" %R_total_single)
print("Thermal resistance for the storm window: %.4f K/W" %R_total_storm)
print("Thermal resistance for the double pane window: %.4f K/W" %R_total_double)

DT = T_si - T_so
```

```
q_single = DT/R_total_single
q_storm = DT/R_total_storm
q_double = DT/R_total_double
print("Single window pane q: %.0f W" %q_single)
print("Storm window q: %.0f W" %q_storm)
print("Double pane window q: %.0f W" %q_double)
```

[ ]:

```

```