Maquinas Térmicas - Tarefa 1

Carlos Adir Ely Murussi Leite
150121059

24 de Setembro de 2022

1 Enunciado

Temos dois estados termodinâmicos (1) e (2) dados por pressão e volume conhecidos

$$P_1 = 150\,000 \text{ Pa} \qquad v_1 = 0.675 \, \frac{\text{m}^3}{\text{kg}}$$

$$P_2 = 40\,000 \text{ Pa} \qquad v_2 = 1.9 \, \frac{\text{m}^3}{\text{kg}}$$

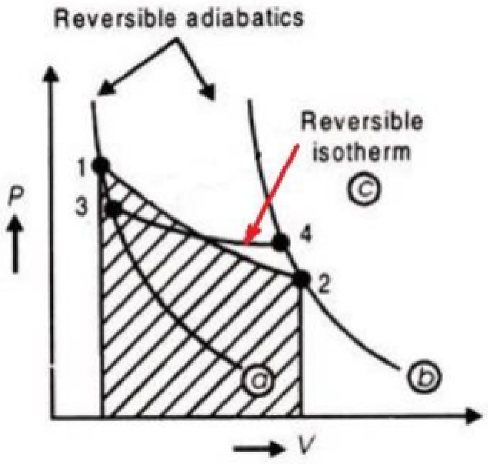

Figure 1: Diagrama inicial

2 Solução

Foi criado um código no EES (*Engineering Equation Solver* que se encontra abaixo para fazer os cálculos e obter os 6 pontos principais. Os valores obtidos são dados pela Tabela (1)

Table 1: Tabela com valores obtidos pelo EES

Estado	Pressão (Pa)	Volume (m^3/kg)	Temperatura (K)	Entropia J/kg · K	Energia Interna J/kg
1	150000	0.675	352.8	5752.13	252265
2	40000	1.900	264.8	5843.01	189072
3	94082	0.942	308.8	5752.13	220668
4	68548	1.293	308.8	5843.01	220668
5	54900	1.384	264.8	5752.13	189072
6	109290	0.926	352.8	5843.01	252265

Código EES

```
"Estado termodinamico 1"
P[1] = 150000  [Pa]
v[1] = 0.675 [m^3/kg]
T[1] = Temperature(Air, P=P[1], V=v[1])
u[1] = IntEnergy(Air, T=T[1])
s[1] = Entropy(Air, P=P[1], v=v[1])
n[1] = Cp(Air, T=T[1])/Cv(Air, T=T[1])

"Estado termodinamico 2"
P[2] = 40000 [Pa]
v[2] = 1.9 [m^3/kg]
T[2] = Temperature(Air,P=P[2], V=v[2])
```

```
u[2] = IntEnergy(Air,T=T[2])
s[2] = Entropy(Air, P=P[2], v=v[2])
n[2] = Cp(Air, T=T[2])/Cv(Air, T=T[2])

"Estado termodinamico 3"
s[3] = s[1]
u[3] = IntEnergy(Air, T=T[3])
P[3] = Pressure(Air, T=T[3], s=s[3])
v[3] = Volume(Air, T=T[3], s=s[3])
n[3] = Cp(Air, T=T[3])/Cv(Air, T=T[3])

"Estado termodinamico 4"
s[4] = s[2]
u[4] = IntEnergy(Air, T=T[4])
P[4] = Pressure(Air, T=T[4], s=s[4])
v[4] = Volume(Air, T=T[4], s=s[4])
n[4] = Cp(Air, T=T[4])/Cv(Air, T=T[4])

"Estado termodinamico 5"
s[5] = s[1]
T[5] = T[2]
u[5] = IntEnergy(Air, T=T[5])
P[5] = Pressure(Air, T=T[5], s=s[5])
v[5] = Volume(Air, T=T[5], s=s[5])
n[5] = Cp(Air, T=T[5])/Cv(Air, T=T[5])

"Estado termodinamico 6"
s[6] = s[2]
T[6] = T[1]
u[6] = IntEnergy(Air, T=T[6])
P[6] = Pressure(Air, T=T[6], s=s[6])
v[6]= Volume(Air, T=T[6], s=s[6])
n[6] = Cp(Air, T=T[6])/Cv(Air, T=T[6])

"Condicao de equilibro"
W_1_to_3 = u[1] - u[3]
W_4_to_2 = u[4] - u[2]
T[3] = T[4]
W_1_to_3 = W_4_to_2
```

3 Gráficos

Os gráficos foram feitos com o Python com a biblioteca CoolProp e os valores obtidos pelo EES

Código Python

```
import numpy as np
import os
from CoolProp.CoolProp import PropsSI
from matplotlib import pyplot as plt

fluido = "Air"
Ps = [15e+4,4e+4,94082.090068594,68548.462880174,54899.605981277,109290.401866385]  # Pa
vs = [0.675,1.9,0.94222389,1.29319300,1.38434509,0.92643085]  # m^3/kg
ds = [1/v for v in vs]  # kg/m^3
Ts = [PropsSI("T", "P", P, "D", d, fluido) for P, d in zip(Ps, ds)]
ss = [PropsSI("S", "P", P, "D", d, fluido) for P, d in zip(Ps, ds)]
us = [PropsSI("U", "P", P, "D", d, fluido) for P, d in zip(Ps, ds)]
print("Ts = ", Ts)
print("ss = ", ss)
print("us = ", us)

npts = 11
Psample = np.linspace(min(Ps), max(Ps), npts)
vsample = np.linspace(min(vs), max(vs), npts)
ssample = np.linspace(min(ss), max(ss), npts)
Tsample = np.linspace(min(Ts), max(Ts), npts)
pairsdotted = [(0, 5), (4, 1), (5, 3), (2, 4)]
pairssolid = [(0, 2), (2, 3), (3, 1)]

dv = max(vs)-min(vs)
dP = max(Ps)-min(Ps)
dT = max(Ts)-min(Ts)
ds = max(ss)-min(ss)
fig1 = plt.figure()
ax1 = plt.gca()
fig2 = plt.figure()
ax2 = plt.gca()
for i in range(6):
    ax1.scatter(vs[i], Ps[i], label="%d" % (i+1))
```

```
    ax2.scatter(ss[i], Ts[i], label="%d" % (i+1))
for a, b in pairsdotted:
    ssample = np.linspace(ss[a], ss[b], npts)
    Tsample = np.linspace(Ts[a], Ts[b], npts)
    Psample = [PropsSI("P", "T", T, "S", s, fluido) for s, T in zip(ssample, Tsample)]
    dsample = [PropsSI("D", "T", T, "S", s, fluido) for s, T in zip(ssample, Tsample)]
    vsample = [1/d for d in dsample]
    ax1.plot(vsample, Psample, color="k", ls="dotted")
    ax2.plot(ssample, Tsample, color="k", ls="dotted")
for a, b in pairssolid:
    ssample = np.linspace(ss[a], ss[b], npts)
    Tsample = np.linspace(Ts[a], Ts[b], npts)
    Psample = [PropsSI("P", "T", T, "S", s, fluido) for s, T in zip(ssample, Tsample)]
    dsample = [PropsSI("D", "T", T, "S", s, fluido) for s, T in zip(ssample, Tsample)]
    vsample = [1/d for d in dsample]
    ax1.plot(vsample, Psample, color="k", ls="solid")
    ax2.plot(ssample, Tsample, color="k", ls="solid")
ax1.set_xlim(min(vs)-0.1*dv, max(vs)+0.1*dv)
ax1.set_ylim(min(Ps)-0.1*dP, max(Ps)+0.1*dP)
ax1.set_xlabel(r"Volume espec fico $v \ (m^3/kg)$")
ax1.set_ylabel(r"Press o $P \ (Pa)$")
ax1.legend()
ax2.set_xlim(min(ss)-0.1*ds, max(ss)+0.1*ds)
ax2.set_ylim(min(Ts)-0.1*dT, max(Ts)+0.1*dT)
ax2.set_xlabel(r"Entropia espec fico $s \ (J/kg\cdot K)$")
ax2.set_ylabel(r"Temperatura $T \ (K)$")
ax2.legend()

plt.show()
```

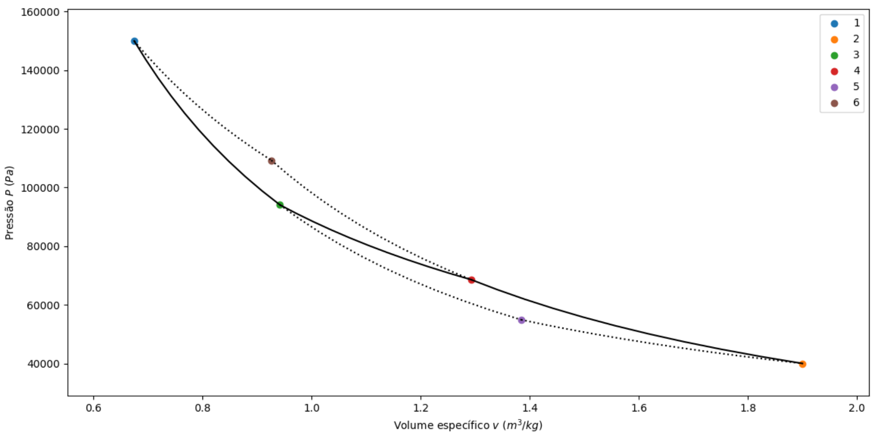

Figure 2: Diagrama Pressão-Volume específico

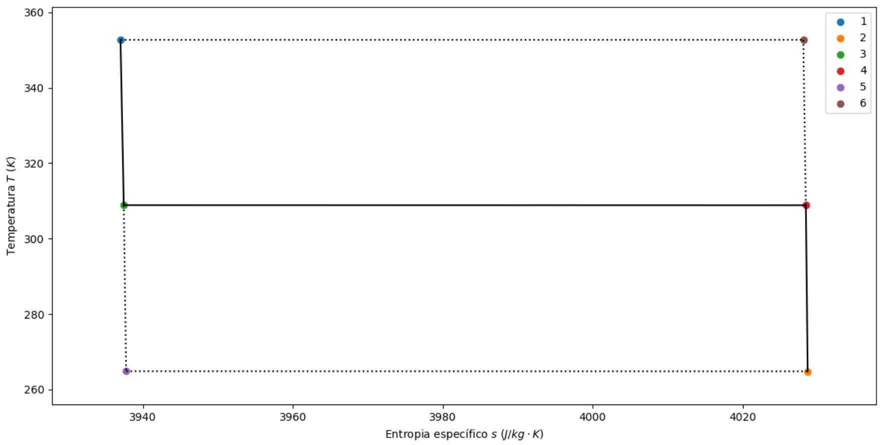

Figure 3: Diagrama Temperatura-Entropia