

Correction TD Cycle REP Palier N4

April 20, 2021

1 Correction TD : cycle thermodynamique du palier N4

On va utiliser la bibliothèque `coolprop`

```
[1]: pip install CoolProp
```

```
Requirement already satisfied: CoolProp in  
/home/florent/anaconda3/lib/python3.8/site-packages (6.4.1)  
Note: you may need to restart the kernel to use updated packages.
```

```
[2]: import CoolProp.CoolProp as CP  
fluid = 'Water'
```

1.1 Cycle de Carnot (2 pts)

Le rendement d'un cycle de Carnot qui fonctionne entre les températures T_c et T_f est:

$$\eta = 1 - \frac{T_f}{T_c}.$$

Attention, les températures doivent être en K .

On cherche la température d'ébullition de l'eau à 72 bars.

```
[3]: Tc = CP.PropsSI('T','P',72e5,'Q',1,fluid) #K  
Tf = 273.15+30 #K  
eta = 1-Tf/Tc  
print("Le rendement du cycle de Carnot équivalent est eta={:.2%}".format(eta))
```

Le rendement du cycle de Carnot équivalent est eta=45.95%

1.2 Cycle de Rankine avec machines idéales (7 pts)

```
[4]: T1 = Tf  
P1 = CP.PropsSI('P','T',T1,'Q',0,fluid) #liquide saturé  
H1 = CP.PropsSI('H','T',T1,'Q',0,fluid)  
S1 = CP.PropsSI('S','T',T1,'Q',0,fluid)
```

```
[5]: P3 = 72e5 #Pa  
T3 = CP.PropsSI('T','P',P3,'Q',1,fluid) #vapeur saturée  
H3 = CP.PropsSI('H','P',P3,'Q',1,fluid)
```

```
S3 = CP.PropsSI('S','P',P3,'Q',1,fluid)
```

```
[6]: print("La pression en 1 vaut P1=%.3f bars" %(P1/1e5))
print("L'\enthalpie massique en 1 vaut H1=%.3f kJ/kg" %(H1/1e3))
print("L'\entropie massique en 1 vaut S1=%.3f kJ/kg" %(S1/1e3))
print("La température en 3 vaut T3=%.1f °C" %(T3-273.15))
print("L'\enthalpie massique en 3 vaut H3=%.3f kJ/kg" %(H3/1e3))
print("L'\entropie massique en 3 vaut S3=%.3f kJ/kg" %(S3/1e3))
```

La pression en 1 vaut $P_1=0.042$ bars
L'enthalpie massique en 1 vaut $H_1=125.734$ kJ/kg
L'entropie massique en 1 vaut $S_1=0.437$ kJ/kg
La température en 3 vaut $T_3=287.7$ °C
L'enthalpie massique en 3 vaut $H_3=2769.991$ kJ/kg
L'entropie massique en 3 vaut $S_3=5.800$ kJ/kg

```
[7]: P2=P3
P4=P1
```

Transformation 1 → 2: compression *isentropique*.

On cherche l'enthalpie massique en 2 h_{2s} telle que $P_2 = 72$ bars et $s_2 = s_1$

Le travail de compression par unité de masse est $w_{1,2s} = h_{2s} - h_1$

```
[8]: S2 = S1
H2s = CP.PropsSI('H','P',P2,'S',S2,fluid)
w12s = H2s - H1
print("Le travail de compression massique vaut %.3f kJ/kg" %(w12s/1e3))
```

Le travail de compression massique vaut 7.216 kJ/kg

Transformation 2 → 3: apport de chaleur.

L'apport de chaleur par unité de masse est $q_{2s,3} = h_3 - h_{2s}$

```
[9]: q2s3 = H3 - H2s
print("L'\apport de chaleur massique vaut %.3f kJ/kg" %(q2s3/1e3))
```

L'apport de chaleur massique vaut 2637.041 kJ/kg

Transformation 3 → 4: détente *isentropique*.

On cherche l'enthalpie massique en 4 h_{4s} telle que $P_4 = 0.042$ bars et $s_4 = s_3$

Le travail de détente par unité de masse est $w_{3,4s} = h_{4s} - h_3$

```
[10]: S4 = S3
H4s = CP.PropsSI('H','P',P4,'S',S4,fluid)
w34s = H4s - H3
print("Le travail de détente massique vaut %.3f kJ/kg" %(w34s/1e3))
X4s = CP.PropsSI('Q','P',P4,'S',S4,fluid)
```

```
print("Le titre en fin de détente isentropique vaut x4s={:.1%}".format(X4s))
q4s1 = H1 - H4s
print("La chaleur massique cédée à la source froide vaut %.3f kJ/kg" %(q4s1/
↪1e3))
```

Le travail de détente massique vaut -1018.264 kJ/kg
 Le titre en fin de détente isentropique vaut x4s=66.9%
 La chaleur massique cédée à la source froide vaut -1625.993 kJ/kg

La définition du rendement est :

$$\eta = \frac{|w_{3,4s} + w_{1,2s}|}{q_{2s,3}}$$

```
[11]: etarankines=abs(w34s+w12s)/q2s3
print("Le rendement du cycle de Rankine isentropique vaut etas={:.1%}".
↪format(etarankines))
```

Le rendement du cycle de Rankine isentropique vaut etas=38.3%

1.3 Cycle de Rankine avec rendement des machines (3 pts)

Pour une pompe, la définition du rendement isentropique est le rapport entre le travail de compression isentropique et le travail de compression réel, c'est-à-dire:

$$\eta_{pump} = \frac{w_s}{w} = \frac{h_{2s} - h_1}{h_2 - h_1}$$

On raisonne donc sur le Δh isentropique auquel on applique le rendement pour trouver le Δh réel et donc la valeur de l'enthalpie massique au point 2

```
[12]: etap=0.9
w12=w12s/etap
H2 = H1 + w12
print("Le travail de compression massique vaut %.3f kJ/kg" %(w12/1e3))
```

Le travail de compression massique vaut 8.018 kJ/kg

Pour une turbine, la définition du rendement isentropique est le rapport entre le travail de détente réel et le travail de détente isentropique, c'est-à-dire:

$$\eta_{turbine} = \frac{w}{w_s} = \frac{h_4 - h_3}{h_{4s} - h_3}$$

On raisonne donc sur le Δh isentropique auquel on applique le rendement pour trouver le Δh réel et donc la valeur de l'enthalpie massique au point 4

```
[13]: etat=0.8
w34=w34s*etat
H4 = H3 + w34
print("Le travail de détente massique vaut %.3f kJ/kg" %(w34/1e3))
X4 = CP.PropsSI('Q','P',P4,'H',H4,fluid)
print("Le titre en fin de détente réelle vaut x4={:.1%}".format(X4))
q23 = H3 - H2
```

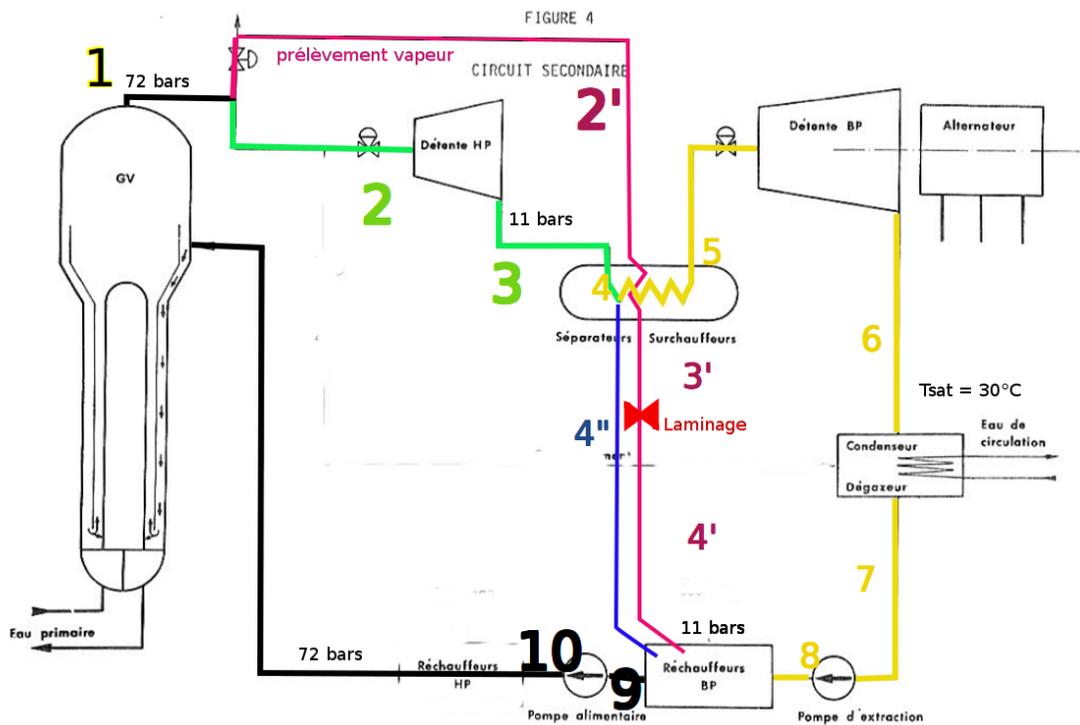
```

etarankine=abs(w34+w12)/q23
print("Le rendement du cycle de Rankine réel vaut eta={:.1%}".
      →format(etarankine))

```

Le travail de détente massique vaut -814.611 kJ/kg
 Le titre en fin de détente réelle vaut $x_4=75.3\%$
 Le rendement du cycle de Rankine réel vaut $\eta=30.6\%$

1.4 Cycle avec détente en deux phases, machines idéales (5 pts)



```

[14]: P1 = 72e5
P2 = P1
P2p = P1
P3p = P1
P10 = P1
P3 = 11e5
P4 = P3
P4p = P3
P4pp = P3
P5 = P3
P8 = P3
P9 = P3
T7 = 30+273.15
P7 = CP.PropsSI('P','T',T7,'Q',0,fluid) #liquide saturé

```

```
P6 = P7
```

```
[15]: M1 = 1 #En noir
M2 = 0.95 #En vert
M2p = M1-M2 #En rose
M3 = M2 #En vert
M3p = M2p #En rose
M4p = M2p #En rose
M9 = M1 #En noir
M10 = M1 #En noir
```

Première détente, comme pour le Rankine, transformation 2 → 3s isentropique

```
[16]: H1 = CP.PropsSI('H','P',P1,'Q',1,fluid)
S1 = CP.PropsSI('S','P',P1,'Q',1,fluid)
H2 = H1
H2p = H2
S3s = S1
H3s = CP.PropsSI('H','P',P3,'S',S3s,fluid)
X3s = CP.PropsSI('Q','P',P3,'S',S3s,fluid)
```

Séparateur: on recueille sur la branche principale la vapeur, sur la branche " le liquide

```
[17]: H4 = CP.PropsSI('H','P',P4,'Q',1,fluid)
M4s = M3*X3s #En jaune
H4pp = CP.PropsSI('H','P',P4,'Q',0,fluid)
M4pps = M3*(1-X3s) #En bleu
```

Surchauffeur: on suppose que toute la chaleur cédée par la vapeur vive prélevée au GV est cédée à la branche principale. C'est-à-dire que:

$$m_{4,5} (h_5 - h_4) + m_{2',3'} (h_{3'} - h_{2'})$$

Il faut fixer une des conditions: on fixe $h_{3'}$ telle qu'on ait le liquide saturé.

```
[18]: H3p = CP.PropsSI('H','P',P3p,'Q',0,fluid) #condition sortie surchauffeur imposée
H5s = (-M3p * (H3p-H2p)/M4s + H4) #surchauffe
```

```
[19]: H4p = H3p #laminage isenthalpe
S5s = CP.PropsSI('S','P',P5,'H',H5s,fluid)
M5s = M4s #En jaune
S6s = S5s #détente isentropique
M6s = M5s #En jaune
H6s = CP.PropsSI('H','P',P6,'S',S6s,fluid)
X6s = CP.PropsSI('Q','P',P6,'S',S6s,fluid)
H7 = CP.PropsSI('H','P',P7,'Q',0,fluid) #condenseur
M7s = M6s #En jaune
S7 = CP.PropsSI('S','P',P7,'Q',0,fluid)
S8s = S7 #compression isentropique
```

```
H8s = CP.PropsSI('H','P',P8,'S',S8s,fluid)
M8s = M7s #En jaune
```

Pour le mélange dans la bache: il faut tenir compte des débits.

$$m_9 h_9 = m_{4'} h_{4'} + m_{4''} h_{4''} + m_8 h_8$$

```
[20]: H9 = 1/M9 * (M4pps * H4pp + M4p * H4p + M8s * H8s)
```

```
[21]: S9 = CP.PropsSI('S','P',P9,'H',H9,fluid)
S10s = S9 #compression isentropique
H10s = CP.PropsSI('H','P',P10,'S',S10s,fluid)
```

```
[22]: qcs=H1-H10s
wt1s=H3s-H2
wt2s=H6s-H5s
wp1s=H8s-H7
wp2s=H10s-H9
print("Le titre en fin de détente HP isentropique vaut x3s={:.1%}".format(X3s))
print("Le titre en fin de détente BP isentropique vaut x6s={:.1%}".format(X6s))
```

Le titre en fin de détente HP isentropique vaut x3s=82.8%

Le titre en fin de détente BP isentropique vaut x6s=78.8%

Pour le rendement du cycle, il faut tenir compte des débits.

```
[23]: etacycles=abs(M2*wt1s + M5s*wt2s + M7s*wp1s + M9*wp2s)/(M9*qcs)
print("Le rendement du cycle isentropique vaut eta={:.1%}".format(etacycles))
```

Le rendement du cycle isentropique vaut eta=39.1%

1.5 Cycle avec détente en deux phases, machines réelles (3 pts)

```
[24]: etap=0.9
etat=0.85
w23=wt1s*etat #détente HP réelle
H3 = H2 + w23
X3 = CP.PropsSI('Q','P',P3,'H',H3,fluid)
```

Les débits dans les différentes branches sont à recalculer

```
[25]: M4 = M3*X3
M4pp = M3*(1-X3)
H5 = (-M3p * (H3p-H2p)/M4 + H4) #surchauffe
S5 = CP.PropsSI('S','P',P5,'H',H5,fluid)
M5 = M4
S6 = S5 #détente isentropique
M6 = M5
H6ss = CP.PropsSI('H','P',P6,'S',S6,fluid)
```

```

w56s = H6ss-H5
w56 = w56s*etat #détente BP réelle
H6 = H5 + w56
X6 = CP.PropsSI('Q','P',P6,'H',H6,fluid)
M7 = M6
w78s = H8s-H7
w78 = w78s/etap #compression BP réelle
H8 = H7 + w78
M8 = M7
H9 = 1/M9 * (M4pp * H4pp + M4p * H4p + M8 * H8)
S9 = CP.PropsSI('S','P',P9,'H',H9,fluid)
S10ss = S9 #compression isentropique
H10ss = CP.PropsSI('H','P',P10,'S',S10ss,fluid)
w910s = H10ss-H9
w910 = w910s/etap #compression HP réelle
H10 = H9 + w910

```

```

[26]: qc=H1-H10
wt1=H3-H2
wt2=H6-H5
wp1=H8-H7
wp2=H10-H9
print("Le titre en fin de détente HP réelle vaut x3s={:.1%}".format(X3))
print("Le titre en fin de détente BP réelle vaut x6s={:.1%}".format(X6))

```

Le titre en fin de détente HP réelle vaut x3s=85.3%
Le titre en fin de détente BP réelle vaut x6s=83.9%

```

[27]: etacycle=abs(M2*wt1 + M5*wt2 + M7*wp1 + M9*wp2)/(M9*qc)
print("Le rendement du cycle isentropique vaut eta={:.1%}".format(etacycle))

```

Le rendement du cycle isentropique vaut eta=33.6%