Mohammad Fattahi - 9628943
Mohammadreza Sabour - 9628193

Department of Mechanical, Materials and Industrial Engineering K. N. Toosi University of Technology

Refrigeration and Air Conditioning Course

Supervisor: Dr. Behbahaninia

Project Report
Ejector Cycle

May, 2021

Topic

Refrigeration and Air Conditioning

Problem statement

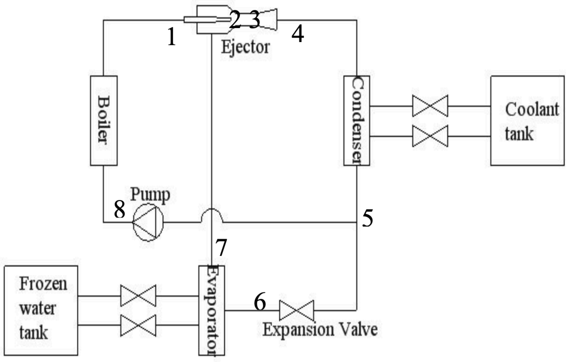
Calculating the ejector cycle's coefficient of performance based on the given data.

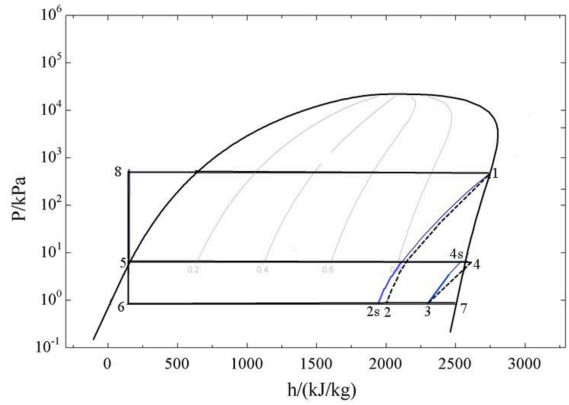
Approach

We implement the governing equations(Continuity, Energy Balance and Momentum) to determine whether the user's input data is valid or not. We write a code in Python programming language to reach this goal. The results are shown in the last section.

Details

Ejector cycle schematic:





Code:

```
# Libraries

import numpy as np   # This library is used for numerical operations
import XSteamPython as stm  # This library is used to calculate the
thermodynamic properties  based on the given data

while True:

    # Input Data

    # mass flow rate ratio
    Mu = input("The mass flow rates Ratio(m_evaporator / m_boiler): ")
    try:
        Mu = float(Mu)
        if Mu == 0:
            print("Mu can't be zero.")
            continue
    except:
        print(" Mu is not valid. Try again.")
        continue

    # Efficiency
    eta_nozzle = float(input("The nozzle efficiency: "))
    eta_diffuser = float(input("The diffuser efficiency: "))
```

```
eta_mixingchamber = float(input("The mixing chamber efficiency: "))

Fluid_density = 997  # Fluid: Water

# Evaporator
Q_evp = float(input("The Evaporator output power(Q_evp) - Scale: KW: "))

# Temperatures

# Evaporator
T_evp =  float(input("The Evaporator temperature (T_evp) - Scale: C : "))

# Condensor
T_cond = float(input("The Condenser temperature (T_cond) - Scale: C : "))

# Boiler
T_boiler = float(input("The Boiler temperature (T_boiler) - Scale: C : "))

# mapping the temperatures
T_0 = T_evp
T_s = T_boiler
T_k = T_cond

# Using XSteam Python library to calculate the required parameters based on
the given data

# Pressures
print("Pressures")
print("--------------------------------------------------------------------------------\n")
P_s = stm.Psat_T(T_s)  # Boiler
print("The Fluid Pressure at Boiler(P_s): {}".format(P_s))
P_0 = stm.Psat_T(T_0)  # Evaporator
print("The Fluid Pressure at Evaporator(P_0): {}".format(P_0))
P_k = stm.Psat_T(T_k)  # Condensor
print("The Fluid Pressure at Condenser(P_k): {}\n\n".format(P_k))

# Enthalpy
print("Enthalpy")
print("--------------------------------------------------------------------------------\n")
h_1 = stm.hV_T(T_s)  # Enthalpy at the boiler outlet
print("The Fluid Enthalpy at Boiler Outlet(h_1): {}".format(h_1))
h_7 = stm.hV_T(T_0)  # Enthalpy at the evaporator outlet
print("The Fluid Enthalpy at Evaporator Outlet(h_7): {}".format(h_7))
```

```
h_5 = stm.hL_T(T_k)  # Enthalpy at the condenser outlet
print("The Fluid Enthalpy at Condenser Outlet(h_5): {}".format(h_5))
print("The Fluid Enthalpy at Evaporator Inlet(h_6): {}".format(h_5))
print("The Fluid Enthalpy at Boiler Inlet(h_8): {}".format(h_5))
h_f_P0 = stm.hL_T(T_0)

# Entropy
s_1 = stm.sV_T(T_s)  # Entropy at boiler outlet
s_7 = stm.sV_T(T_0)  # Entropy at evaporator outlet

# Fluid mass flow rate at evaporator outlet
m_7 = Q_evp / (h_7 - h_5)

# Vapor Fraction
# Assumption: Isentropic condition(S_1 = S_2)
X_prim_2 = stm.x_ps(P_0, s_1)

# Enthalpy at nozzle exit (Isentropic)
h_2s = h_f_P0 + (X_prim_2 * (h_7 - h_f_P0))
h_2 = h_1 - (eta_nozzle * (h_1 - h_2s))  # Enthalpy at nozzle exit (Actual)
print("The Fluid Enthalpy at Nozzle Exit(h_2): {}".format(h_2))

s_f_P0 = stm.sL_T(T_0)
# The fluid entropy at nozzle outlet
s_2 = s_f_P0 + (X_prim_2 * (s_7 - s_f_P0))

# Motive vapor's Velocity (Leaving Nozzle)
u_2 = (eta_nozzle * 2 * (h_1 - h_2s) * 1000)**(1 / 2)
u_3a = ((eta_mixingchamber * (u_2**2) * Mu / (Mu + 1))
        ** (1 / 2))  # Fluid Valocity at diffuser inlet
# Fluid Enthalpy at diffuser inlet
h_3a = ((((Mu * h_1) + h_7) / (Mu + 1)) - ((u_3a**2) / 2000))
print("The Fluid Enthalpy at Shock Inlet(h_3a): {}".format(h_3a))

X_prim_3 = stm.x_ph(P_0, h_3a)  # Vapor Fraction at shock inlet

# Specific Volumns
v_7 = stm.vV_T(T_0)  # Evaporator outlet
v_f_P0 = stm.vL_T(T_0)
v_3a = (X_prim_3 * v_7) + ((1 - X_prim_3) * v_f_P0)  # Shock inlet
# Fluid entropy at shock inlet
s_3a = (X_prim_3 * s_7) + ((1 - X_prim_3) * s_f_P0)
m_3 = u_3a / v_3a
```

```
# Location: Shock
for i in range(500):
    u_3b = float(input("The Fluid Velocity in the Shock Outlet: "))

    P_3a = (0.01) * P_0
    # Pressure at point b (shock outlet) # Scale: Bar
    P_3b = P_3a + ((u_3a - u_3b) * m_3 / (10**5))
    # Fluid enthalpy at the shock outlet
    h_3b = h_3a + ((u_3a**2) / 2000) - ((u_3b**2) / 2000)
    print("The Fluid Enthalpy at Shock Outlet(h_3b): {}".format(h_3b))
    # The fluid specific volume in the shock outlet
    v_3b = (u_3b * v_3a) / u_3a

    # Fluid properties in saturation condition at P_3b
    T_g = stm.Tsat_p(P_3b * 100)  # Temperature(C)
    v_g = stm.vV_p(P_3b * 100)   # Specific Volume
    h_g = stm.hV_p(P_3b * 100)   # Enthalpy
    #print("h_g" , h_g)
    #print("T_g" , T_g)
    s_g = stm.sV_p(P_3b * 100)   # Entropy

    # Temperature (Shock Outlet) (K)
    T_3b = ((T_g + 273.15) * v_3b) / v_g
    #print("T_3b" , T_3b)
    # Temperature difference (Superheat) (C)
    Delta_T_1 = (T_3b - 273) - T_g
    #print("Delta_T_1" , Delta_T_1)
    # New Value for Enthalpy at shock outlet (Based on the assumed
                velocity(u_3b))
    h_3b_new = h_g + (1.885 * Delta_T_1)

    if abs(h_3b_new - h_3b) < 10:
        print("The Fluid Enthalpy at Shock Outlet(New Value: h_3b_new): {}
                    \n”.format(h_3b_new))

        print("The value you have chosen for u_3b is correct. (u_3b = {})
                    \n”.format(u_3b))
        break

    else:
        print("The value for u_3b is not valid. Try again")
        continue
```

```
# Fluid entropy at the shock outlet
s_3b = s_g + (1.885 * np.log(v_3b / v_g))

# Diffuser
# The rise in fluid isentropic enthalpy
Delta_h_isen = (eta_diffuser * (u_3b**2)) / 2000
h_prim_4 = h_3b + Delta_h_isen    # The Fluid Enthalpy After Isentropic
          Diffusion
# The fluid enthalpy at condenser inlet
h_4 = ((h_prim_4 - h_3a) / eta_diffuser) + h_3a
print("The Fluid Enthalpy at Condenser Inlet(h_4): {}\n\n".format(h_4))

# The fluid entropy after isentropic diffusion
s_prim_4 = s_g + (1.885 * np.log(v_3b / v_g))
P_4 = stm.P_hs(h_prim_4, s_prim_4)  # obtained pressure for condenser
s_4 = stm.s_ph(P_4, h_4)  # Fluid entropy at condenser inlet




# Condenser
s_5 = stm.sL_T(T_k)  # Fluid entropy at condenser outlet

# Evaporator Inlet
X_prim_6 = stm.x_ph(P_0, h_5)
s_6 = s_f_P0 + (X_prim_6 * (s_7 - s_f_P0))

# Boiler
s_8 = stm.s_ph(P_s, h_5)  # Fluid entropy at boiler inlet

# Entropy(Results)

print("Entropy")
print("--------------------------------------------------------------------------------\n")
print("The Fluid Entropy at Boiler Outlet(S_1): {}".format(s_1))
print("The Fluid Entropy at Evaporator Outlet(S_7): {}".format(s_7))
print("The Fluid Entropy at nozzle Outlet(S_2): {}".format(s_2))
print("The Fluid Entropy at shock inlet(S_3a): {}".format(s_3a))
print("The Fluid Entropy at condenser inlet(S_4): {}".format(s_4))
print("The Fluid Entropy at condenser outlet(S_5): {}".format(s_5))
print("The Fluid Entropy at evaporator inlet(S_6): {}".format(s_6))
print("The Fluid Entropy at Boiler inlet(S_8): {}\n\n".format(s_8))
```

```
# Velocities(Results)
print("Velocity")
print("--------------------------------------------------------------------------------\n")
print("The fluid velocity at nozzle outlet(u_2): {}".format(u_2))
print("The fluid velocity at shock inlet(u_3a): {}".format(u_3a))
print("The fluid velocity at shock outlet(u_3b): {}\n\n".format(u_3b))

if abs(P_4 - P_k) < 0.6:
    print("The Value for the Saturation Entropy at condenser temperature(s_g
                @ T_k) is valid\n\n")
    m_1 = m_7 * Mu
    W = m_1 * (h_1 - h_5)
    # Coefficient of Performance
    COP = Q_evp / W

    # Heat Transfer Rates
    Q_boiler = W  # Boiler
    Q_cond = (m_1 + m_7) * (abs((h_4 - h_5)))  # Condenser
print("Results")
print("--------------------------------------------------------------------------------\n")
print("The heat transfer rate in boiler: {}".format(Q_boiler))
print("The heat transfer rate in condenser: {}".format(Q_cond))
print("Coefficient of Performance(COP): {}".format(COP))
break

if abs(P_4 - P_k) > 0.6:
    print("The Mu that you have chosen is not valid. Try again.")
continue
```

Results

We executed the code with sample input data, and the result is shown below;

```
Temperatures
--------------------------------------------------------------------------------

The fluid temperature at the evaporator: 10
The fluid temperature at the condenser: 43
The fluid temperature at the boiler: 160

Pressures
--------------------------------------------------------------------------------

The Fluid Pressure at Boiler(P_s): 618.1391967220546
The Fluid Pressure at Evaporator(P_0): 1.2281838693402238
The Fluid Pressure at Condenser(P_k): 8.650261232862325


Enthalpy
--------------------------------------------------------------------------------

The Fluid Enthalpy at Boiler Outlet(h_1): 2757.4305309693405
The Fluid Enthalpy at Evaporator Outlet(h_7): 2519.2298389404236
The Fluid Enthalpy at Condenser Outlet(h_5): 180.07854032438908
The Fluid Enthalpy at Evaporator Inlet(h_6): 180.07854032438908
The Fluid Enthalpy at Boiler Inlet(h_8): 180.07854032438908
The Fluid Enthalpy at Nozzle Exit(h_2): 2037.3253187138075
The Fluid Enthalpy at Shock Inlet(h_3a): 2321.529435351995
The Fluid Enthalpy at Shock Outlet(h_3b): 2674.5129304491056
The Fluid Enthalpy at Shock Outlet(New Value: h_3b_new): 2664.597620313988

The value you have chosen for u_3b is correct. (u_3b = 209.0)

The Fluid Enthalpy at Condenser Inlet(h_4): 2847.63207120501


Entropy
--------------------------------------------------------------------------------

The Fluid Entropy at Boiler Outlet(S_1): 6.74910384530512
The Fluid Entropy at Evaporator Outlet(S_7): 8.899845920563175
The Fluid Entropy at nozzle Outlet(S_2): 6.749103845305122
The Fluid Entropy at shock inlet(S_3a): 8.201627190458385
The Fluid Entropy at condenser inlet(S_4): 8.928262808981716
The Fluid Entropy at condenser outlet(S_5): 0.6122716853324988
The Fluid Entropy at evaporator inlet(S_6): 0.6386627383440787
The Fluid Entropy at Boiler inlet(S_8): 0.6105032019215527


Velocity
--------------------------------------------------------------------------------

The fluid velocity at nozzle outlet(u_2): 1200.08767367683
The fluid velocity at shock inlet(u_3a): 865.8221469760526
The fluid velocity at shock outlet(u_3b): 209.0


The Value for the Saturation Entropy at condenser temperature(s_g @ T_k) is valid


Results
--------------------------------------------------------------------------------

The heat transfer rate in boiler: 15.976565469049739
The heat transfer rate in condenser: 22.237678205316723
Coefficient of Performance(COP): 0.31295837704831764
```

نقاط سیکل

S(Kj / Kg.C)	h(Kj / Kg)	P(bar)	T(C)	Location	Point
6.749	2757.43	618.14	160	Boiler Outlet	1
6.749	2037.33	1.2282	10	Nozzle Outlet	2
8.202	2321.53	1.2282	10	Shock Inlet	3a
-	2674.51	-	-	Shock Outlet	3b
8.928	2847.63	8.65	43	Condenser Inlet	4
0.612	180.079	8.65	43	Condenser Outlet	5
0.638	180.079	1.2282	10	Evaporator Inlet	6
8.899	2519.23	1.2282	10	Evaporator Outlet	7
0.611	180.079	618.14	160	Boiler Inlet	8

انتقال حرارت

Value(KW)	Location	Row
15.977	انتقال حرارت در بویلر	1
22.238	انتقال حرارت در کندانسور	2
Input Value(Assumption: 5 KW)	انتقال حرارت در اواپراتور	3

راندمان اجزا

Value(%)	Location	Row
70	دیفیوزر	1
70	اختلاط	2
85	نازل	3

سرعت

Value(m/s)	Location	Row
1200.088	2	1
865.822	3a	2
Input Value(Assumption: 209)	3b	3

Coefficient of Performance(COP): 0.3129

Note:

You need Python(version > 3.6) for running this code and installing the required libraries and packages. The links below may help you to install it sooner and easier.

Python

XSteam package