Refrigeration Coursework

Submission Details

Working Repository

Student: Joseph Ashton
SID: 27047440 / 22828071
Email: 27047440@students.lincoln.ac.uk

Institution: University of Lincoln
Course: Mechatronics BEng
Module: Energy Systems and Conversion - EGR3030
Module Leader: Aliyu Aliyu, AAliyu@lincoln.ac.uk
Contribution to Final Module Mark: 25
Coursework Title: Coursework 1

Abstract

This report discusses a practical demonstration of a refrigeration cycle using the R634 Demonstration unit. It focuses on the impact of flow rate of coolant in the condenser.

This report dose not explicitly address the questions in the brief but in the annexes there are specific answers to each question posed.

Table of Contents

- Abstract
  - Symbols
- Introduction
  - Background
    - Carnot Cycle
    - R634 Demonstration unit
      - The role of water
    - Coefficient of Performance COP
  - Aims

- Methodology
    - Experimental Methodology
    - Analytical Methodology
        - State Variables
            - Reservoir heat transfer rate
                - Method 1 - Control
                - Method 2 - PYroMat
                - Method 3 - CoolProp
        - Efficiency Metrics
            - COP
            - Isentropic Compressor Efficiency
        - Carnot cycle plots
- Results
- References

Symbols

Experimental variables

Symbol	Label	Description
$\dot{m}_c$	`m/t_c`	**Mass flow rate** of the water in the heat transfer coils of the **condenser**.
$\dot{m}_e$	`m/t_e`	**Mass flow rate** of the water in the heat transfer coils of the **evaporator**.
$T_{e^{in}}$	`T1`	**Temperature** of the water **entering** the heat transfer coils of the **evaporator**.
$T_{e^{out}}$	`T2`	**Temperature** of the water **leaving** the heat transfer coils of the **evaporator**.
$T_{c^{out}}$	`T3`	**Temperature** of the water **leaving** the heat transfer coils of the **condenser**.
$T_{c^{in}}$	`T4`	**Temperature** of the water **entering** the heat transfer coils of the **condenser**.
T_E	`T5`	**Temperature** of the evaporation chamber
T_C	`T6`	**Temperature** of the condensing chamber
p_E	`p_e`	**Pressure** of the evaporation chamber

Symbol	Label	Description
p_C	`p_c`	**Pressure** of the condensing chamber
T_{sh}	`T7`	**Temperature** of the refrigerant leaving the compressor

Calculated variables

Introduction

Background

Carnot Cycle

The Carnot cycle is an idealised thermodynamic cycle that can be used to estimate the behaviour of the refrigeration cycles used in chillers and heat pumps.

R634 Demonstration unit

The role of water

The water supplied to the unit is used as a thermal reservoir to absorb and release thermal energy predictably, that is to say without changing temperature. Naturally when water absorbs / releases energy in the condenser / evaporator respectively it dose change temperature but, by the fact that it is flowing, the system reaches a steady state with a constant and predictable temperature gradient. This average temperature at steady state is the effective temperature of the thermal reservoir.
In the experiment detailed in this report, the flow rate has been used as an independent variable to affect the system. This works by changing the effective temperature of the thermal reservoir and thus effecting the the rate at which it absorbs and emits energy.
If the temperature of the water supplied to the unit is change it will have a similar effect, varying the effective temperature of the thermal reservoirs i.e. the heat exchange coils in the evaporator and condenser.

Disambiguation

In the R634 demonstration unit water is used to transfer heat to and from the evaporator and condenser respectively, and as such it can be seen as playing different roles depending on if it is interoperated as demonstrating a chiller or a heat pump. Throughout this report it is referred to as coolant as a linguistic compromise. At points you may prefer to translate this in your head to "heat source" "thermal transfer medium".

Coefficient of Performance COP

Coefficient of performance (COP) is a measure of efficiency typically associated with heat pumps and heating systems as a whole. In the case of Heat Pumps COP is the ratio of the useful heating power at the condenser per unit of electrical input power. [1]

$$COP = \frac{P\mathrm{f}_{\mathrm{Cond}}}{P_{\mathrm{elec}}}$$

Where:
$P\mathrm{f}_{\mathrm{Cond}}$: Heating capacity of the condenser (kW)
P_{elec} : Input electrical power (kW)

Being a critical measure of energy efficiency there are strict and standardised methodologies for testing and calculating COP[2]. On top of this manufacturer COP claims are often verified by 3rd party organisations via various certification schemes[3] like KEYMARK[4], Qlable and Eurovent[5]. This is necessary as for COP values to be a useful in communicating system performance they must be comparable, like for like with other known values.

The measures available in this experiment are not sufficient to determine of electrical input power or the heating capacity of the condenser as such any approximation of COP will not be comparable with values for heat pumps taken in accordance with the standards. The COP's calculated and discussed from this point onwards apply only to the refrigeration cycle itself, and should only be compared with values based on the same system boundaries.

Aims

Methodology

Experimental Methodology

All testing was done using a stock version of the R634 refrigeration cycle demonstration unit from P. A. HILTON LTD.

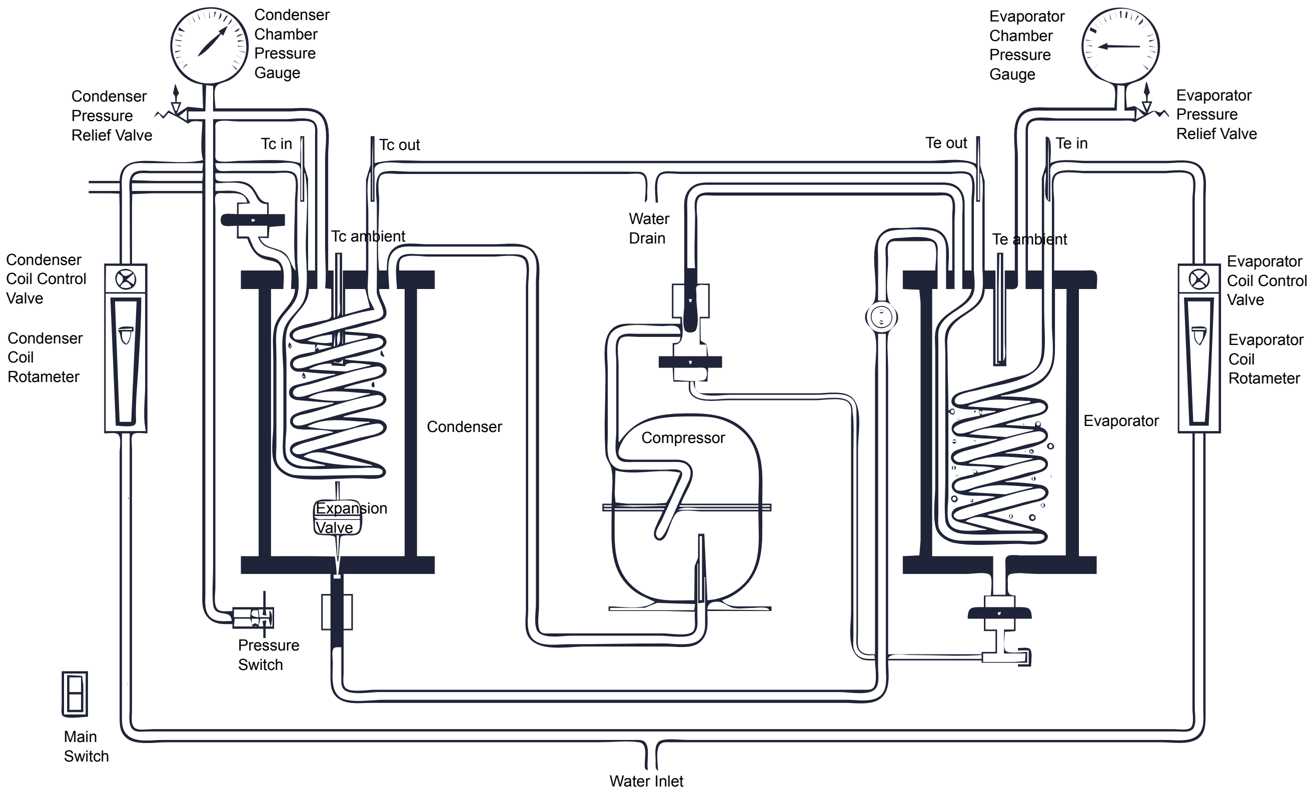
Experimental variables

Symbol	Label	Description
$\dot{m}_c$	`m/t_c`	**Mass flow rate** of the water in the heat transfer coils of the **condenser**.
$\dot{m}_e$	`m/t_e`	**Mass flow rate** of the water in the heat transfer coils of the **evaporator**.
$T_{e^{in}}$	`T1`	**Temperature** of the water **entering** the heat transfer coils of the **evaporator**.
$T_{e^{out}}$	`T2`	**Temperature** of the water **leaving** the heat transfer coils of the **evaporator**.
$T_{c^{out}}$	`T3`	**Temperature** of the water **leaving** the heat transfer coils of the **condenser**.
$T_{c^{in}}$	`T4`	**Temperature** of the water **entering** the heat transfer coils of the **condenser**.
T_E	`T5`	**Temperature** of the evaporation chamber
T_C	`T6`	**Temperature** of the condensing chamber
p_E	`p_e`	**Pressure** of the evaporation chamber
p_C	`p_c`	**Pressure** of the condensing chamber
T_{sh}	`T7`	**Temperature** of the refrigerant leaving the compressor

The main independent variable is condenser flow rate ($\dot{m}_c$) with evaporator flow rate ($\dot{m}_c$) being a secondary independent variable.

The condenser flow rate ($\dot{m}_c$) was tested at regular intervals between $2\ \mathrm{g/s}$ to $12\ \mathrm{g/s}$ at a low and a high evaporator flow rate ($\dot{m}_c$), being $10\ \mathrm{g/s}$ and $20\ \mathrm{g/s}$ respectively.

The mass flow rates determined by readings taken from rotameters placed before the coils in the R634 demonstration unit. The coils in the evaporator and condenser both have thermometers measuring the fluid temperature as it enters and leaves the respective coil as well as thermometers measuring the ambient temperature of the phase change chamber.


Condenser Chamber Pressure Gauge
Condenser Pressure Relief Valve
Tc in
Tc out
Te out
Te in
Evaporator Chamber Pressure Gauge
Evaporator Pressure Relief Valve
Water Drain
Tc ambient
Te ambient
Condenser Coil Control Valve
Condenser Coil Rotameter
Condenser
Compressor
Evaporator
Evaporator Coil Control Valve
Evaporator Coil Rotameter
Expansion Valve
Pressure Switch
Main Switch
Water Inlet

The tests was carried out as follows:

1. Carry out the initial setup for the R634 unit as described in the R634 refrigeration cycle demonstration unit manual.pdf which can be accessed here in the working repository. Set the evaporator flow rate ($\dot{m}_c$) flow rate to 20 g/s and condenser flow rate ($\dot{m}_c$) to 12 g/s and monitor the temperatures until they have remained constant for 5 minuets.
2. Carry out a reading, record all the variables listed in Table of Symbols > Experimental variables
3. Reduce the condenser flow rate ($\dot{m}_c$) by 2 g/s and allow system to stabilise as described in step 1 and repeat from step 2 until condenser flow rate ($\dot{m}_c$) is at it's minimum or the unit approaches a maximum condenser pressure of $150\ kN/m^2$.
4. Set the evaporator flow rate ($\dot{m}_c$) to 10 g/s and repeat from step 2.

Analytical Methodology

State Variables

Reservoir heat transfer rate

The rate of heat transfer in the evaporator and condenser can be said to be equal to the change in energy of the fluid between the input and output of their respective coils if the systems are assumed to be perfectly insulated. In practice this is of course not the case as the system will inevitably loose energy to and gain energy from the environment by means of conduction, radiation and sound.

As this is a fairly straightforward value to compute it was taken as an opportunity to compare several methodologies.

There are many means of identifying fluid properties ranging in their ease and accuracy. In this module we are taught to seek out property tables which can be impractical at scale and prone to human error. As such I set out to understand the alternatives, of the options I found 2 stood out Pyromat and CoolProp.

See Finding fluid properties in the annexes or here in my module repo for more details

Method 1 - Control

The energy change of the water across the coils can be found as the product of its specific heat capacity c, the change in temperature ΔT, and the mass flow rate $\dot{m}$.

$$Q = c\dot{m}\Delta T$$

While specific heat capacity c dose vary by temperature and pressure both of which will vary along the length of the coil and by the setting of the control valve, this will be ignored as the variation is negligible at less than $\pm 0.01\ kJ/kg \cdot {}^{\circ}K$. [6] [7] Therefore a value of $4.1813\ kJ/kg \cdot {}^{\circ}K$ will be used corresponding to the constant pressure specific heat capacity c_p at a standard temperature and pressure of $298.15^{\circ}K$ and $101.33\ kPa$ respectively. [8]

This was implemented pragmatically in python with the function seen below:

```python
def method_1(lab_readings): # Method 1 - $q = \dot{m} c_{p} \detla T$

    c_p = 4181.3  # constant pressure specific heat capacity of water @ 101325 Pa, 298.15 K

    # temperature change of the fluid in the coolant coils (K)
    dT_e = lab_readings['T2'].values - lab_readings['T1'].values # evaporator coil
    dT_c = lab_readings['T3'].values - lab_readings['T4'].values
```

```
# condenser coil

    # energy transfer (W) product of mass flow rate, temperature
change & specific heat capacity
    dQ_e = c_p*dT_e*lab_readings['m/t e'].values
    dQ_c = c_p*dT_c*lab_readings['m/t c'].values

    return (dQ_e,dQ_c)
```

The full file `heat_flux.py` **can be found in the annexes and** [here in the working repository]()

Method 2 - PYroMat

PYroMat[9] is a intuitive and lightweight python library, It's simple to make thermodynamic property queries and perfect for quick and scrappy use in the console. I could recommend 90% of the time.

The function below utilises PYroMat's multi phase property model based on:

> T. Petrova, “Revised release on the iapws formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use,” tech. rep., 2014.[10]

For more information see http://pyromat.org/pdf/handbook.pdf#chapter.7 [11]

```
def method_2(lab_readings): # Method 2 PYroMat for heat transfer
rate
    import pyromat as pm

    water = pm.get("mp.H2O")  # fetches multiphase thermodynamic
property model

    # specific internal energy change of the fluid across the
coolant coils (kJ)
    de_e = water.e(T=lab_readings['T2'].values) -
water.e(T=lab_readings['T1'].values) # evaporator coil
    de_c = water.e(T=lab_readings['T3'].values) -
water.e(T=lab_readings['T4'].values) # condenser coil

    # energy transfer rate (W) product of energy change and mass
```

```
flow rate
    dQ_e = de_e*lab_readings['m/t e'].values*1000
    dQ_c = de_c*lab_readings['m/t c'].values*1000

    return (dQ_e,dQ_c)
```

The full file `heat_flux.py` can be found in the annexes and here in the working repository

Method 3 - CoolProp

CoolProp[12] is a free open source C++ implementation of REFPROP[13] the NIST's thermodynamic property calculator. It has both high and low level functionality. It also has wrappers for anything your likely to want to use even excel. For the sake of consistency this report will only use Python.

CoolProp is slightly more involved and less convenient for quick queries but is far more capable.

```
def method_3(lab_readings): # Method 3 CoolProp for heat transfer
rate

    from CoolProp.CoolProp import PropsSI

    # specific internal energy change of the fluid across the
coolant coils (J)
    de_e = PropsSI('H', 'T', lab_readings['T2'].values, 'P',
101325, 'H2O') - PropsSI('H', 'T', lab_readings['T1'].values,
'P', 101325, 'H2O') # evaporator coil
    de_c = PropsSI('H', 'T', lab_readings['T3'].values, 'P',
101325, 'H2O') - PropsSI('H', 'T', lab_readings['T4'].values,
'P', 101325, 'H2O') # condenser coil

    # energy transfer rate (W), product of energy change and mass
flow rate
    dQ_e = de_e*lab_readings['m/t e'].values
    dQ_c = de_c*lab_readings['m/t c'].values

    return (dQ_e,dQ_c)
```

The full file `heat_flux.py` can be found in the annexes and here in the working repository

Efficiency Metrics

COP

As discussed in the Background > Coefficient of Performance COP section COP depends heavily on the system boundaries chosen. This methodology defines the system boundaries at:

- Input: work done to refrigerant by the compressor
- Output: work done by the refrigerant in the condenser

Making COP a measure of:

$$COP = \frac{\dot{Q}_{\text{condensation}}}{P_{\text{compression}}}$$

Where:
$\dot{Q}_{\text{condensation}}$: rate of heat released by the refrigerant in the condenser
$P_{\text{compression}}$: effective power of the compressor

Under the assumption that all work done in the condenser is converted to heat, which is a reasonable approximation, Q becomes W. Further all work done to and by the refrigerant W is to be represented by the sum of it's change in enthalpy H, and time t.

$$COP = \frac{\dot{Q}_{\text{condensation}}}{P_{\text{compression}}} \triangleq \frac{\delta H_{\text{cond}} \times t}{\delta H_{\text{comp}} \times t}$$

As the the refrigerant is in an essentially closed system that has been allowed to reach a steady state, the mass flow rate is uniform through out. As such we can cancel it out from numerator and denominator to give:

$$\frac{\delta H_{\text{cond}} \times t}{\delta H_{\text{comp}} \times t} \div \frac{mt^{-1}}{mt^{-1}} = \frac{\delta h_{\text{cond}}}{\delta h_{\text{comp}}}$$

The enthalpy of the refrigerant in this case SES36 can be found based on it's temperature and pressure using a thermodynamic chart, table or model. The source used to predict the enthalpy will effect the values as in all cases they are approximations. It is important to verify that the model is sufficiently accurate in the temperature/pressure range you are working in and that it is from a reputable source. In this case the model is based on:

> Monika Thol, Eric W. Lemmon, and Roland Span. Equation of State for a Refrigerant Mixture of R365mfc (1,1,1,3,3-Pentafluorobutane) and Galden HT 55 (Perfluoropolyether). Unpublished, 2012.[14]

Despite being unpublished this was chosen based on recognition of one of the authors Eric W. Lemmon who is prolific in the field having developed 2 of the most predominant methodologies for identifying equations of state[15] as well as RefProp the NIST's thermodynamic property calculator[16]. It was also verified by visualy comparing the P-H diagram it produces against the one included in the annexes of the R634 refrigeration cycle demonstration unit manual.pdf.

By assuming no change in temperature enthalpy change in the condenser can be found as the difference between the enthalpy of the refrigerant SES36 at saturated vapour vs saturated liquid.
By assuming that the temperature and pressure of the refrigerant at the inlet of the compressor are the same as they where at the evaporator, and that the refrigerant is in a state of saturated vapour the initial enthalpy can be found. The final enthalpy is the same but based on the compressor discharge temperature and the condenser pressure.

This was implemented problematically using CoolProp[17] as seen below:

```python
def method_1(lab_readings):
    from CoolProp.CoolProp import PropsSI
    material = 'SES36'

    # Condenser enthalpy change
    P_initial = lab_readings['p c'].values
    T_initial = lab_readings['T6'].values

    H_initial = PropsSI('H', 'T', T_initial, 'P|gas', P_initial, material)

    P_final = P_initial
    T_final = T_initial

    H_final = PropsSI('H', 'T', T_final, 'P|liquid', P_final, material)
    dH_condensation = H_initial - H_final

    # Compressor ethalpy change
    P_initial = lab_readings['p e'].values
    T_initial = lab_readings['T5'].values

    H_initial = PropsSI('H', 'T', T_initial, 'P|gas', P_initial,
material)

    P_final = lab_readings['p c'].values
    T_final = lab_readings['T7'].values

    H_final = PropsSI('H', 'T', T_final, 'P|gas', P_final,
material)

    dH_compression = H_initial - H_final

    cop = abs(dH_condensation/dH_compression)

    return cop
```

The full file `cop.py` can be found in the annexes and here in the working repository

Isentropic Compressor Efficiency

Isentropic compressor Efficiency (η_s) is a measure of performance based on the ratio of the work necessary to achieve the compression isentropically (the ideal case where no entropy is added) over the actual work done.

$$\eta_s = \frac{W_{\text{isentropic}}}{W_{\text{actual}}}$$

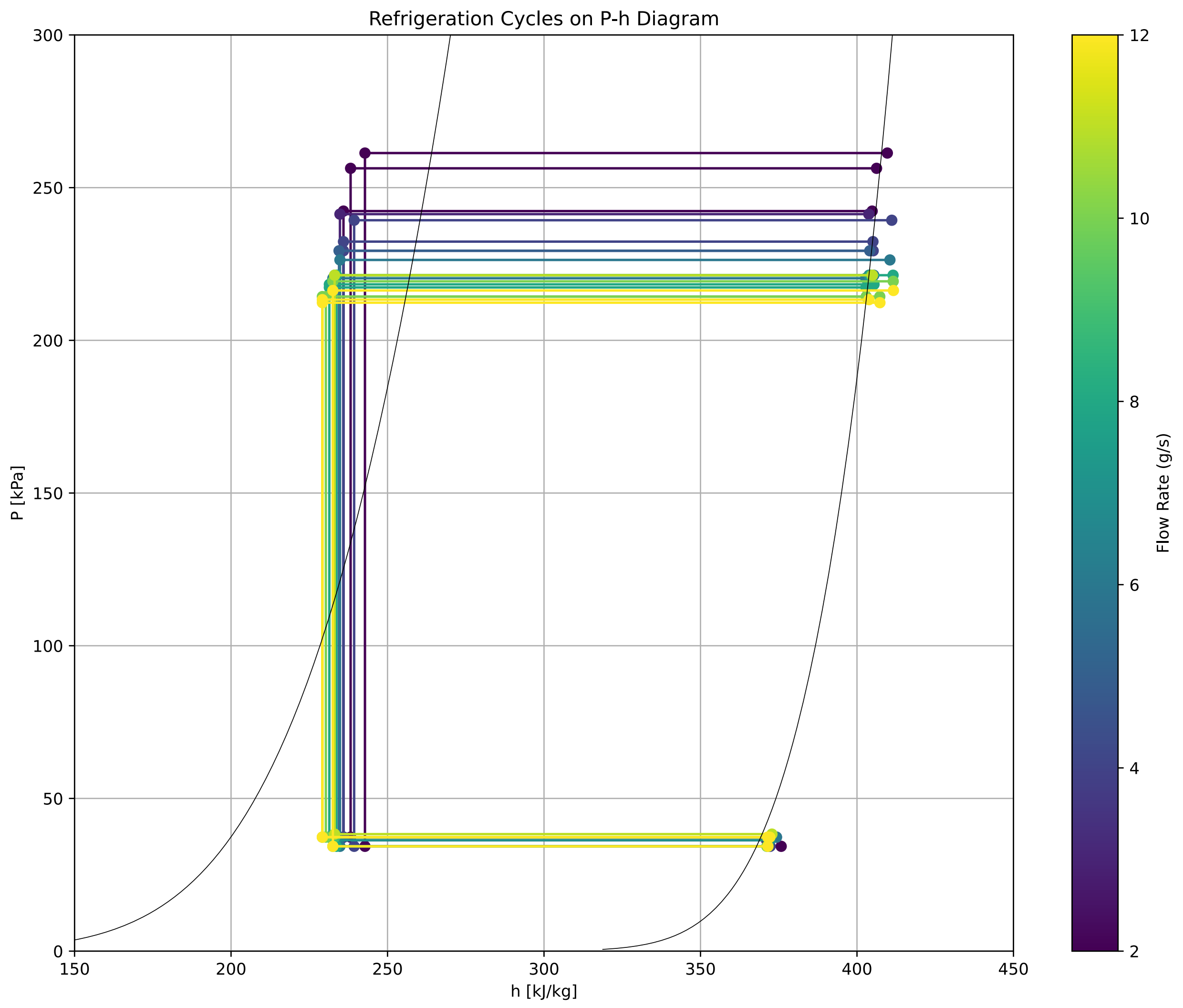
This was implemented pragmatically with the function seen below:

```
def isentropic_efficiency(T_in, P_in, T_out, P_out):
    import CoolProp.CoolProp as CP

    h_in = CP.PropsSI('H', 'T', T_in, 'P|gas', P_in, "SES36")
    S_in =  CP.PropsSI('S', 'T', T_in, 'P|gas', P_in, "SES36")

    h_out_max = CP.PropsSI('H', 'S', S_in, 'P|gas', P_out, 
"SES36")
    h_out_actual = CP.PropsSI('H', 'T', T_out, 'P|gas', P_out, 
"SES36")

    return (h_out_max-h_in)/(h_out_actual-h_in)
```

The full file `cop.py` **can be found in the annexes and** here in the working repository

Carnot cycle plots

Thermodynamic cycle plots provide a visual intuition for a given process. The Carnot cycle is an idealised approximation of a refrigeration cycle like the one that occurs in the R634 unit. Below are idealised approximations of what all the cycles would have been for the flow rates tested.

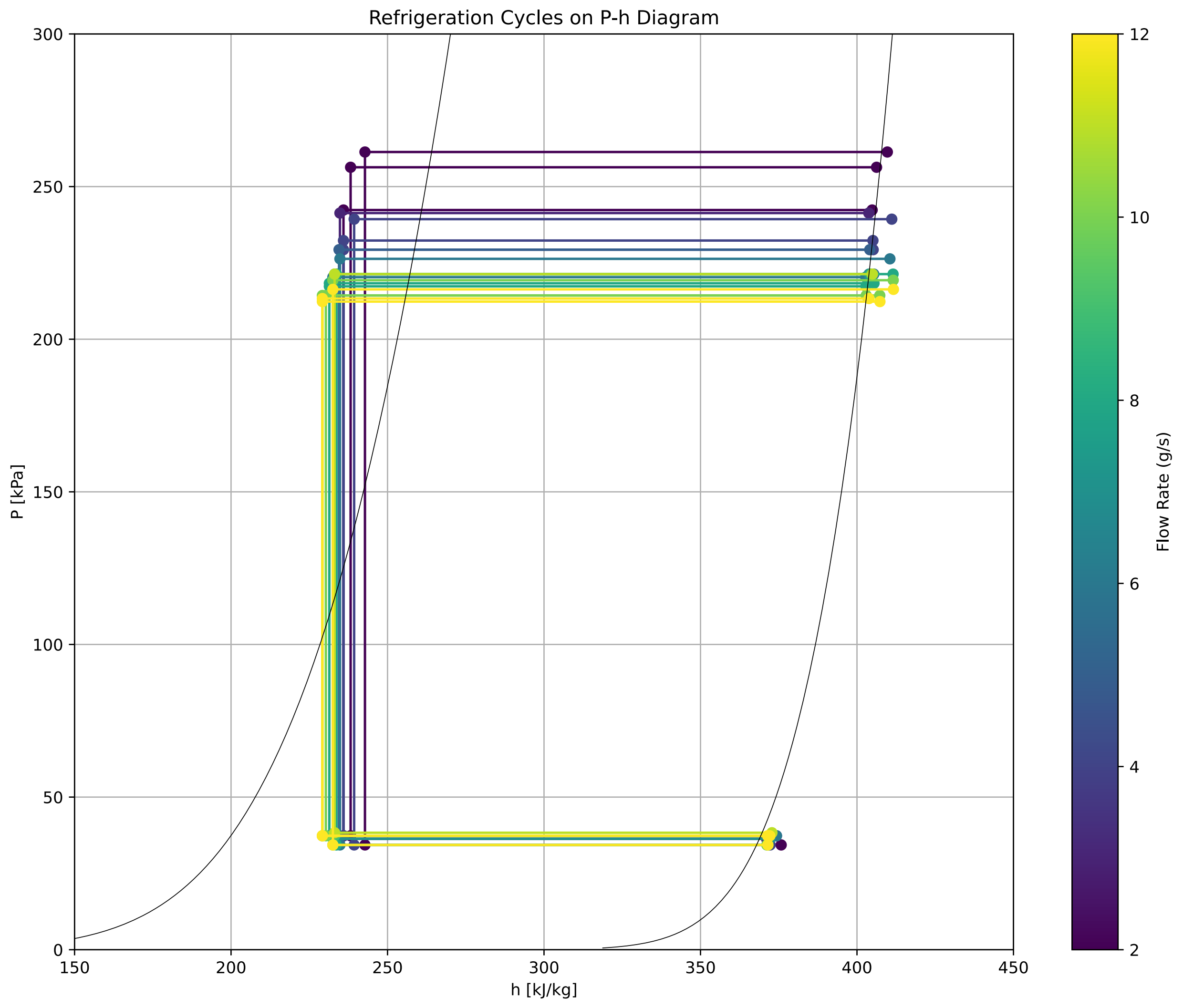


To achieve this I wrote the `plot_PH` function that can be found in plot_cycles.py. It's based loosely on this example I found on stack overflow[18] utilising my `isentropic_efficiency` function mentioned above along with with the SimpleCyclesCompression module from coolProp[19]. It is part of the cop.py file that can be seen in it's entirety in the annexes or here in the repo.

```python
import CoolProp.CoolProp as CP
from CoolProp.Plots import PropertyPlot
from CoolProp.Plots import SimpleCompressionCycle
import matplotlib.pyplot as plt
from scipy import interpolate

from cop import isentropic_efficiency

def plot_PH(lab_readings):
    flow_rate = lab_readings["m/t c"].values
    T_evap = lab_readings["T5"].values
    P_evap = lab_readings["p e"].values
    T_cond = lab_readings["T6"].values
    P_cond = lab_readings["p c"].values
    T_comp = lab_readings["T7"].values

    eta_com = isentropic_efficiency(T_evap,P_evap,T_comp,P_cond)

    ph_plt = PropertyPlot('SES36', 'PH', unit_system='KSI')

    ph_plt.xlabel("h [kJ/kg]")
    ph_plt.ylabel("P [kPa]")

    ph_plt.calc_isolines(CP.iQ)

    cycle = SimpleCompressionCycle("SES36", "PH",
unit_system="KSI")
    cycle.set_axis_limits([0, 500, 0, 3500])
    colours = plt.cm.viridis((flow_rate-
flow_rate.min())/(flow_rate.max()-flow_rate.min()))

    for i, entry in enumerate(zip(T_evap,P_evap,T_cond,P_cond,
eta_com, colours)):
        cycle.simple_solve(*entry[:-1])

        sc = cycle.get_state_changes()
        colour = colours[i]
        ph_plt.draw_process(sc, line_opts={"color":colour})

    ph_plt.title("Refrigeration Cycles on P-h Diagram")
    ph_plt.grid()
    sm = plt.cm.ScalarMappable(norm=plt.Normalize(vmin=2,
vmax=12), cmap="viridis")
    cbar = plt.colorbar(sm,ax=ph_plt.axis)
    cbar.set_label("Flow Rate (g/s)")
    ph_plt.show()
```

ⓘ **The full file `plot_cycles.py` can be found in the annexes and** here in the working repository

Results



References

1. Certification reference standard NF 414, AFNOR NF 414, 2024. Available: https://www.eurovent-certification.com/media/images/349/3c3/3493c319d9de88e18e3ecea2156c1e8d6521a7d4.pdf ↩
2. Heating systems and water based cooling systems in buildings. Method for calculation of system energy requirements and system efficiencies. Part 4-2. Space heating generation systems, heat pump systems, BS EN 14511-3, 2018. Download ↩
3. "EHPA Certification." [Online]. Available: https://www.ehpa.org/quality/. [Accessed: 22-Nov-2024]. ↩
4. "Testing and Certification." [Online]. Available: https://keymark.eu/en/products/heatpumps/testing-and-certification. [Accessed: 22-Nov-2024]. ↩
5. "NF414 | Eurovent Certita Certification." [Online]. Available: https://www.eurovent-certification.com/en/third-party-certification/certification-programmes/nf414. [Accessed: 22-Nov-2024]. ↩
6. "Water - Properties vs. Temperature and Pressure." [Online]. Available: https://www.engineeringtoolbox.com/water-properties-d_1258.html. [Accessed: 13-Nov-2024]. ↩
7. "About | PYroMat." [Online]. Available: http://www.pyromat.org/about.html. [Accessed: 13-Nov-2024].

```
>>> import pyromat as pm
>>> water = pm.get("mp.H2O")
>>> water.cp(p=1, T=283.15) - water.cp(p=1.5, T=323.15)
array([0.01393411]) # The variation in c_p
```

↩

8. "About | PYroMat." [Online]. Available: http://www.pyromat.org/about.html. [Accessed: 13-Nov-2024].

```
>>> import pyromat as pm
>>> pm.get("mp.H2O").cp(T=298.15,p=1.01325)
array([4.1813595]) # c_p @ STP
```

↩

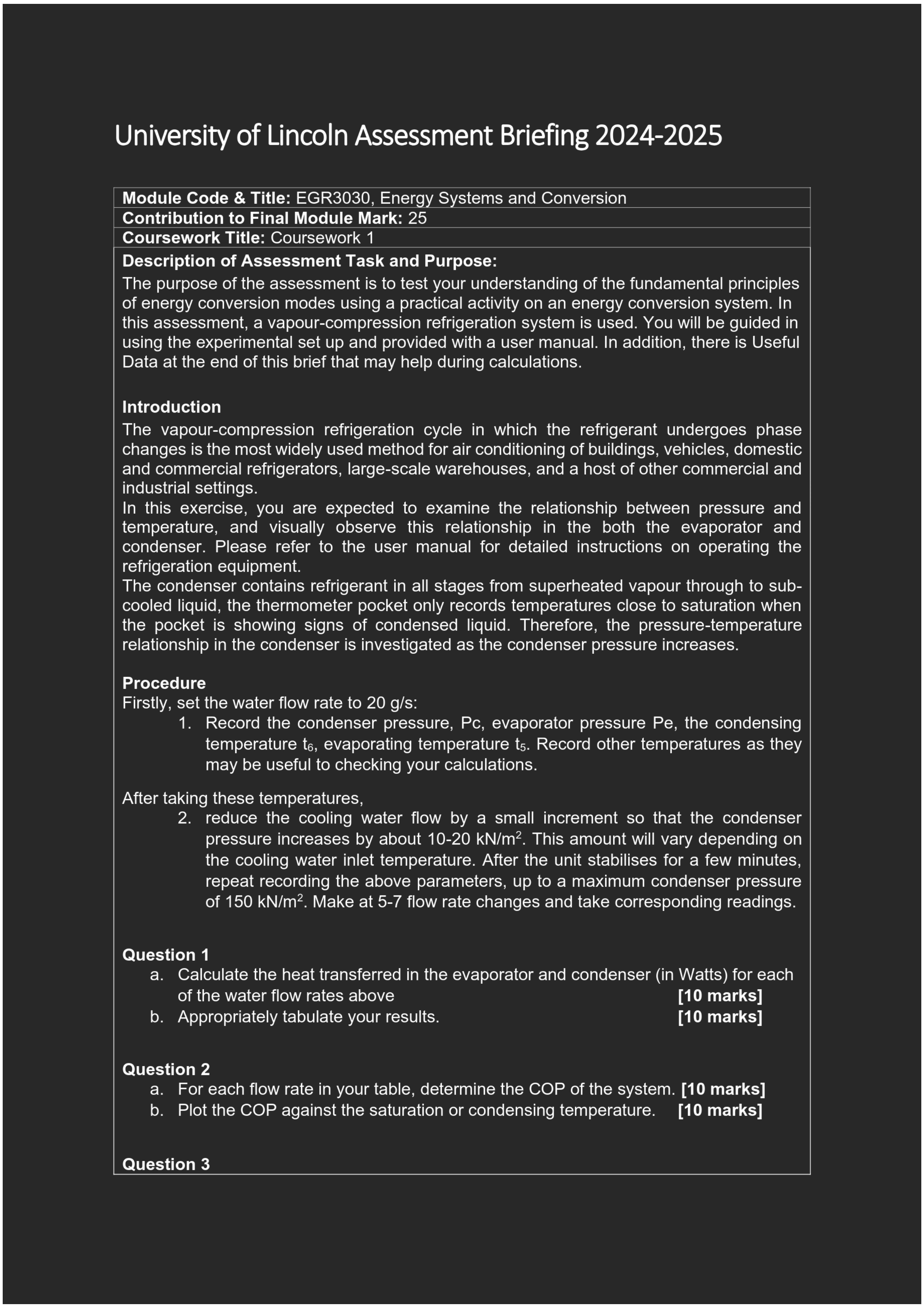
9. "Home | PYroMat." [Online]. Available: http://pyromat.org/. [Accessed: 15-Nov-2024]. ↩
10. T. Petrova, "Revised release on the iapws formulation 1995 for the thermodynamic properties of ordinary water substance for general and scientific use," tech. rep., 2014. ↩

11. "Chapter 7 Multi-phase substance models | PYroMat Handbook." [Online]. Available: http://pyromat.org/pdf/handbook.pdf#chapter.7. [Accessed: 15-Nov-2024]. ↩
12. “Welcome to CoolProp — CoolProp 6.6.0 documentation.” [Online]. Available: http://www.coolprop.org/. [Accessed: 28-Nov-2024]. ↩
13. “REFPROP,” 18-Apr-2013. [Online]. Available: https://www.nist.gov/srd/refprop. [Accessed: 28-Nov-2024]. ↩
14. Monika Thol, Eric W. Lemmon, and Roland Span. Equation of State for a Refrigerant Mixture of R365mfc (1,1,1,3,3-Pentafluorobutane) and Galden HT 55 (Perfluoropolyether). Unpublished, 2012. ↩
15. E. W. Lemmon and R. Tillner-Roth, “A Helmholtz energy equation of state for calculating the thermodynamic properties of fluid mixtures,” 04-Nov-1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378381299002629. [Accessed: 28-Nov-2024]. ↩
16. “Eric W. Lemmon,” 09-Oct-2019. [Online]. Available: https://www.nist.gov/people/eric-w-lemmon. [Accessed: 28-Nov-2024]. ↩
17. “SES36 — CoolProp 6.6.0 documentation.” [Online]. Available: http://www.coolprop.org/fluid_properties/fluids/SES36.html. [Accessed: 28-Nov-2024]. ↩
18. “Put labels in Coolprop Chart.” [Online]. Available: https://stackoverflow.com/questions/70864726/put-labels-in-coolprop-chart. [Accessed: 26-Nov-2024]. ↩
19. “CoolProp.Plots.SimpleCyclesCompression module — CoolProp 6.6.0 documentation.” [Online]. Available: http://www.coolprop.org/apidoc/CoolProp.Plots.SimpleCyclesCompression.html. [Accessed: 26-Nov-2024]. ↩

Annexes

Brief

of 4



University of Lincoln Assessment Briefing 2024-2025

Module Code & Title: EGR3030, Energy Systems and Conversion
Contribution to Final Module Mark: 25
Coursework Title: Coursework 1

Description of Assessment Task and Purpose:

The purpose of the assessment is to test your understanding of the fundamental principles of energy conversion modes using a practical activity on an energy conversion system. In this assessment, a vapour-compression refrigeration system is used. You will be guided in using the experimental set up and provided with a user manual. In addition, there is Useful Data at the end of this brief that may help during calculations.

Introduction

The vapour-compression refrigeration cycle in which the refrigerant undergoes phase changes is the most widely used method for air conditioning of buildings, vehicles, domestic and commercial refrigerators, large-scale warehouses, and a host of other commercial and industrial settings.

In this exercise, you are expected to examine the relationship between pressure and temperature, and visually observe this relationship in the both the evaporator and condenser. Please refer to the user manual for detailed instructions on operating the refrigeration equipment.

The condenser contains refrigerant in all stages from superheated vapour through to sub-cooled liquid, the thermometer pocket only records temperatures close to saturation when the pocket is showing signs of condensed liquid. Therefore, the pressure-temperature relationship in the condenser is investigated as the condenser pressure increases.

Procedure

Firstly, set the water flow rate to 20 g/s:

1. Record the condenser pressure, Pc, evaporator pressure Pe, the condensing temperature t_6, evaporating temperature t_5. Record other temperatures as they may be useful to checking your calculations.

After taking these temperatures,

2. reduce the cooling water flow by a small increment so that the condenser pressure increases by about 10-20 kN/m^2. This amount will vary depending on the cooling water inlet temperature. After the unit stabilises for a few minutes, repeat recording the above parameters, up to a maximum condenser pressure of 150 kN/m^2. Make at 5-7 flow rate changes and take corresponding readings.

Question 1

a. Calculate the heat transferred in the evaporator and condenser (in Watts) for each of the water flow rates above **[10 marks]**
b. Appropriately tabulate your results. **[10 marks]**

Question 2

a. For each flow rate in your table, determine the COP of the system. **[10 marks]**
b. Plot the COP against the saturation or condensing temperature. **[10 marks]**

Question 3

Lab Notes

Procedure

[Brief - Energy Systems and Conversion Coursework.pdf]

1. Set the water flow rate to $20\ g/s$
   Record:
   - condenser pressure, P_c,
   - evaporator pressure P_e,
   - condensing temperature t_6,
   - evaporating temperature t_5,
   - any other temperatures.
2. Reduce the cooling water flow by a small increment so that the condenser pressure increases by about $10 - 20\ kN/m^2$.
   This amount will vary depending on the cooling water inlet temperature.
   Allow system to stabilise.
   Repeat Recordings of:
   - condenser pressure, P_c,
   - evaporator pressure P_e,
   - condensing temperature t_6,
   - evaporating temperature t_5,
   - any other temperatures.
3. Repeat step 2 for 5-7 flow rate changes up to a maximum condenser pressure of $150\ kN/m^2$ and take corresponding readings.

Readings

Lab Readings - Energy Systems and Conversion Coursework.csv

m/t c (24)	m/t e	T1	T2	T3	T4	T5	T6	T7
3	10	19.5	16.5	35.25	19	11.5	41	60.75
5	10	20.75	18	32	20.75	13	40.75	60.75
7	10	21	18.25	29.25	21	13	39.75	60.25
9	10	21	18.5	28	21	13	39.5	61
11	10	21.5	18.5	27.5	21.5	14	39.5	61.5
2	10	21	18	48	21	18	48	68
4	10	21	18	35	21	13	45	69
6	10	21	18	30	21	12	41	68
8	10	21	18	28	21	12	40	69

m/t c (24)	m/t e	T1	T2	T3	T4	T5	T6	T7
10	10	21	18	27	21	11.5	39	69
12	10	20	17	26	20	12	39	69
2	20	19	18	40	19	13	42	62
4	20	19	15	37	18	16	42	62
6	20	18	15	27	18	16	39	59
8	20	19	18	27	19	13	38	59
10	20	19	18	26	19	13	37	59
12	20	19	18	25	19	13	36	60
2	20	19	18	43	20	13	44	64
4	20	19	18	34	19	13	42	62
6	20	18.5	18	29	19	12.5	40	62
8	20	19	18	26.5	19	13	38	62
10	20	19	18	25.5	19	12.5	36	64
12	20	19	17.5	24	19	12	36	64

Limited width table

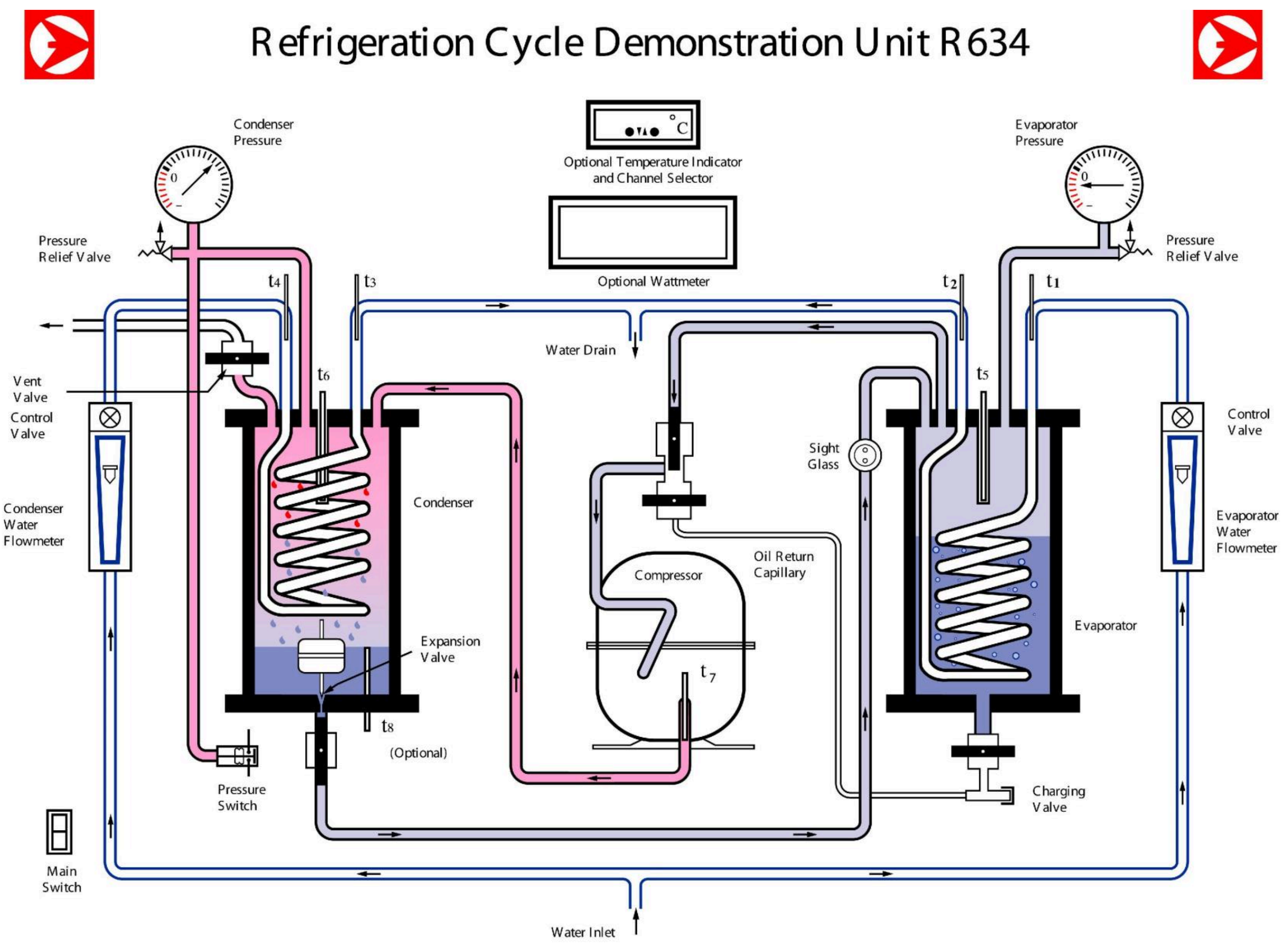
$\dot{m}_c$	$\dot{m}_e$	T_E	$T_{e^{in}}$	$T_{e^{out}}$	T_C	$T_{c^{in}}$	$T_{c^{out}}$	p_C	p_E
2	10	18	21	18	48	48	21	160	-67
4	10	13	21	18	45	35	21	138	-67
6	10	12	21	18	41	30	21	125	-67
8	10	12	21	18	40	28	21	120	-67
10	10	11.5	21	18	39	27	21	118	-67
12	10	12	20	17	39	26	20	115	-67

Additional Resources

Demonstration unit Manual

Blackboard Link
Local file

Demonstration Unit Diagram



Notes

Condenser flow rate oscillatory behaviour

During the experiment while waiting for the system to stabilise after adjusting the condenser flow rate valve the rotameter float would observably fall and rise multiple times. It appeared there was a distinctively oscillatory process taking place. There was a significant time delay between our inputs, rising events and falling events in which the float would appear stationary.

During the experiment the following possible factors where proposed:

1. The change in mass flow rate changes the effective thermal conductance throughout the condenser coil by several mechanisms:
    - flow rate will affect the boundary temperature differential throughout the pipe.
    - flow rate is proportional to Reynolds number which affects boundary thermal conductance.

2. The condenser flow rate valve also effects the pressure in the condensation coil which will change the evaporating/condensing point.
3. Rapid changes in flow rate could induce pressure waves in the pipes which would be partially reflected at any elements with associated pressure changes.
4. The system is connected to the INB buildings plumbing, it could simply be observation bias that we have correlated it with our inputs. The flow rate changes could be associated with pressure drops in the buildings water system when toilets are flushed etc.

Mechanisms for the effect where proposed based on these factors

#WIP

https://www.sciencedirect.com/science/article/pii/S1110016821004646

Questions

Table of Contents

- Q1 - Heat Transfer Rate
- Q2 - COP
- Q3 - PvT
- Q4 - dQvT
- Q5 - Vary inlet water temp
- Q6 - Water as a refrigerant

Question 1

a) Calculate the heat transferred in the evaporator and condenser (in Watts) for each of the water flow rates above

[10 marks]

Ambiguity

It should be noted that when being used as an approximation of internal energy heat is measured in joules (J) and so the heat transferred would be too. However it is fair to assume this question intends to ask for the heat transfer rate and not it's quantity.

The rate of heat transfer in the evaporator and condenser can be said to be equal to the change in energy of the fluid between the input and output of their respective coils if the systems are assumed to be perfectly insulated. In practice this is of course not the case as the system will inevitably loose energy to and gain energy from the environment by means of conduction, radiation and sound. If this is to be considered, with the aim of distinguishing between:

1. energy transferred within the evaporator/condenser from refrigerant to coolant and,
2. energy transfer of the evaporator/condenser to and from the environment,

then one would have to measure the energy change of the refrigerant and coolant independently.

In this case the following measures have been recorded:

Symbol	Label	Description
$\dot{m}_c$	`m/t_c`	Mass flow rate of the water in the heat transfer coils of the condenser.
$\dot{m}_e$	`m/t_e`	Mass flow rate of the water in the heat transfer coils of the evaporator.
$T_{e^{in}}$	`T1`	Temperature of the water entering the heat transfer coils of the evaporator.
$T_{e^{out}}$	`T2`	Temperature of the water leaving the heat transfer coils of the evaporator.
$T_{c^{out}}$	`T3`	Temperature of the water leaving the heat transfer coils of the condenser.
$T_{c^{in}}$	`T4`	Temperature of the water entering the heat transfer coils of the condenser.
T_E	`T5`	Temperature of the evaporation chamber
T_C	`T6`	Temperature of the condensing chamber
p_E	`p_e`	Pressure of the evaporation chamber

Symbol	Label	Description
p_C	`p_c`	Temperature of the evaporation chamber
T_{sh}	`T7`	Temperature of the refrigerant leaving the compressor

The mass flow rates are readings taken from rotameters placed before the coils in the R634 demonstration unit. The coils in the evaporator and condenser both have thermometers measuring the fluid temperature as it enters and leaves the respective coil the difference between these readings is considered the temperature change.

Method 1

The energy change of the water across the coils can be found as the product of its specific heat capacity c, the change in temperature ΔT, and the mass flow rate $\dot{m}$.

$$Q = c\dot{m}\Delta T$$

While specific heat capacity c dose vary by temperature and pressure both of which will vary along the length of the coil and by the setting of the control valve, this will be ignored as the variation is negligible at less than $\pm 0.01\ kJ/kg \cdot {}^{\circ}K$. [1] [2] Therefore a value of $4.1813\ kJ/kg \cdot {}^{\circ}K$ will be used corresponding to the constant pressure specific heat capacity c_p at a standard temperature and pressure of $298.15^{\circ}K$ and $101.33\ kPa$ respectively. [3]

This was implemented pragmatically in python with the function seen below:

```python
def method_1(lab_readings): # Method 1 - $q = \dot{m} c_{p} \detla T$

    c_p = 4181.3  # constant pressure specific heat capacity of water
@ 101325 Pa, 298.15 K

    # temperature change of the fluid in the coolant coils (K)
    dT_e = lab_readings['T2'].values - lab_readings['T1'].values  #
evaporator coil
    dT_c = lab_readings['T3'].values - lab_readings['T4'].values  #
condenser coil

    # energy transfer (W) product of mass flow rate, temperature
```

```
change & specific heat capacity
    dQ_e = c_p*dT_e*lab_readings['m/t e'].values
    dQ_c = c_p*dT_c*lab_readings['m/t c'].values

    return (dQ_e,dQ_c)
```

ⓘ The full file `heat_flux.py` can be found in the annexes and here in the working repository

Method 2 - PYroMat

```
def method_2(lab_readings): # Method 2 PYroMat for heat transfer rate

    import pyromat as pm

    """
    utilising pyromat multi phase property model based on:
      - T. Petrova, “Revised release on the iapws formulation 1995
for the
        thermodynamic properties of ordinary water substance for
general
        and scientific use,” tech. rep., 2014.
    For more information see
http://pyromat.org/pdf/handbook.pdf#chapter.7
    """

    water = pm.get("mp.H2O")  # fetches multiphase thermodynamic
property model

    # specific internal energy change of the fluid across the coolant
coils (kJ)
    de_e = water.e(T=lab_readings['T2'].values) -
water.e(T=lab_readings['T1'].values) # evaporator coil
    de_c = water.e(T=lab_readings['T3'].values) -
water.e(T=lab_readings['T4'].values) # condenser coil

    # energy transfer rate (W) product of energy change and mass flow
rate
    dQ_e = de_e*lab_readings['m/t e'].values*1000
    dQ_c = de_c*lab_readings['m/t c'].values*1000
    return (dQ_e,dQ_c)
```

ⓘ **The full file `heat_flux.py` can be found in the annexes and here in the working repository**

Method 3 - CoolProp

```
def method_3(lab_readings): # Method 3 CoolProp for heat transfer rate

    from CoolProp.CoolProp import PropsSI

    # specific internal energy change of the fluid across the coolant coils (J)
    de_e = PropsSI('H', 'T', lab_readings['T2'].values, 'P', 101325, 'H2O') - PropsSI('H', 'T', lab_readings['T1'].values, 'P', 101325, 'H2O') # evaporator coil
    de_c = PropsSI('H', 'T', lab_readings['T3'].values, 'P', 101325, 'H2O') - PropsSI('H', 'T', lab_readings['T4'].values, 'P', 101325, 'H2O') # condenser coil

    # energy transfer rate (W), product of energy change and mass flow rate
    dQ_e = de_e*lab_readings['m/t e'].values
    dQ_c = de_c*lab_readings['m/t c'].values

    return (dQ_e,dQ_c)
```

ⓘ **The full file `heat_flux.py` can be found in the annexes and here in the working repository**

b) Appropriately tabulate your results.

[10 marks]

ⓘ **All key values calculated are tabulated in `All Data.csv` which can be found in the annex and here in the working repository**

1. “Water - Properties vs. Temperature and Pressure.” [Online]. Available: https://www.engineeringtoolbox.com/water-properties-d_1258.html. [Accessed: 13-Nov-2024]. ↩
2. “About | PYroMat.” [Online]. Available: http://www.pyromat.org/about.html. [Accessed: 13-Nov-2024].

   ```
   >>> import pyromat as pm
   >>> water = pm.get("mp.H2O")
   >>> water.cp(p=1, T=283.15) - water.cp(p=1.5, T=323.15)
   array([0.01393411]) # The variation in c_p
   ```

   ↩
3. “About | PYroMat.” [Online]. Available: http://www.pyromat.org/about.html. [Accessed: 13-Nov-2024].

   ```
   >>> import pyromat as pm
   >>> pm.get("mp.H2O").cp(T=298.15,p=1.01325)
   array([4.1813595]) # c_p @ STP
   ```

   ↩

Question 2

a) For each flow rate in your table, determine the COP of the system.

[10 marks]

Background

coefficient of performance (COP) is a measure of performance typically associated with heat pumps and heating systems as a whole. In the case of Heat Pumps COP is the ratio of the useful heating power at the condenser per unit of

electrical input power. [1]

$$COP = \frac{Pf_{\text{Cond}}}{P_{\text{elec}}}$$

Where:
Pf_{Cond} **: Heating capacity of the condenser (**kW**)**
P_{elec} **: Input electrical power (**kW**)**

Being a critical measure of energy efficiency there are strict and standardised methodologies for testing and calculating COP[2]. On top of this manufacturer COP claims are often verified by 3rd party organisations via various certification schemes[3] like KEYMARK[4], Qlable and Eurovent[5]. This is necessary as for COP values to be a useful in communicating system performance they must be comparable, like for like with other known values.

The measures available in this experiment are not sufficient to determine of electrical input power or the heating capacity of the condenser as such any approximation of COP will not be comparable with values for heat pumps taken in accordance with the standards. The COP's calculated and discussed from this point onwards apply only to the refrigeration cycle itself, and should only be compared with values based on the same system boundaries.

Methodology

COP

This methodology defines the system boundaries at:

- **Input: work done to refrigerant by the compressor**
- **Output: work done by the refrigerant in the condenser**

Making COP a measure of:

$$COP = \frac{\dot{Q}_{\text{condensation}}}{P_{\text{compression}}}$$

Where:
$\dot{Q}_{\text{condensation}}$ **: rate of heat released by the refrigerant in the condenser**
$P_{\text{compression}}$ **: effective power of the compressor**

Under the assumption that all work done in the condenser is converted to heat, which is a reasonable approximation, Q becomes W. Further all work done to and by the refrigerant W is to be represented by the sum of it's change in enthalpy H, and time t.

$$COP = \frac{\dot{Q}_{\text{condensation}}}{P_{\text{compression}}} \triangleq \frac{\delta H_{\text{cond}} \times t}{\delta H_{\text{comp}} \times t}$$

As the the refrigerant is in an essentially closed system that has been allowed to reach a steady state, the mass flow rate is uniform through out. As such we can cancel it out from numerator and denominator to give:

$$\frac{\delta H_{\text{cond}} \times t}{\delta H_{\text{comp}} \times t} \div \frac{mt^{-1}}{mt^{-1}} = \frac{\delta h_{\text{cond}}}{\delta h_{\text{comp}}}$$

The enthalpy of the refrigerant in this case SES36 can be found based on it's temperature and pressure using a thermodynamic chart, table or model. The source used to predict the enthalpy will effect the values as in all cases they are approximations. It is important to verify that the model is sufficiently accurate in the temperature/pressure range you are working in and that it is from a reputable source. In this case the model is based on:

> **Monika Thol, Eric W. Lemmon, and Roland Span. Equation of State for a Refrigerant Mixture of R365mfc (1,1,1,3,3-Pentafluorobutane) and Galden HT 55 (Perfluoropolyether). Unpublished, 2012.**[6]

Despite being unpublished this was chosen based on recognition of one of the authors Eric W. Lemmon who is prolific in the field having developed 2 of the most predominant methodologies for identifying equations of state[7] **as well as RefProp the NIST's thermodynamic property calculator**[8]**. It was also verified by visualy comparing the P-H diagram it produces against the one included in the annexes of the R634 refrigeration cycle demonstration unit manual.pdf.**

By assuming no change in temperature enthalpy change in the condenser can be found as the difference between the enthalpy of the refrigerant SES36 at saturated vapour vs saturated liquid.
By assuming that the temperature and pressure of the refrigerant at the inlet of the compressor are the same as they where at the evaporator, and that the refrigerant is in a state of saturated vapour the initial enthalpy can be found. The final enthalpy is the same but based on the compressor discharge temperature and the condenser pressure.

This was implemented problematically using CoolProp[9] as seen below:

```python
def method_1(lab_readings):
    from CoolProp.CoolProp import PropsSI
    material = 'SES36'

    # Condenser enthalpy change
    P_initial = lab_readings['p c'].values
    T_initial = lab_readings['T6'].values

    H_initial = PropsSI('H', 'T', T_initial, 'P|gas', P_initial,
material)

    P_final = P_initial
    T_final = T_initial

    H_final = PropsSI('H', 'T', T_final, 'P|liquid', P_final,
material)

    dH_condensation = H_initial - H_final

    # Compressor ethalpy change
    P_initial = lab_readings['p e'].values
    T_initial = lab_readings['T5'].values

    H_initial = PropsSI('H', 'T', T_initial, 'P|gas', P_initial,
material)

    P_final = lab_readings['p c'].values
    T_final = lab_readings['T7'].values

    H_final = PropsSI('H', 'T', T_final, 'P|gas', P_final, material)

    dH_compression = H_initial - H_final

    cop = abs(dH_condensation/dH_compression)

    return cop
```

ⓘ The full file `cop.py` can be found in the annexes and here in the working repository

b) Plot the COP against the saturation or condensing temperature.

[10 marks]

---

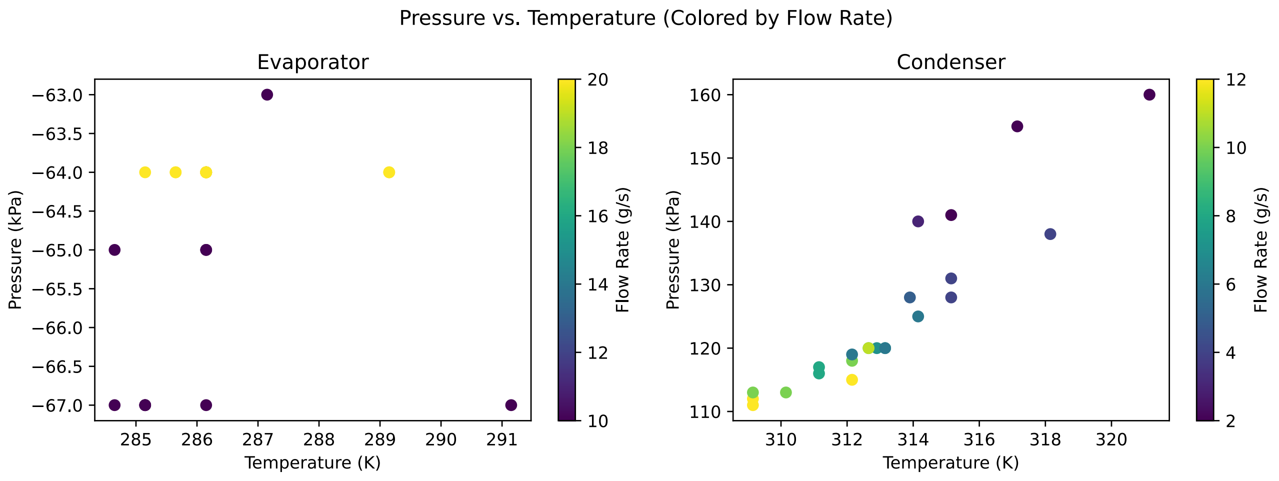
1. Certification reference standard NF 414, AFNOR NF 414, 2024. Available: https://www.eurovent-certification.com/media/images/349/3c3/3493c319d9de88e18e3ecea2156c1e8d6521a7d4.pdf ↩ ↩
2. Heating systems and water based cooling systems in buildings. Method for calculation of system energy requirements and system efficiencies. Part 4-2. Space heating generation systems, heat pump systems, BS EN 14511-3, 2018. Download ↩
3. “EHPA Certification.” [Online]. Available: https://www.ehpa.org/quality/. [Accessed: 22-Nov-2024]. ↩
4. “Testing and Certification.” [Online]. Available: https://keymark.eu/en/products/heatpumps/testing-and-certification. [Accessed: 22-Nov-2024]. ↩
5. “NF414 | Eurovent Certita Certification.” [Online]. Available: https://www.eurovent-certification.com/en/third-party-certification/certification-programmes/nf414. [Accessed: 22-Nov-2024]. ↩
6. Monika Thol, Eric W. Lemmon, and Roland Span. Equation of State for a Refrigerant Mixture of R365mfc (1,1,1,3,3-Pentafluorobutane) and Galden HT 55 (Perfluoropolyether). Unpublished, 2012. ↩
7. E. W. Lemmon and R. Tillner-Roth, “A Helmholtz energy equation of state for calculating the thermodynamic properties of fluid mixtures,” 04-Nov-1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378381299002629. [Accessed: 28-Nov-2024]. ↩
8. “Eric W. Lemmon,” 09-Oct-2019. [Online]. Available: https://www.nist.gov/people/eric-w-lemmon. [Accessed: 28-Nov-2024]. ↩
9. “SES36 — CoolProp 6.6.0 documentation.” [Online]. Available: http://www.coolprop.org/fluid_properties/fluids/SES36.html. [Accessed: 28-Nov-2024]. ↩

Question 3

a) Plot the pressure vs temperature for the evaporator and condenser at the different water flow rates.

[10 marks]

Plot



b) Discuss the trend and any other observations.

[10 marks]

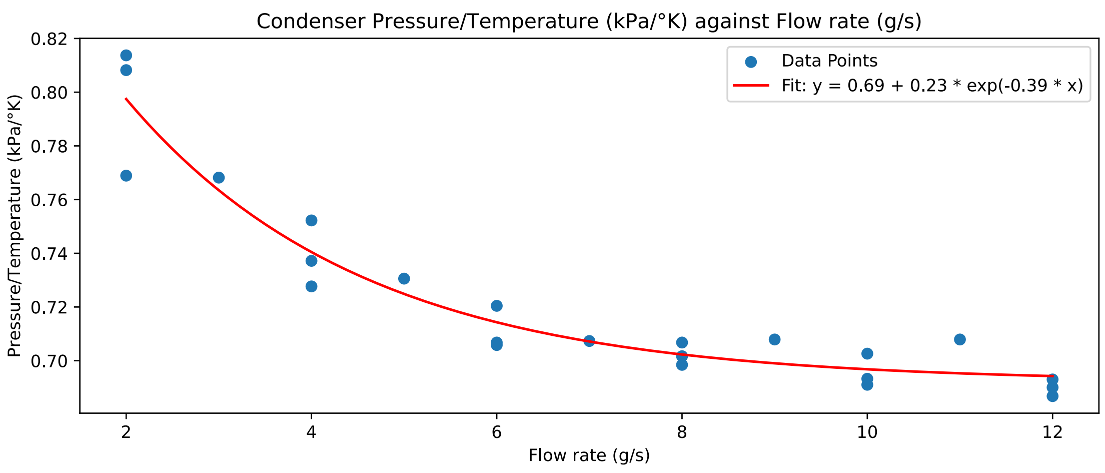
Evaporator

There is no clear trend displayed between pressure and temperature in the evaporator. That is not to say there is no relationship but from the data collected it cannot be stated either way as only 2 flow rates where tested and these did not form distinct groupings. It may be the case that if more flow rates are tested and other variables are controlled a trend might emerge.

Condenser

In the condenser clear trends can be seen. The pressure and temperature display a linear correlation with a Pearson pairwise value of 0.92. The mass flow rate of the coolant has an inverse correlation with both pressure and

temperature. By plotting Pressure divided by temperature against mass flow rate, it becomes clear the relationship is of an exponentially diminishing nature.

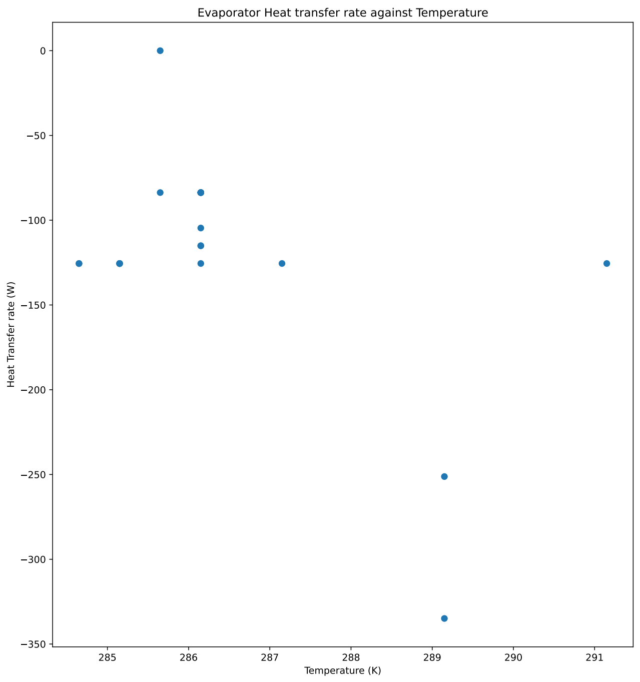


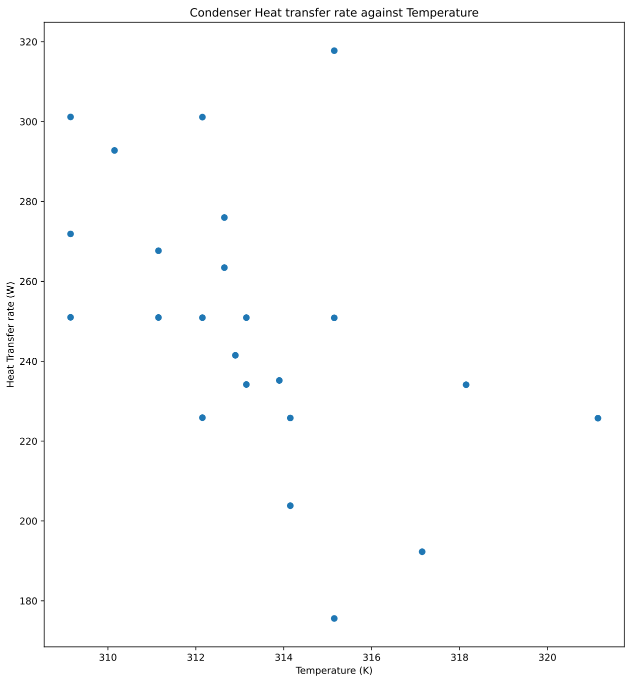
The relationship with just

Question 4

a) Plot the evaporator and condenser heat transferred against temperature

[10 marks]



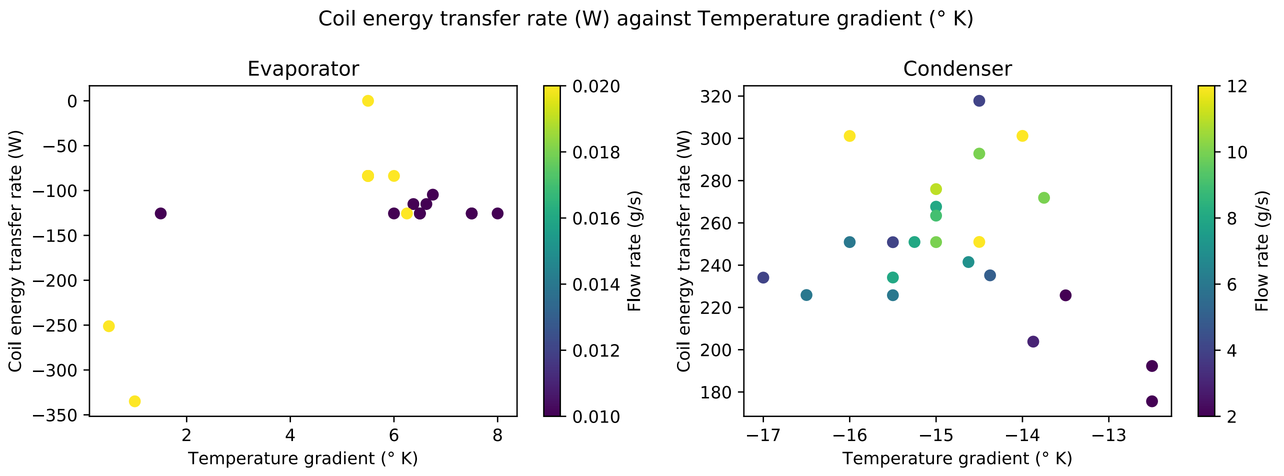


b) Discuss the trend and any other observations.

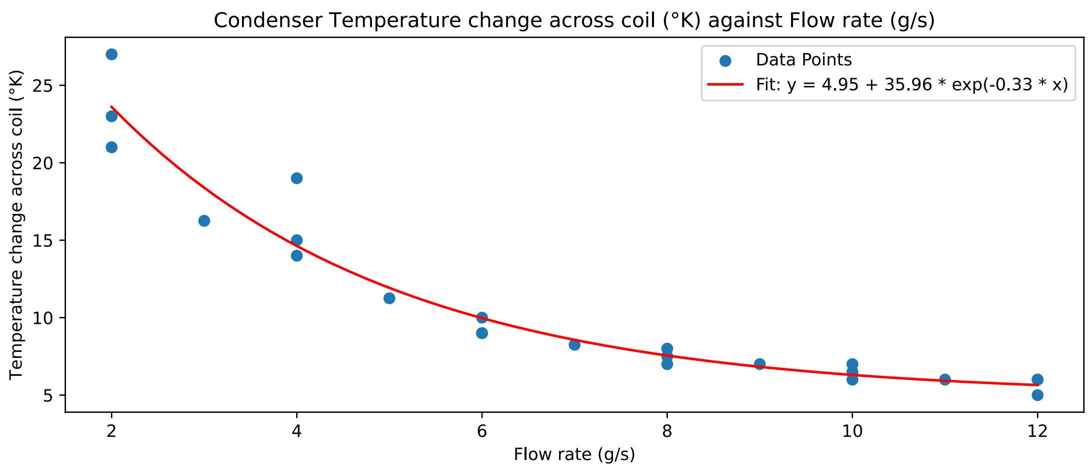
[10 marks]

On first inspection, the trends between heat transfer rate and phase change chamber temperature are not obvious in either the evaporator or condenser as they are not particularly strong, with Pearson pairwise correlations of close to -0.5. Specifically the -0.52 and -0.49 for the evaporator and condenser respectively. This is to say the data suggests a weak negative correlation between the heat transfer rate and the temperature of the phase change chamber.

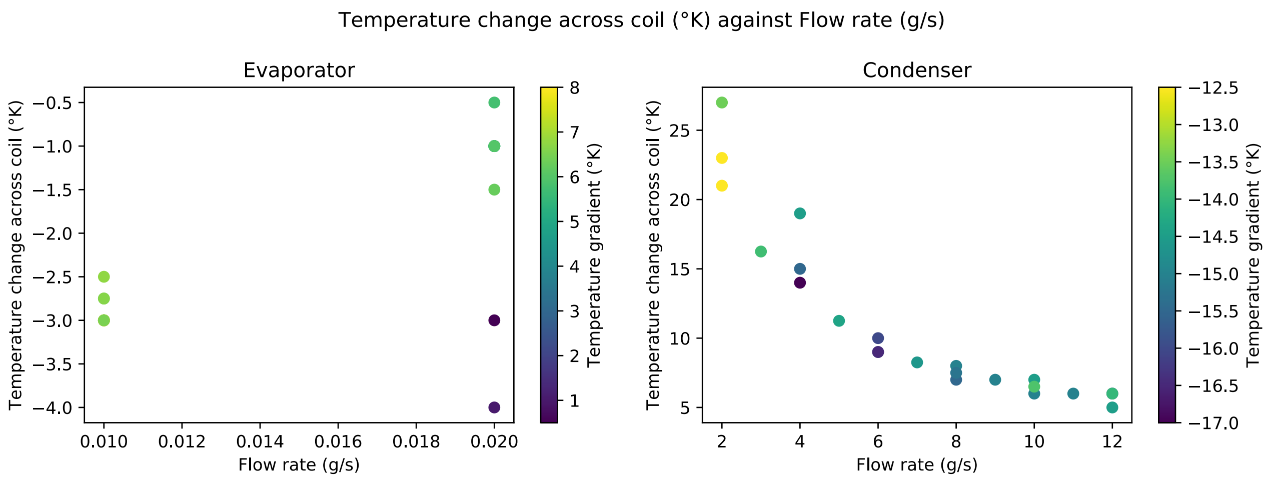
It would be intuitive to assume that the rate of heat transfer is proportional to the temperature gradient between the phase change chamber and the thermal reservoir, and that, increasing the flow rate will increase the temperature gradient. However, as can be seen below, the experimental data shows only a very loose correlation. *(note the negative axes)*



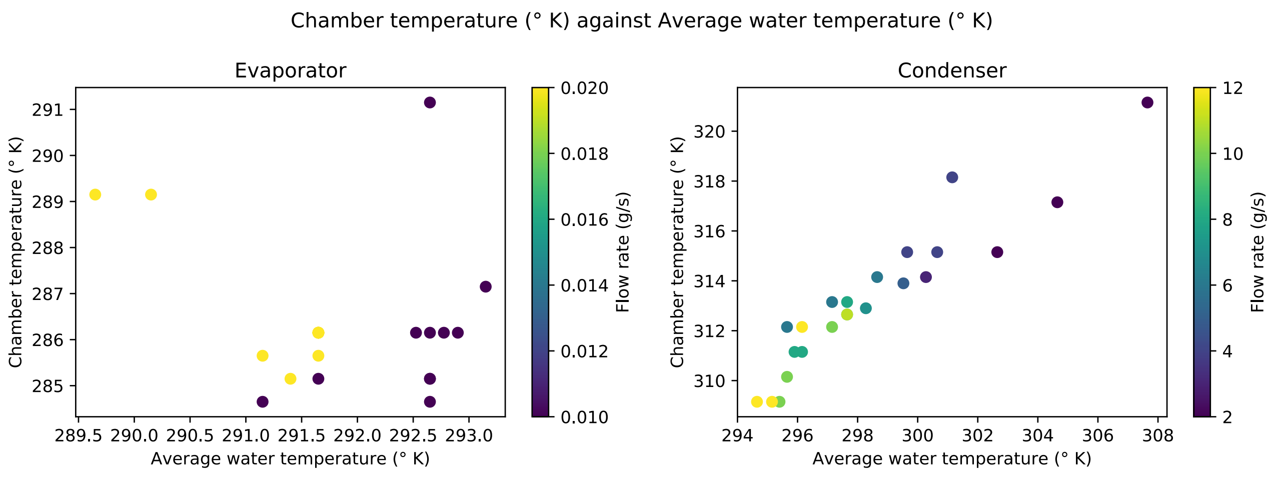
The foundational assumption is correct that increasing the flow rate minimises the temperature change across the coil. Each gram of water is being exposed to the chamber for fewer seconds and therefore absorbing fewer Joules of energy and less energy added per unit mass means less temperature change. This is confirmed by the exponentially diminishing temperature with flow rate in the condenser graph below.



However as can be seen by the colouring in the graphs below the relationship between temperature change across the coil and the temperature gradient is not as simple.



This is because as the faster flow decreases the temperature of the coolant it can absorb more energy than is being released by the refrigerant as latent heat and therefore it also decreases the temperature of the chamber. This in turn means the compressor must provide a higher pressure requiring more power.



Maximising the energy transfer is usually not the only objective. In the case of a heat pump the lower grade heat provided by a higher flow rate may be undesirable as not only dose it make the system and it's plumbing louder and more energy intensive but the output temperature might be too low to be useful.

Question 5

Explain what happens if the water supplied into the unit is cooled or warmed below and above those used in tasks 1-4 above? How will the heat transferred, and COP be affected?

[5 marks]

The water supplied to the unit is used as a thermal reservoir to absorb and release thermal energy predictably, that is to say without changing temperature. Naturally when water absorbs / releases energy in the condenser / evaporator respectively it dose change temperature but, by the fact that it is flowing, the system reaches a steady state with a constant and predictable temperature gradient. This average temperature at steady state is the effective temperature of the thermal reservoir.
In the experiment detailed in this report, the flow rate has been used as an independent variable to affect the system. This works by changing the effective temperature of the thermal reservoir and thus effecting the the rate at which it absorbs and emits energy.
If the temperature of the water supplied to the unit is change it will have a similar effect, varying the effective temperature of the thermal reservoirs i.e. the heat exchange coils in the evaporator and condenser.

Question 6

It is proposed that water should be used as the refrigerant instead of R-134a in an air conditioner

where the minimum temperature never goes below the freezing point. Would you support this proposal or not? Kindly explain.

[5 marks]

Report Requirements

Pay particular attention to detail, appropriately formatting your report with a good introduction, procedure, diagrams of experimental with appropriate labelling, discussion accompanying your answers, and uncertainty analysis.

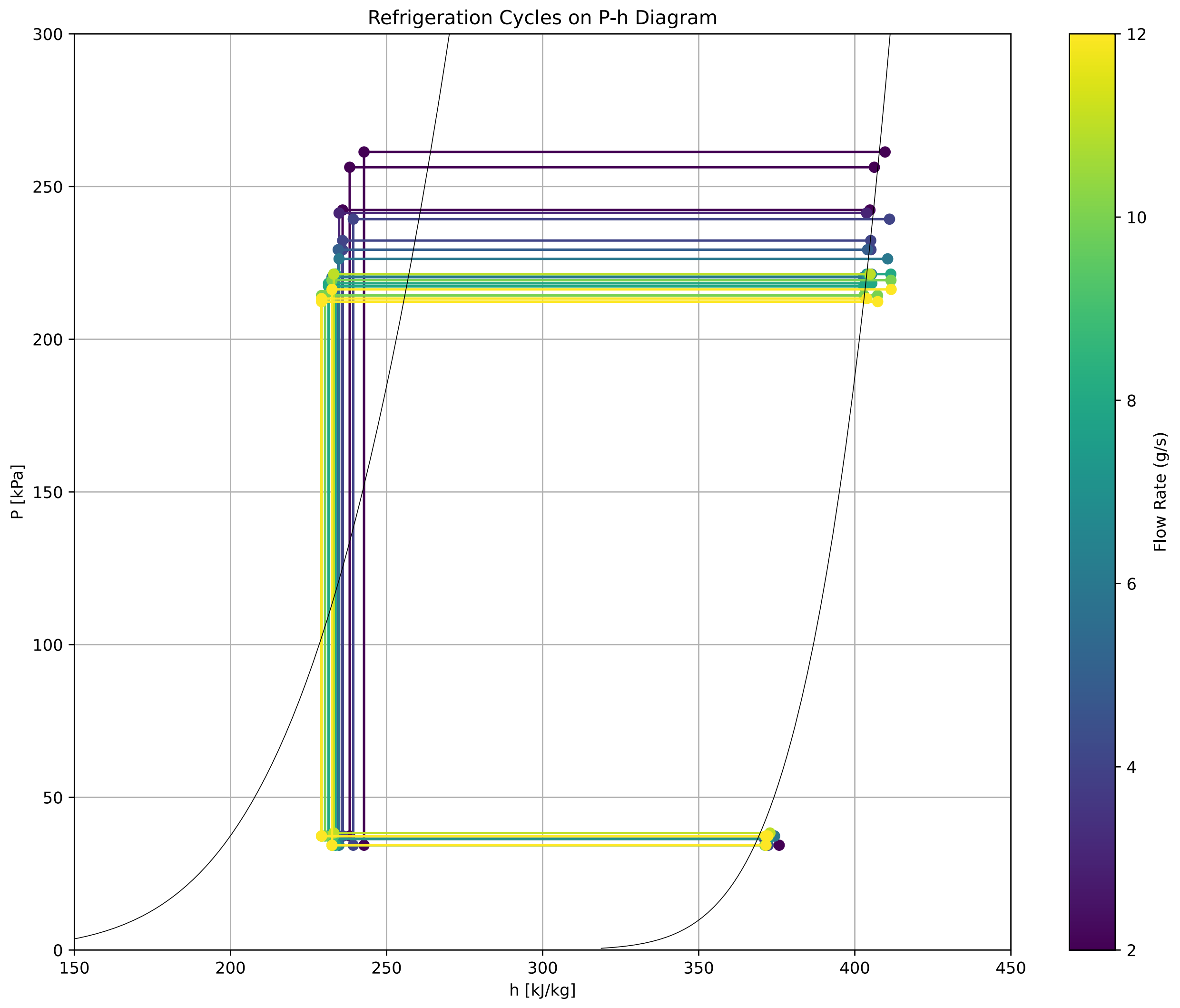
[10 marks]

Plots

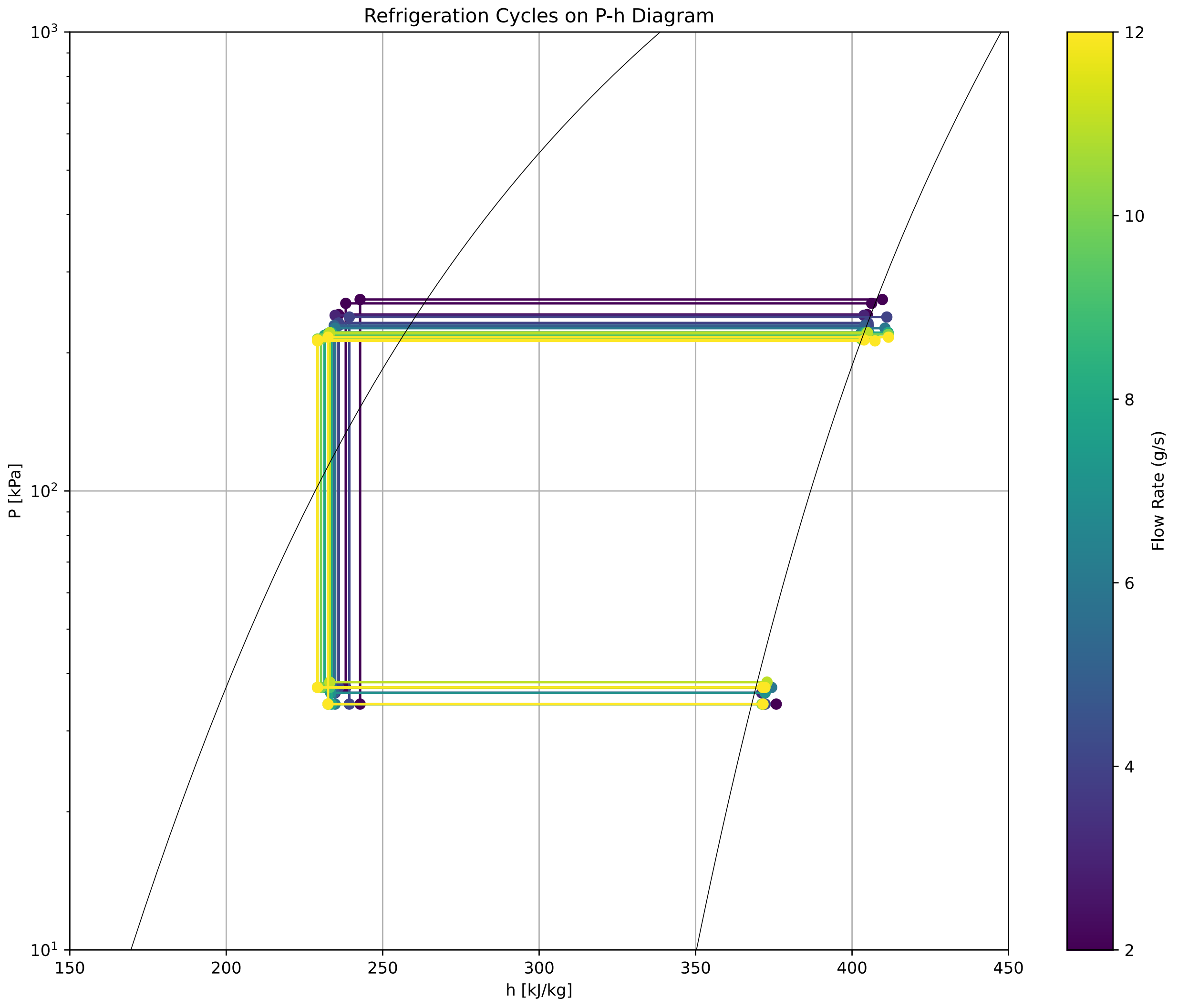
[All Plots can be seen here in the working repository]

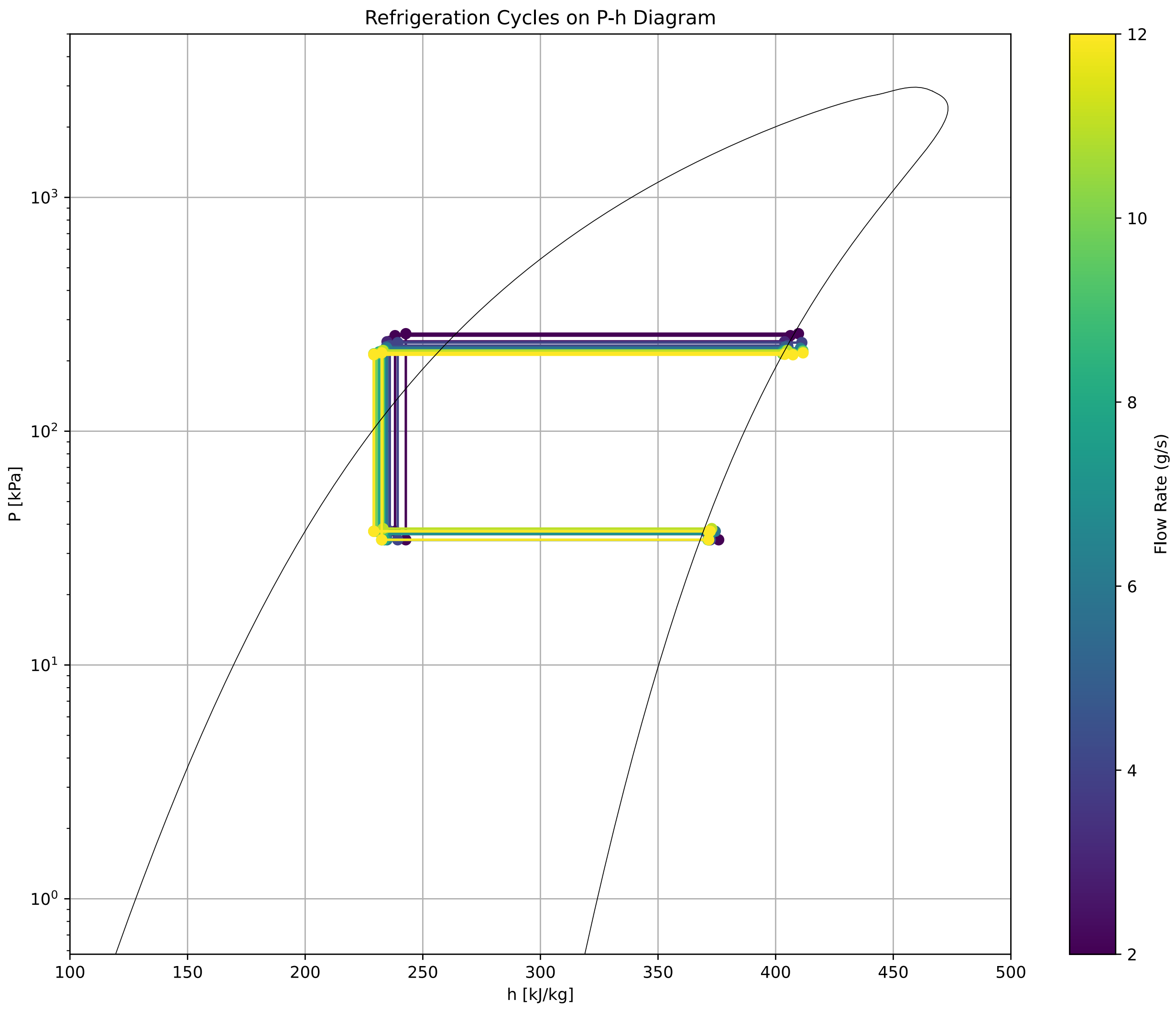
Refrigeration Cycles

Linear


Refrigeration Cycles on P-h Diagram
P [kPa]
h [kJ/kg]
Flow Rate (g/s)
0
50
100
150
200
250
300
150
200
250
300
350
400
450
2
4
6
8
10
12

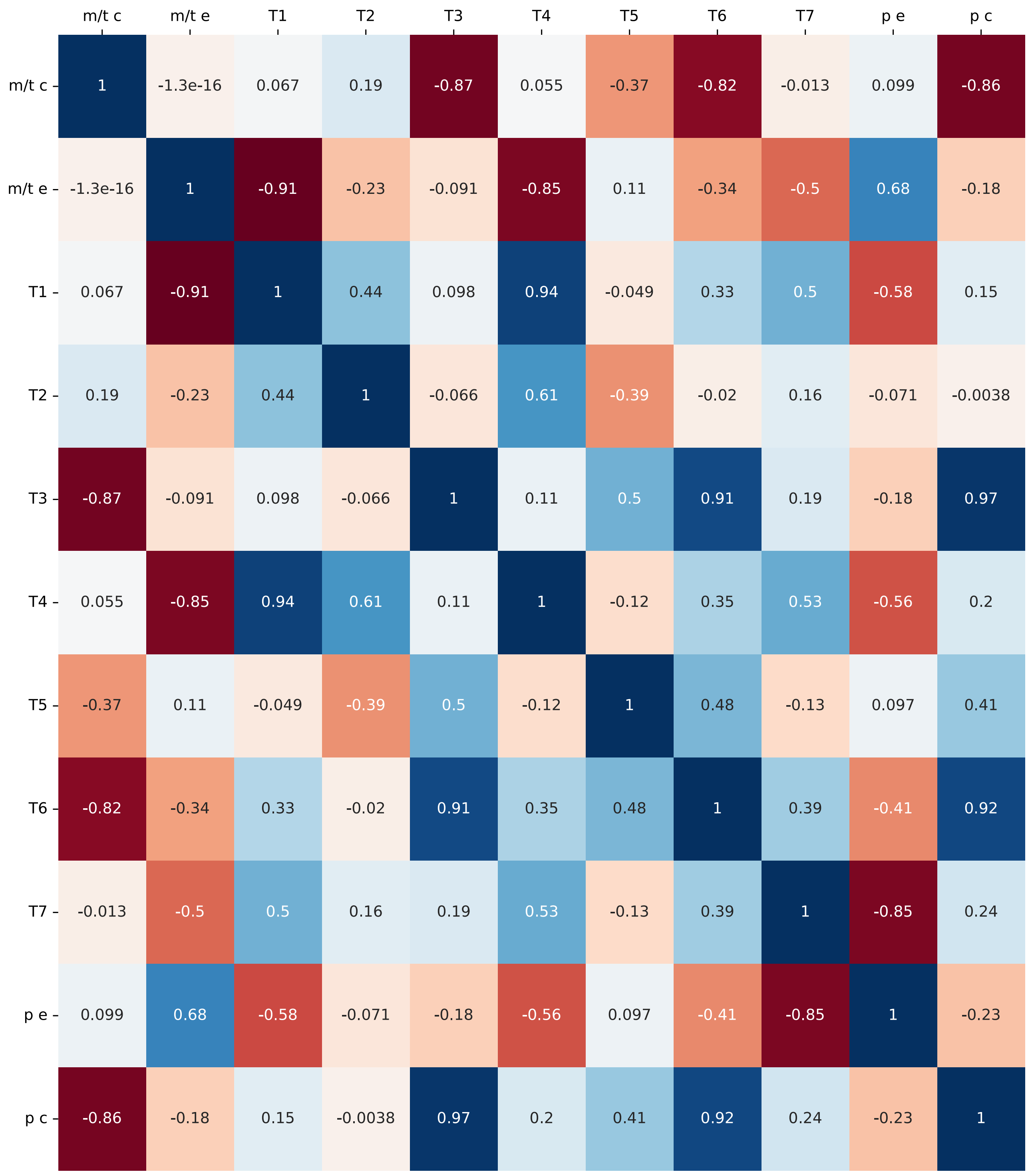
Log


Refrigeration Cycles on P-h Diagram
P [kPa]
10^3
10^2
10^1
h [kJ/kg]
150
200
250
300
350
400
450
Flow Rate (g/s)
2
4
6
8
10
12



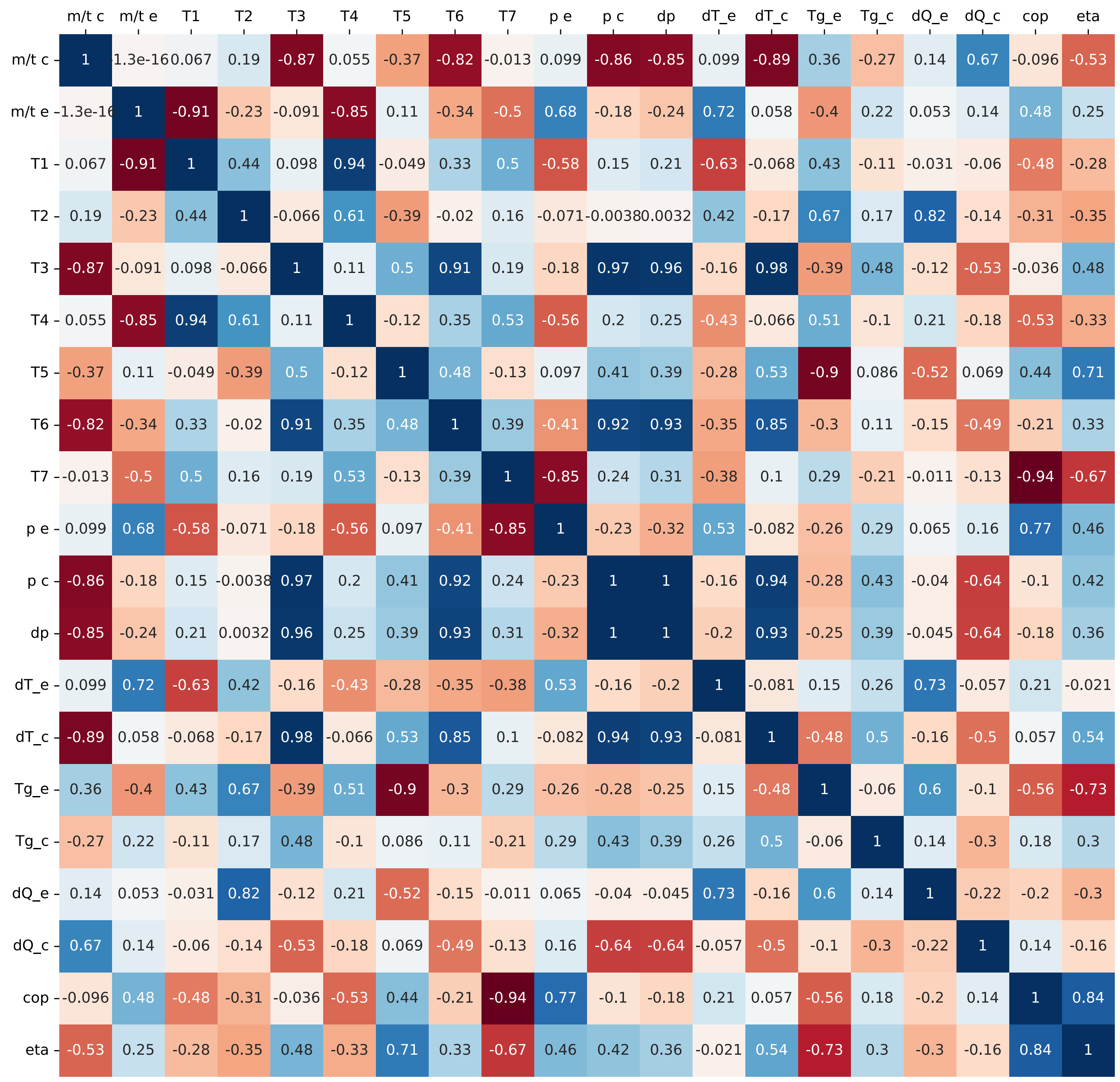
Pairwise Pearson Correlation maps

Readings



	m/t c	m/t e	T1	T2	T3	T4	T5	T6	T7	p e	p c
m/t c	1	-1.3e-16	0.067	0.19	-0.87	0.055	-0.37	-0.82	-0.013	0.099	-0.86
m/t e	-1.3e-16	1	-0.91	-0.23	-0.091	-0.85	0.11	-0.34	-0.5	0.68	-0.18
T1	0.067	-0.91	1	0.44	0.098	0.94	-0.049	0.33	0.5	-0.58	0.15
T2	0.19	-0.23	0.44	1	-0.066	0.61	-0.39	-0.02	0.16	-0.071	-0.0038
T3	-0.87	-0.091	0.098	-0.066	1	0.11	0.5	0.91	0.19	-0.18	0.97
T4	0.055	-0.85	0.94	0.61	0.11	1	-0.12	0.35	0.53	-0.56	0.2
T5	-0.37	0.11	-0.049	-0.39	0.5	-0.12	1	0.48	-0.13	0.097	0.41
T6	-0.82	-0.34	0.33	-0.02	0.91	0.35	0.48	1	0.39	-0.41	0.92
T7	-0.013	-0.5	0.5	0.16	0.19	0.53	-0.13	0.39	1	-0.85	0.24
p e	0.099	0.68	-0.58	-0.071	-0.18	-0.56	0.097	-0.41	-0.85	1	-0.23
p c	-0.86	-0.18	0.15	-0.0038	0.97	0.2	0.41	0.92	0.24	-0.23	1

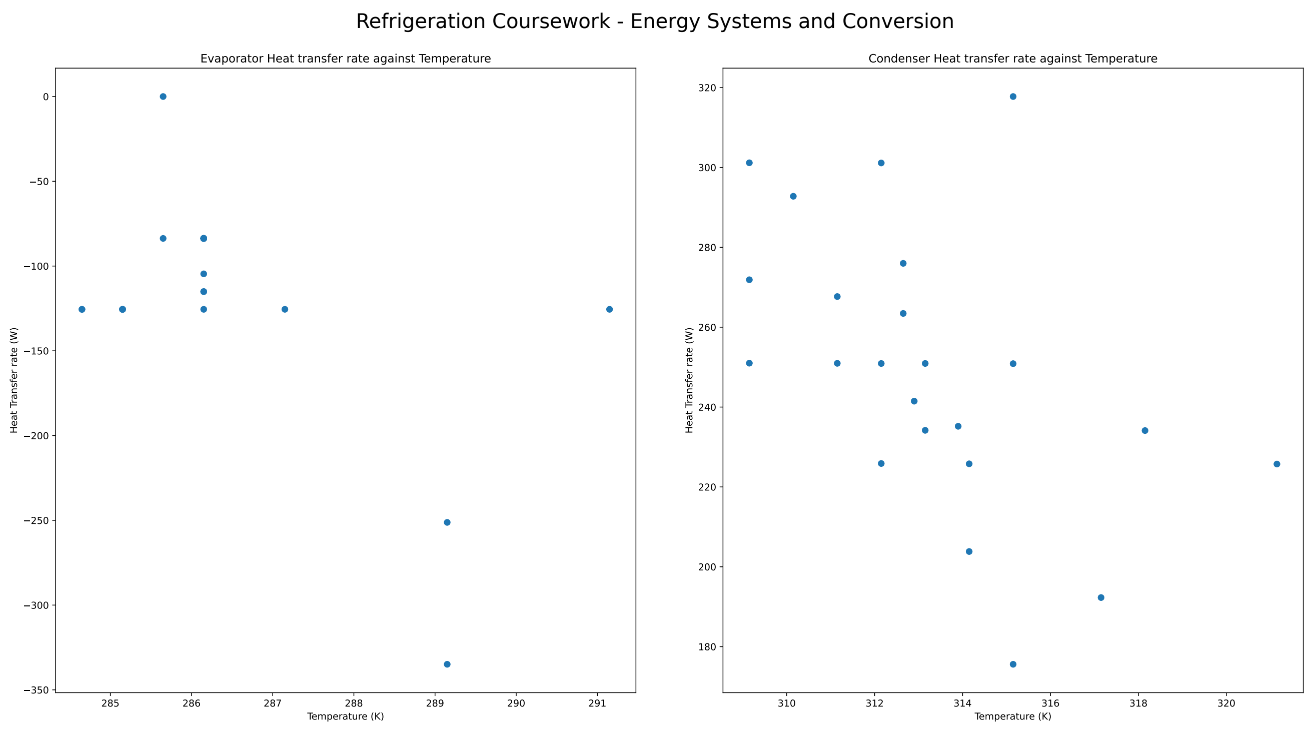
All Data

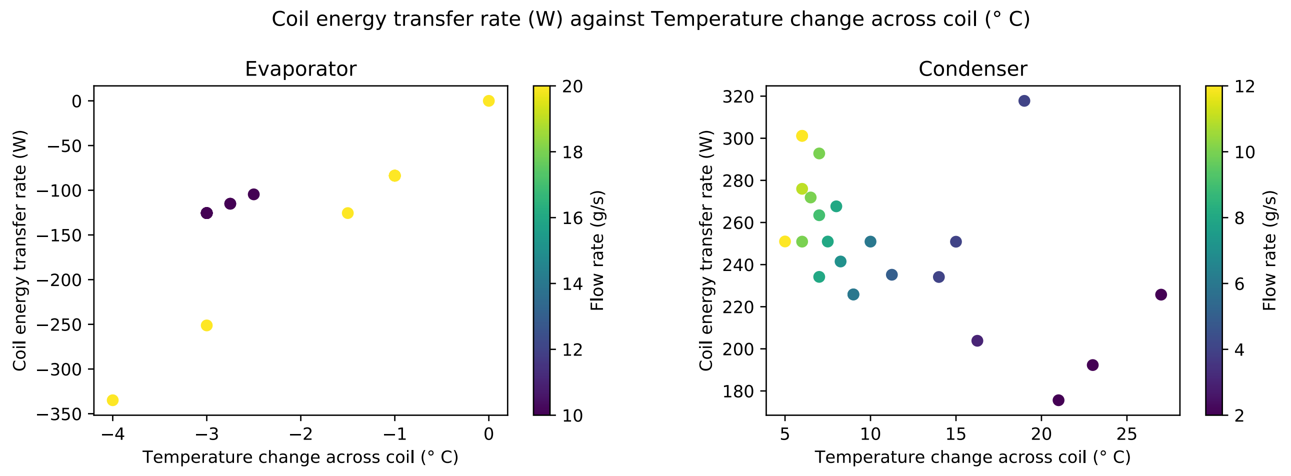


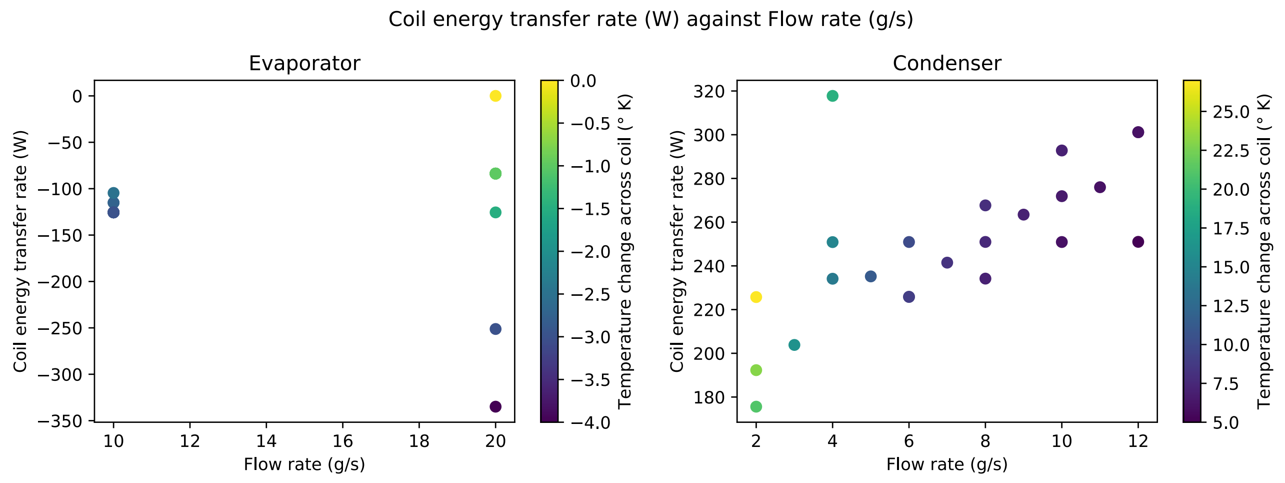
	m/t c	m/t e	T1	T2	T3	T4	T5	T6	T7	p e	p c	dp	dT_e	dT_c	Tg_e	Tg_c	dQ_e	dQ_c	cop	eta
m/t c	1	1.3e-16	0.067	0.19	-0.87	0.055	-0.37	-0.82	-0.013	0.099	-0.86	-0.85	0.099	-0.89	0.36	-0.27	0.14	0.67	-0.096	-0.53
m/t e	-1.3e-16	1	-0.91	-0.23	-0.091	-0.85	0.11	-0.34	-0.5	0.68	-0.18	-0.24	0.72	0.058	-0.4	0.22	0.053	0.14	0.48	0.25
T1	0.067	-0.91	1	0.44	0.098	0.94	-0.049	0.33	0.5	-0.58	0.15	0.21	-0.63	-0.068	0.43	-0.11	-0.031	-0.06	-0.48	-0.28
T2	0.19	-0.23	0.44	1	-0.066	0.61	-0.39	-0.02	0.16	-0.071	-0.0038	0.0032	0.42	-0.17	0.67	0.17	0.82	-0.14	-0.31	-0.35
T3	-0.87	-0.091	0.098	-0.066	1	0.11	0.5	0.91	0.19	-0.18	0.97	0.96	-0.16	0.98	-0.39	0.48	-0.12	-0.53	-0.036	0.48
T4	0.055	-0.85	0.94	0.61	0.11	1	-0.12	0.35	0.53	-0.56	0.2	0.25	-0.43	-0.066	0.51	-0.1	0.21	-0.18	-0.53	-0.33
T5	-0.37	0.11	-0.049	-0.39	0.5	-0.12	1	0.48	-0.13	0.097	0.41	0.39	-0.28	0.53	-0.9	0.086	-0.52	0.069	0.44	0.71
T6	-0.82	-0.34	0.33	-0.02	0.91	0.35	0.48	1	0.39	-0.41	0.92	0.93	-0.35	0.85	-0.3	0.11	-0.15	-0.49	-0.21	0.33
T7	-0.013	-0.5	0.5	0.16	0.19	0.53	-0.13	0.39	1	-0.85	0.24	0.31	-0.38	0.1	0.29	-0.21	-0.011	-0.13	-0.94	-0.67
p e	0.099	0.68	-0.58	-0.071	-0.18	-0.56	0.097	-0.41	-0.85	1	-0.23	-0.32	0.53	-0.082	-0.26	0.29	0.065	0.16	0.77	0.46
p c	-0.86	-0.18	0.15	-0.0038	0.97	0.2	0.41	0.92	0.24	-0.23	1	1	-0.16	0.94	-0.28	0.43	-0.04	-0.64	-0.1	0.42
dp	-0.85	-0.24	0.21	0.0032	0.96	0.25	0.39	0.93	0.31	-0.32	1	1	-0.2	0.93	-0.25	0.39	-0.045	-0.64	-0.18	0.36
dT_e	0.099	0.72	-0.63	0.42	-0.16	-0.43	-0.28	-0.35	-0.38	0.53	-0.16	-0.2	1	-0.081	0.15	0.26	0.73	-0.057	0.21	-0.021
dT_c	-0.89	0.058	-0.068	-0.17	0.98	-0.066	0.53	0.85	0.1	-0.082	0.94	0.93	-0.081	1	-0.48	0.5	-0.16	-0.5	0.057	0.54
Tg_e	0.36	-0.4	0.43	0.67	-0.39	0.51	-0.9	-0.3	0.29	-0.26	-0.28	-0.25	0.15	-0.48	1	-0.06	0.6	-0.1	-0.56	-0.73
Tg_c	-0.27	0.22	-0.11	0.17	0.48	-0.1	0.086	0.11	-0.21	0.29	0.43	0.39	0.26	0.5	-0.06	1	0.14	-0.3	0.18	0.3
dQ_e	0.14	0.053	-0.031	0.82	-0.12	0.21	-0.52	-0.15	-0.011	0.065	-0.04	-0.045	0.73	-0.16	0.6	0.14	1	-0.22	-0.2	-0.3
dQ_c	0.67	0.14	-0.06	-0.14	-0.53	-0.18	0.069	-0.49	-0.13	0.16	-0.64	-0.64	-0.057	-0.5	-0.1	-0.3	-0.22	1	0.14	-0.16
cop	-0.096	0.48	-0.48	-0.31	-0.036	-0.53	0.44	-0.21	-0.94	0.77	-0.1	-0.18	0.21	0.057	-0.56	0.18	-0.2	0.14	1	0.84
eta	-0.53	0.25	-0.28	-0.35	0.48	-0.33	0.71	0.33	-0.67	0.46	0.42	0.36	-0.021	0.54	-0.73	0.3	-0.3	-0.16	0.84	1

Evaporator and Condenser Comparisons

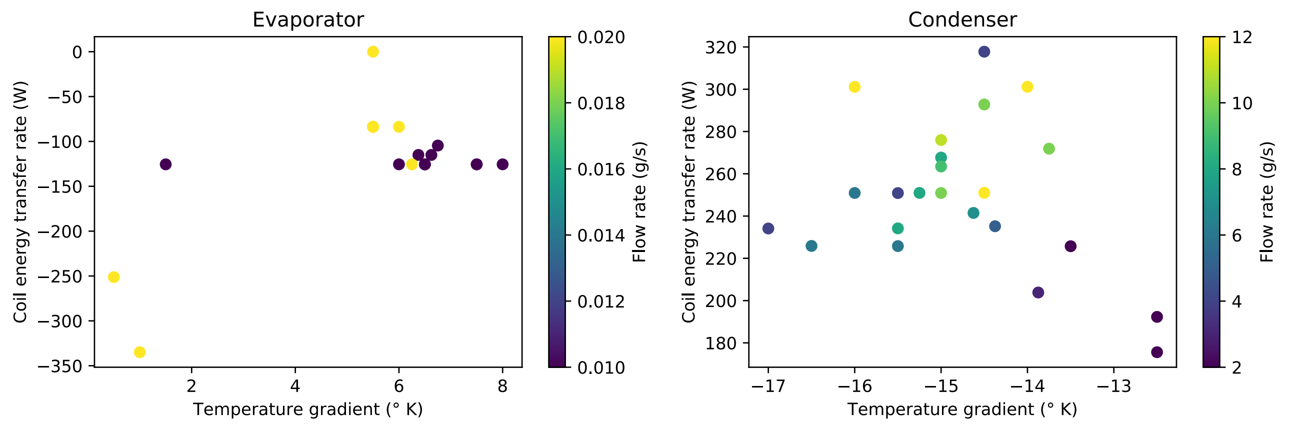
Coolant heat transfer rate


Refrigeration Coursework - Energy Systems and Conversion
Evaporator Heat transfer rate against Temperature
Heat Transfer rate (W)
Temperature (K)
Condenser Heat transfer rate against Temperature
Heat Transfer rate (W)
Temperature (K)


Coil energy transfer rate (W) against Temperature change across coil (° C)
Evaporator
Coil energy transfer rate (W)
Temperature change across coil (° C)
Flow rate (g/s)
Condenser
Coil energy transfer rate (W)
Temperature change across coil (° C)
Flow rate (g/s)


Coil energy transfer rate (W) against Flow rate (g/s)
Evaporator
Coil energy transfer rate (W)
Flow rate (g/s)
Temperature change across coil (° K)
Condenser
Coil energy transfer rate (W)
Flow rate (g/s)
Temperature change across coil (° K)

Coil energy transfer rate (W) against Temperature gradient (° K)



Spreadsheets

All SpreadSheets can be found here in the working repository

Lab_Readings

m/t c (24)	m/t e	T1	T2	T3	T4	T5	T6	T7
3	10	19.5	16.5	35.25	19	11.5	41	60.75
5	10	20.75	18	32	20.75	13	40.75	60.75
7	10	21	18.25	29.25	21	13	39.75	60.25
9	10	21	18.5	28	21	13	39.5	61
11	10	21.5	18.5	27.5	21.5	14	39.5	61.5
2	10	21	18	48	21	18	48	68
4	10	21	18	35	21	13	45	69
6	10	21	18	30	21	12	41	68
8	10	21	18	28	21	12	40	69
10	10	21	18	27	21	11.5	39	69
12	10	20	17	26	20	12	39	69
2	20	19	18	40	19	13	42	62
4	20	19	15	37	18	16	42	62
6	20	18	15	27	18	16	39	59
8	20	19	18	27	19	13	38	59
10	20	19	18	26	19	13	37	59

m/t c (24)	m/t e	T1	T2	T3	T4	T5	T6	T7
12	20	19	18	25	19	13	36	60
2	20	19	18	43	20	13	44	64
4	20	19	18	34	19	13	42	62
6	20	18.5	18	29	19	12.5	40	62
8	20	19	18	26.5	19	13	38	62
10	20	19	18	25.5	19	12.5	36	64
12	20	19	17.5	24	19	12	36	64

All_Data

(24)	m/t c	m/t e	T1	T2	T3	T4	T5	T6
0	0.003	0.01	292.65	289.65	308.4	292.15	284.65	314.15
1	0.005	0.01	293.9	291.15	305.15	293.9	286.15	313.9
2	0.007	0.01	294.15	291.4	302.4	294.15	286.15	312.9
3	0.009	0.01	294.15	291.65	301.15	294.15	286.15	312.65
4	0.011	0.01	294.65	291.65	300.65	294.65	287.15	312.65
5	0.002	0.01	294.15	291.15	321.15	294.15	291.15	321.15
6	0.004	0.01	294.15	291.15	308.15	294.15	286.15	318.15
7	0.006	0.01	294.15	291.15	303.15	294.15	285.15	314.15
8	0.008	0.01	294.15	291.15	301.15	294.15	285.15	313.15
9	0.01	0.01	294.15	291.15	300.15	294.15	284.65	312.15
10	0.012	0.01	293.15	290.15	299.15	293.15	285.15	312.15
11	0.002	0.02	292.15	291.15	313.15	292.15	286.15	315.15
12	0.004	0.02	292.15	288.15	310.15	291.15	289.15	315.15
13	0.006	0.02	291.15	288.15	300.15	291.15	289.15	312.15
14	0.008	0.02	292.15	291.15	300.15	292.15	286.15	311.15
15	0.01	0.02	292.15	291.15	299.15	292.15	286.15	310.15
16	0.012	0.02	292.15	291.15	298.15	292.15	286.15	309.15
17	0.002	0.02	292.15	291.15	316.15	293.15	286.15	317.15
18	0.004	0.02	292.15	291.15	307.15	292.15	286.15	315.15
19	0.006	0.02	291.15	291.15	302.15	292.15	285.65	313.15

(24)	m/t c	m/t e	T1	T2	T3	T4	T5	T6
20	0.008	0.02	292.15	291.15	299.65	292.15	286.15	311.15
21	0.01	0.02	292.15	291.15	298.65	292.15	285.65	309.15
22	0.012	0.02	292.15	290.65	297.15	292.15	285.15	309.15

Finding Thermodynamic Properties

Motivation

There are many means of identifying fluid properties ranging in their ease and accuracy. In my university Energy Systems and Conversion Module we are taught to seek out property tables which to my 21st century mind seems an arcane practice, impractical at scale and prone to human error. As such I set out to understand the alternatives.

Brief

Search for means to query thermodynamic properties for fluids, that suits my preferences.
Preferably the solution would be (in order of priority):

1. Open Source / Free / Easy to pirate
2. Reasonably well documented
3. A python package
4. Simple to make queries with
5. Have a large library of material models / datasets
6. Be expandable to "custom" models

Options

CoolProp / PyFluids

http://www.coolprop.org/

Condition	Rating	
Open Source / Free / Easy to pirate	+++++	Source

Condition	Rating	
Reasonably well documented	++++	Docs
A python package	+++++	Wrapper
Simple to make queries with	+++	
Have a large library of material models / datasets	+++++	
Be expandable to "custom" models	+++++	

This is a FOSS C++ implementation of REFPROP functionality both high and low level. It has wrappers for anything your likely to want to use and the documentation while a bit of nest isn't half bad.

Pyromat

http://pyromat.org/

Condition	Rating	
Open Source / Free / Easy to pirate	+++++	Source
Reasonably well documented	+++++	Docs
A python package	+++++	
Simple to make queries with	+++++	
Have a large library of material models / datasets	++	
Be expandable to "custom" models	-	

Simple and lightweight, 90% of the time this could be my go to.

Further notes on PYroMat

REFPROP

https://www.nist.gov/srd/refprop

Found by reading this excellent lecture from the NIST. Seems like the real deal, if NASA's using it.

Condition	Rating	
Open Source / Free / Easy to pirate	-	Source
Reasonably well documented		Docs

Condition	Rating	
A python package	+++	Wrapper
Simple to make queries with		
Have a large library of material models / datasets	+++++	
Be expandable to "custom" models		

https://trc.nist.gov/refprop/MINIREF/MINIREF.HTM

EES

https://fchartsoftware.com/ees/

Is a general engineering software package with some thermodynamics property querying functionality based on REFPROP. It comes with some materials (List of fluid models) and can import NISTs .fld format for thermodynamic properties.

Condition	Rating	
Open Source / Free / Easy to pirate	-	Licence
Reasonably well documented	-	Youtube
A python package	-	
Simple to make queries with		
Have a large library of material models / datasets	-	
Be expandable to "custom" models	++	

Code

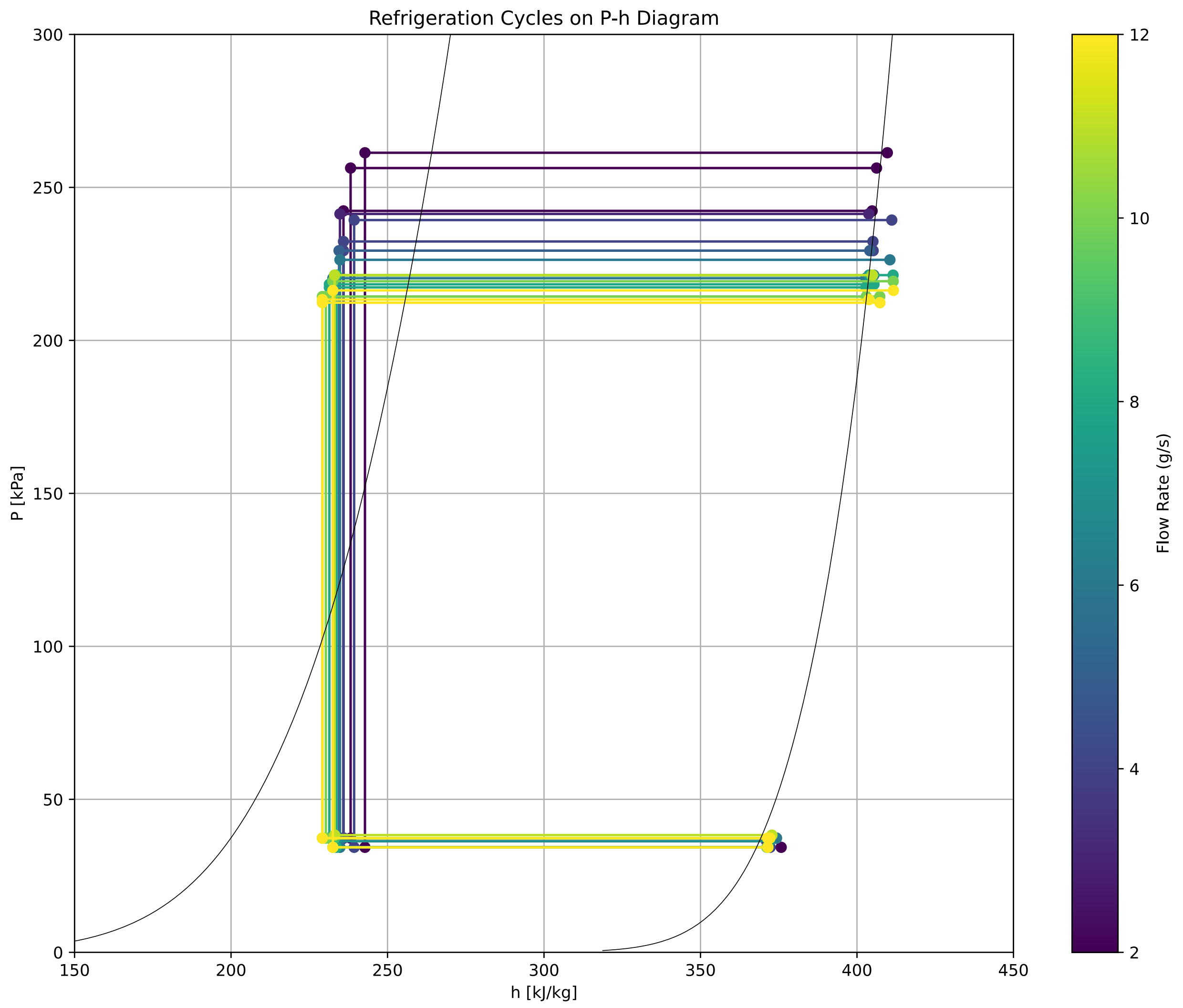
All of the code can be viewed here in the working repository

plot_cycles.py

View here in the working repository

Plotting Cycles over SES36 Iso Lines on a PH diagram

For some more intuition It's worth seeing what the cycles look like, below are idealised approximations of what all the cycles would have been for the flow rates tested.

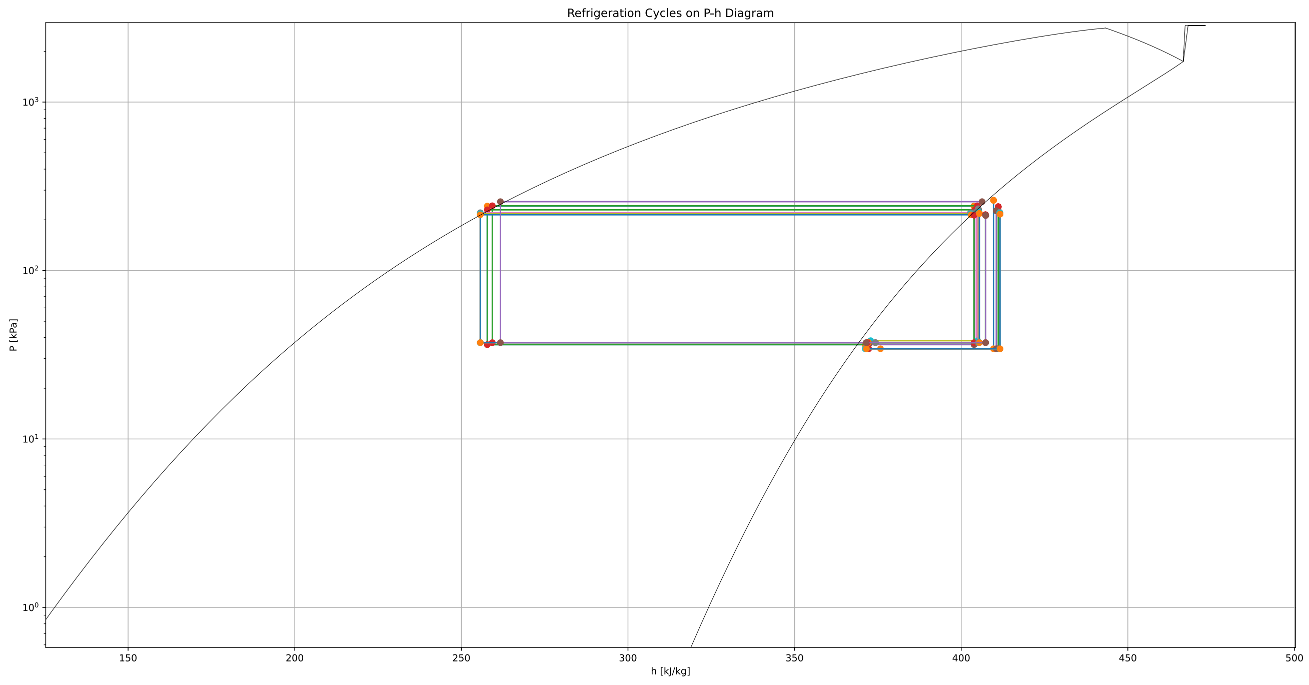


To achieve this I wrote the `plot_PH` function that can be found in plot_cycles.py. It's based loosely on this example I found on stack overflow[1] utilising my `isentropic_efficiency` function mentioned above along with with the SimpleCyclesCompression module from coolProp[2]. It is part of the cop.py file that can be seen in it's entirety in the annexes or here in the repo.

```
def plot_PH(lab_readings):
    import CoolProp.CoolProp as CP
    from CoolProp.Plots import PropertyPlot
    from CoolProp.Plots import SimpleCompressionCycle
    import matplotlib.pyplot as plt

    T_evap = lab_readings["T5"].values
    P_evap = lab_readings["p e"].values
    T_comp = lab_readings["T7"].values
    P_comp = lab_readings["p c"].values
    eta_com = isentropic_efficiency(T_evap,P_evap,T_comp,P_comp)

    ph_plt = PropertyPlot('SES36', 'PH', unit_system='KSI')

    ph_plt.xlabel("h [kJ/kg]")
    ph_plt.ylabel("P [kPa]")

    ph_plt.calc_isolines(CP.iQ)

    cycle = SimpleCompressionCycle("SES36", "PH",
unit_system="KSI")

    print(CP.PropsSI('Phase', 'T', T_evap, 'P', P_evap, "SES36"))
    print(CP.PropsSI('Phase', 'T', T_comp, 'P', P_comp, "SES36"))
    print(CP.PhaseSI('T', T_evap[0], 'P', P_evap[0], "SES36"))
    print(CP.PhaseSI('T', T_comp[1], 'P', P_comp[1], "SES36"))

    for entry in zip(T_evap,P_evap,T_comp,P_comp, eta_com):
        cycle.simple_solve(*entry)
        sc = cycle.get_state_changes()
        ph_plt.draw_process(sc)

    ph_plt.title("Refrigeration Cycles on P-h Diagram")
    ph_plt.grid()
    ph_plt.show()
```

Issues

I'm having some trouble getting the `SimpleCOmpressionCycle.simple_solver` method to complete the cycles. It will only plot the compression stage points provided failing to find the values for the liquid states.



Note that only 7 of the 23 cycles are complete.

If I add a few lines to check the predicted state of the refrigerant at the values indicated for compressor inlet and outlet, pressure and temperature:

```
T_evap = lab_readings["T5"].values
P_evap = lab_readings["p e"].values
T_comp = lab_readings["T7"].values
P_comp = lab_readings["p c"].values

print(CP.PropsSI('Phase', 'T', T_evap, 'P', P_evap, "SES36"))
print(CP.PropsSI('Phase', 'T', T_comp, 'P', P_comp, "SES36"))
```

It ouputs the following for;

- Inlet: `[5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]`
- Outlet: `[0. 0. 5. 5. 5. 5. 5. 5. 5. 5. 5. 0. 5. 0. 0. 0. 5. 0. 5. 5. 5. 5. 5.]`

where `5` indicates gas, and `0` indicates liquid. Interestingly also 7 instances where the outlet phase was predicted to be liquid. Of course the pump can't be dealing with phase changes else there would be cavitation and all sorts of issues.

I wrote a quick function to check how much I would have to change the values to push them all into liquid phase:

```python
def check_state(lab_readings):
    import CoolProp.CoolProp as CP

    T_evap = lab_readings["T5"].values
    P_evap = lab_readings["p e"].values
    T_comp = lab_readings["T7"].values
    P_comp = lab_readings["p c"].values

    print(CP.PropsSI('Phase', 'T', T_evap, 'P', P_evap, "SES36"))
    print(CP.PropsSI('Phase', 'T', T_comp, 'P', P_comp, "SES36"))
    print(CP.PhaseSI('T', T_evap[0], 'P', P_evap[0], "SES36"))
    print(CP.PhaseSI('T', T_comp[1], 'P', P_comp[1], "SES36"))

    # Check difference for temperature adjustment
    print(T_comp)
    print(CP.PropsSI('Phase', 'T', T_comp, 'P', P_comp, "SES36"))
    T_comp_sat = (CP.PropsSI('T', 'Q', 1.0, 'P', P_comp,
"SES36"))
    print(T_comp_sat)
    print(CP.PropsSI('Phase', 'T', T_comp_sat, 'P', P_comp,
"SES36"))

    print (T_comp-T_comp_sat)

    # Check difference for temperature adjustment
    print(P_comp)
    print(CP.PropsSI('Phase', 'T', T_comp, 'P', P_comp, "SES36"))
    P_comp_sat = (CP.PropsSI('P', 'Q', 1.0, 'T', T_comp,
"SES36"))
    print(P_comp_sat)
    print(CP.PropsSI('Phase', 'T', T_comp, 'P', P_comp_sat,
"SES36"))

    print (P_comp-P_comp_sat)
```

and this is what it spat out:

- For outlet Temperature:
  - Current values are:
    ```
    [333.9 333.9 333.4 334.15 334.65 341.15 342.15 341.15 342.15
    342.15 342.15 335.15 335.15 332.15 332.15 332.15 333.15
    337.15 335.15 335.15 335.15 337.15 337.15]
    ```

> - Adjusted values would be:
>   ```
>   [336.32859836 334.58300919 333.3785215 333.3785215
>   333.3785215 339.09075948 336.04252383 334.13530266
>   333.3785215 333.07199452 332.60800168 336.47093127
>   334.58300919 333.22553503 332.76323261 332.29581558
>   332.13885048 338.4163316 335.02608424 333.3785215
>   332.91789529 332.29581558 331.9812973 ]
>   ```
> - Possible with a adjustments of:
>   ```
>   [-2.42859836 -0.68300919 0.0214785 0.7714785 1.2714785
>   2.05924052 6.10747617 7.01469734 8.7714785 9.07800548
>   9.54199832 -1.32093127 0.56699081 -1.07553503 -0.61323261
>   -0.14581558 1.01114952 -1.2663316 0.12391576 1.7714785
>   2.23210471 4.85418442 5.1687027 ]
>   ```

- For Outlet Pressure:

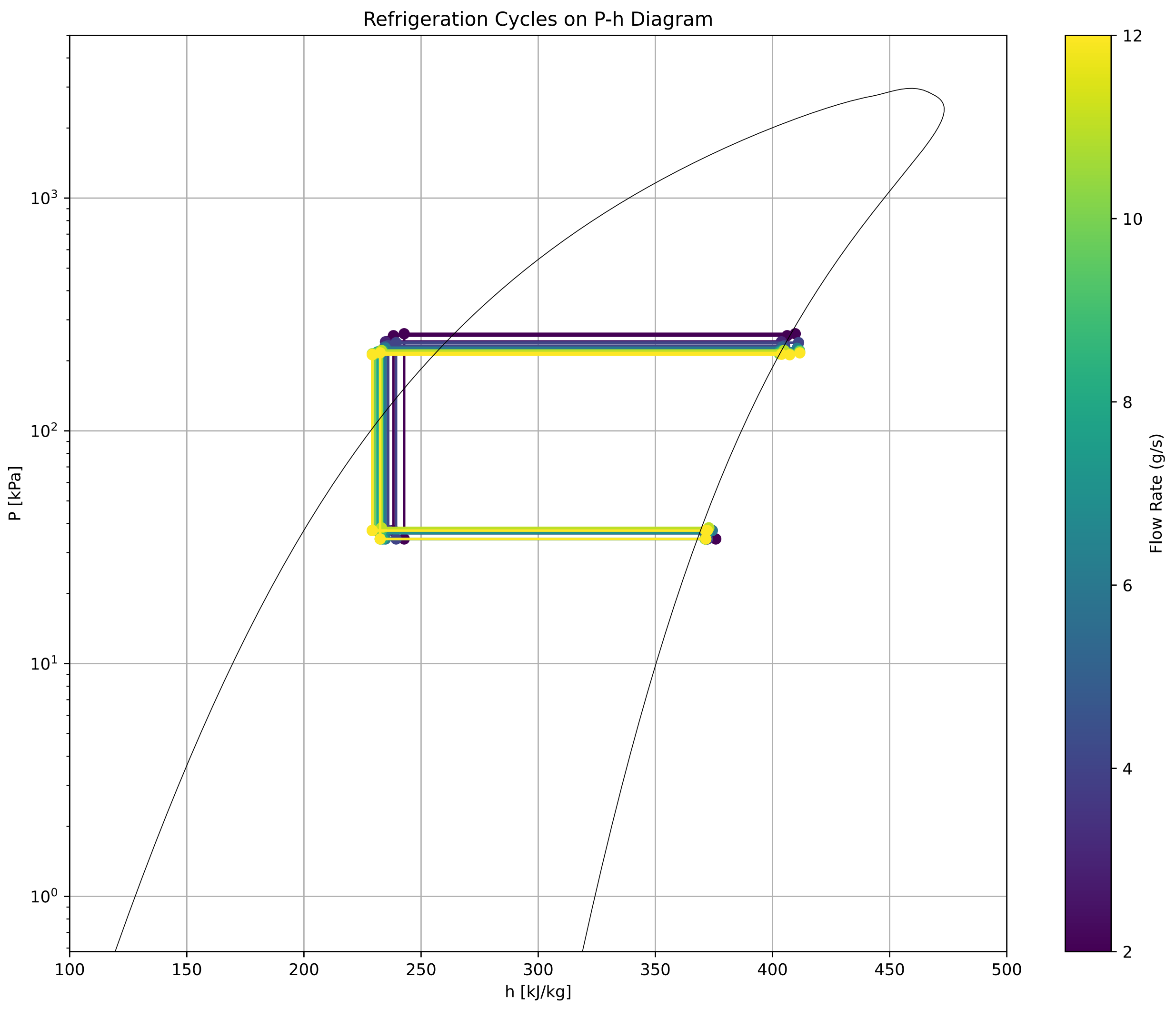
> - Current values are:
>   ```
>   [241325 229325 221325 221325 221325 261325 239325 226325
>   221325 219325 216325 242325 229325 220325 217325 214325
>   213325 256325 232325 221325 218325 214325 212325]
>   ```
> - Adjusted values would be:
>   ```
>   [224760.84184103 224760.84184103 221465.68359062
>   226422.98611947 229776.59909676 277069.49076882
>   284979.85730193 277069.49076882 284979.85730193
>   284979.85730193 284979.85730193 233169.57068941
>   233169.57068941 213395.90881022 213395.90881022
>   213395.90881022 219832.59212109 247141.29339878
>   233169.57068941 233169.57068941 233169.57068941
>   247141.29339878 247141.29339878]
>   ```
> - Possible with a adjustments of:
>   ```
>   [ 16564.15815897 4564.15815897 -140.68359062 -5097.98611947
>   -8451.59909676 -15744.49076882 -45654.85730193
>   -50744.49076882 -63654.85730193 -65654.85730193
>   -68654.85730193 9155.42931059 -3844.57068941 6929.09118978
>   3929.09118978 929.09118978 -6507.59212109 9183.70660122
>   -844.57068941 -11844.57068941 -14844.57068941
>   -32816.29339878 -34816.29339878]
>   ```

Based on this I've figured that instead of using the compressor discharge temperature I should use the condenser temperature so that the little supercooling between will push it past the critical point.

As I cant get it to complete the cycles I'm going to sacrifice the compression & sub-cooling stages as this is likely the least accurate. This allows me to have better plots for condensation, expansion and evaporation.

I also added:

- cubic interpolation with numpy to fix the top of the isolines.
- colour cycles by flow rate
- colour bar legend
- linear pressure axis option
- large labels for readability



---

1. “Put labels in Coolprop Chart.” [Online]. Available: https://stackoverflow.com/questions/70864726/put-labels-in-coolprop-chart. [Accessed: 26-Nov-2024]. ↩

2. “CoolProp.Plots.SimpleCyclesCompression module — CoolProp 6.6.0 documentation.” [Online]. Available: http://www.coolprop.org/apidoc/CoolProp.Plots.SimpleCyclesCompression.html. [Accessed: 26-Nov-2024]. ↩