Numerical analysis of the thermal performance of vapour compression heat pump heat exchanger using Python and Computational Fluids Dynamics (CFD)

Prepared by:

Sehobai Elliot Sehobai

SHBSEH001

Department of Mechanical Engineering

University of Cape Town

Supervisor:

Prof. Tunde Bello-Ochende

October 2023

Submitted to the Department of Mechanical Engineering at the University of Cape Town in fulfilment of the academic requirements for a Master of Science degree in Mechanical Engineering

Abstract

Numerical analysis on fin and tube heat exchangers contributes towards the implementation of energy-efficient technologies in the industrial and building sectors. Fin and tube heat exchangers are found in various mechanical applications including heating, ventilation, and air conditioning (HVAC) and refrigeration systems, the oil and gas extraction industry, power plants and many more.

Due to the rapid depletion of energy resources worldwide, there is a need to reduce energy consumption, especially for systems that use electricity such as heat pump systems. This led to several studies on the heat exchangers used in heat pumps including analyses of the heat exchanger geometry and working fluid impacts on the thermal performance.

This study presents numerical analyses on the fin and tube heat exchanger model developed in Python, using nonuniform airflow velocities calculated in Ansys Fluent. The geometrical parameters of the modelled heat exchanger are based on the literature values. The heat transfer rates, pressure losses, vapour quality and all refrigerant properties are calculated by discretizing each tube on each tube circuit and tube row into several increments and incorporating nonuniform airflow in three dimensional. The model is validated using experimental data which shows that the maximum variation between the model and experimental results is less than 10.0%.

The velocity contours from the Ansys Fluent heat exchanger model suggest that airflow varies significantly in three dimensional. The results from the modelled heat exchanger in Python show that the nonuniformity of airflow consequently affects the refrigerant pressure losses, heat transfer and vapour quality in the refrigerant tubes. Thus, assuming uniform airflow over the heat exchanger results in underestimating the actual refrigerant pressure losses, heat transfer and vapour quality in the upper refrigerant tube circuits (those located closer to the top of the heat exchanger) while overestimating these parameters on lower tube circuits (those located towards the bottom, farther from the fan location). This leads to a maximum variation exceeding 10.0%.

Moreover, the coefficient of performance (COP) was also calculated from the heat pump model developed in Python. These model results suggest that generally, assuming uniform airflow on the heat exchanger underpredicts the heat pump COP by a maximum variation of 11,07% for all four operating conditions of the heat pump discussed in this study. These

results highlight the importance of performing analysis in three-dimensional space, considering non-uniform airflow.

Key Words: energy use, heat transfer rate, thermal efficiency, heat exchanger, heat pump system, coefficient of performance (COP).

Plagiarism Declaration

I, **Sehobai Elliot Sehobai**, hereby declare that the work on which this dissertation is based is my original work (except where acknowledgements indicate otherwise) and that neither the whole work nor any part of it has been, is being, or is to be submitted for another degree in this or any other university. I authorise the University to reproduce for the purpose of research either the whole or any portion of the contents in any manner whatsoever.

Signature: Signed by candidate Date: 27/10/2023

Acknowledgements

First and foremost, I would like to express my gratitude to God, whose unwavering presence and guidance have been the cornerstone of my life's journey. Without His grace and blessings, none of this would have been possible.

I express my deep appreciation to my supervisor, Prof. Tunde Bello-Ochende, for his expert guidance, unwavering patience, and constant encouragement. His mentorship has been instrumental in shaping the direction of this research and helping me navigate the intricacies of the subject matter.

I am indebted to the ATProM group, with special thanks to Mr. Gosai Priyesh, for fostering an environment conducive to learning and exploration. His dedication to academic excellence and willingness to offer assistance whenever needed has been truly invaluable.

A special mention is reserved for Miss Denise Botha, Mechanical Engineering departmental postgraduate administrator, for her administrative support and contributions that have streamlined the research process. I also extend my sincere appreciation to Dr. Tanimu Jatau for his invaluable technical advice, which significantly enriched the quality of this dissertation.

My heartfelt appreciation goes to my family, particularly my mother Arciliah Sehobai, for her unwavering support, understanding, and belief in my abilities. Your sacrifices and encouragement have been a driving force behind my pursuit of higher education.

To my beloved wife Grace, your unwavering faith in me and your understanding during the long hours of work and study has been my greatest source of strength. Your love and encouragement have made this journey not only possible but also meaningful.

I would like to extend my gratitude to all my friends and colleagues who have shared their knowledge, insights, and experiences with me. As well as everyone who has indirectly influenced and supported me during my academic pursuit. Your discussions and feedback have enriched my understanding and improved the quality of this dissertation.

In conclusion, I dedicate this dissertation to the memory of my beloved father, Karabo Phineas Sehobai, whose unwavering support and profound wisdom continue to inspire me, even though he is no longer with us. His legacy of resilience, determination, and love has

been a guiding light throughout this journey. This work is a tribute to his enduring influence on my life and my academic pursuits.

Contents

List of Figures

List of Tables

Nomenclature

Symbol	Description
T_{db}	Dry bulb temperature [℃]
T_{wb}	Wet-bulb temperature [℃]
T_{amb}	Ambient temperature [℃]
T_{sw}	Wall surface temperature [℃]
T_{air}	Air temperature [℃]
T_{ref}	Refrigerant temperature [℃]
T_{evap}	Evaporating temperature [℃]
ΔT	Change in temperature [℃]
$\dot{Q}_{sen}$	Total sensible heat load [kW]
$\dot{Q}_{lat}$	Total latent heat load [kW]
$\dot{Q}_{ref}$	Rate of heat transfer by the refrigerant [kW]
$\dot{Q}_{air}$	Rate of heat transfer by air [kW]
$\dot{Q}_{evap}$	Rate of heat transfer in the evaporator [kW]
$\Delta\dot{Q}_{w}$	Change in the rate of heat transfer by water film [kW]
$\Delta\dot{Q}_{air}$	Change in the rate of heat transfer by air stream [kW]
SHR	Sensible heat ratio
RH	Relative humidity
$\dot{m}_{ref}$	Mass flow rate of refrigerant [kg/s]
$\dot{m}_{H2O}$	The mass flow rate of water [kg/s]
$\dot{m}_{air}$	The mass flow rate of air [kg/s]
$\Delta\dot{m}_{w}$	Rate of change of water film [kg/s]
Δh	Change in specific enthalpy [J/kgK]
h_{gw}	Specific enthalpy of saturated gas at vapour partial pressure [J/kgK]
h_{g}	Specific enthalpy of saturated gas [J/kgK]
h_{f}	Specific enthalpy of saturated liquid [J/kgK]
h_{fgw}	Specific enthalpy of vaporization [kJ/kg]
COP	Coefficient of performance
C_p	Specific heat capacity at constant pressure [J/kgK]

w_{air}	Humidity ratio of air [kg/kg dry air]
w_{sw}	Humidity ratio of air on the tube wall surface [kg/kg dry air]
Δw_{air}	Change in humidity ratio of air [kg/kg dry air]
V_{air}	Air velocity [m/s]
u_x	Velocity component in x-axis [m/s]
u_y	Velocity component in y-axis [m/s]
u_z	Velocity component in z-axis [m/s]
$\boldsymbol{u}$	Velocity vector [m/s]
A_{af}	Heat exchanger frontal area [m^2]
$A_{ht_{air}}$	Fin side heat transfer area [m^2]
$A_{ht_{ref}}$	Tube side heat transfer area [m^2]
A_{fin}	The total area of fins [m^2]
A_{tot}	The total surface area of the heat exchanger air side [m^2]
A_i	Evaporator tubes cross-sectional area [m^2]
A_{ff}	The free flow area of the heat exchanger [m^2]
V_{tot}	The total volume of the heat exchanger [m^3]
D_o	The outer diameter of evaporator tubes [m]
D_h	Flow passage hydraulic diameter of the finned-tube evaporator [m]
htc_{air}	Fin side convective heat transfer coefficient [W/m^2K]
htc_{ref}	Tube side convective heat transfer coefficient [W/m^2K]
L_e	Lewis's factor
R_f	Fouling factor
R_o	The outer radius of the heat exchanger tubes [m]
R_i	The inner radius of the heat exchanger tubes [m]
Δp	Pressure drops [kPa]
UA	Overall heat transfer coefficient [W/K]
f_{DW}	Darcy-Weisbach friction factor
f_F	Fanno friction factor
j_H	Colburn j factor
K	Thermal conductivity [W/mK]

Nu	Nusselt number
Pr	Prandtl number
Re	Reynolds number
H	Operating condition (heating) of the heat pump system
incr	Tube increment
ρ_{air}	Air density [kg/m^2]
ρ_{ref}	Refrigerant density [kg/m^2]
α	Heat transfer area/total volume of the heat exchanger [m^2/m^3]
σ	Free flow area/heat exchanger frontal area
μ	Dynamic viscosity [Ns/m^2]
ε	Heat exchanger effectiveness
	Surface Roughness [m]
η_{air}	Airside surface efficiency
η_{fin}	Fin efficiency

1. Introduction

1.1 Background to the study

According to the Environmental group Greenpeace (2022), Africa's contribution to global greenhouse emissions, primarily through fossil fuel burning has had a profound impact on climate change, acid rain, groundwater quality, and human health. In addition, Africa contributes 1 to 1.5% of global greenhouse emissions due to the usage of coal to generate electricity [1].

As a consequence of fossil fuel usage, the world is shifting toward renewable sources of energy through the implementation of relevant policies and legislation. Thus, to decrease the world's reliance on fossil fuels and promote sustainability, one important strategy is the adoption of energy-efficient technologies. This not only contributes to the conservation of natural resources but also leads to significant long-term cost savings [2].

Energy-efficient technologies can be implemented in various divisions such as the industrial, transportation, agriculture, and building sectors including both commercial and residential. Therefore, this study investigates the optimisation of residential heat pumps as one of these energy-efficient technologies to promote decarbonization by reducing energy use in the building sector.

1.2 Problem statement

About 40% of energy consumption accounts for the building sector worldwide, and this consumption leads to approximately one-third of total greenhouse emissions worldwide [3]. HVAC systems consume approximately half of the energy utilised in buildings [4].

This study will therefore concentrate on the reduction of energy use in buildings by performing numerical analysis on the heat pump systems integrated within HVAC systems. That is, this study addresses some of the Sustainable Development Goals (SDGs), particularly SDG 12: ensuring sustainable consumption and production patterns, that is, optimization of heat pumps will help in reducing the amount of energy utilized by the building sector and SDG 13: taking urgent action to combat climate change and its impact. Thus, the study argues for efficient use of energy in heat pumps as a way of reducing carbon prints on the environment, as detailed by The 2030 Agenda for Sustainable Development [5].

1.3 The aim of the study

The primary focus of the study lies in conducting numerical analyses on heat pump systems to identify avenues for enhancing their energy efficiency. Through this investigation, the aim is to develop strategies and recommendations that can be implemented to minimize energy usage without compromising the comfort or functionality of the building environment. By optimizing heat pump systems, the study aims to achieve tangible reductions in energy consumption within the building sector.

1.4 Objectives

The objectives of the study are therefore to:

- determine the air side nonuniform airflow (in 3D) impacts on the pressure drop, heat transfer coefficient, exit refrigerant temperature and heat transfer rate of the fin and tube heat exchanger,
- discretize each tube segment by segment (increase resolution/for accuracy) to investigate how heat transfer rate and pressure drop vary from segment to segment along the length of each tube of the heat exchanger and,
- investigate the impacts of uniform airflow assumption on the thermal performance of the heat pump system,
- Assess thermodynamic performance of heat pump system.

1.5 Research questions

The study aims to answer the following questions:

1.5.1 Primary research question

How can non-uniform airflow on the fin and tube heat exchanger airside affect heat transfer and hence the Coefficient of Performance (COP) of the heat pump system?

1. 5.2 Secondary research question

How can the thermal performance of the fin and tube heat exchanger of the vapour compression heat pump be improved for better energy efficiency and system cost-effectiveness?

1.6 Plan of development

This document comprises six chapters including appendices and Chapter 1 describes the problem statement and the significance of the study. Chapter 2 analyses the present

literature on heat pump system optimisations and the limitations associated with all the implementations applied regarding heat pumps.

Chapter 3 discusses the methodology used to model all the tools developed to produce results. The use of Python to develop both the heat exchanger and heat pump cycle are explained in this chapter as well as the theory or fundamentals of heat transfer and fluid mechanics applied in the model.

Chapter 4 compares the results generated from the heat exchanger and heat pump system models developed in Python with the known results in the literature. Chapter 5 discusses the velocity profiles generated from the Ansys Fluent as well as results obtained from both heat exchanger and heat pump system Python models. Chapter 6 concludes by addressing the research questions, and these conclusions are based on all the findings from the study. The chapter also provides some recommendations on future related work that is not included in this study.

2. Literature Review

2.1 Energy efficiency in building technologies

High percentages of energy consumption in buildings have prompted researchers to focus on energy-efficient technologies in the building sector. This emphasis is particularly noticeable in areas such as domestic hot water generation and HVAC systems, where energy use is substantial. Among the various technologies explored, heat pump systems have emerged as a promising solution for enhancing energy efficiency in domestic water heating, space heating, and HVAC systems [6]. These heat pump systems leverage renewable electricity to provide thermal energy and contribute to the decarbonization of energy usage.

2.2 Understanding heat pump systems

A heat pump is an electrically powered machine that transfers thermal energy against the natural flow of heat, that is, the transfer of heat from a higher temperature to a location of a lower temperature. It comprises four essential components, namely: evaporator, compressor, condenser, and an expansion or throttling valve. A working fluid circulates within these components and facilitates the transfer of heat.

There are two classifications of heat pump systems, known as ground-source and air-source heat pumps. As the names suggest, in a ground-source heat pump, the ground is used as the source of heat transferred via the working fluid while in an air-source heat pump, the air is used as the source of heat [7].

Air-source heat pump systems are dominant worldwide in space heating, domestic water heating, and HVAC systems due to their advantages of high energy efficiency, reduction of greenhouse gas emissions, and economic viability when compared with other conventional heating systems which consume fossil fuels such as coal and natural gas to produce thermal energy [8]. Figure 1 illustrates how air source heat pump systems work in typical air conditioning of a building under heating conditions.

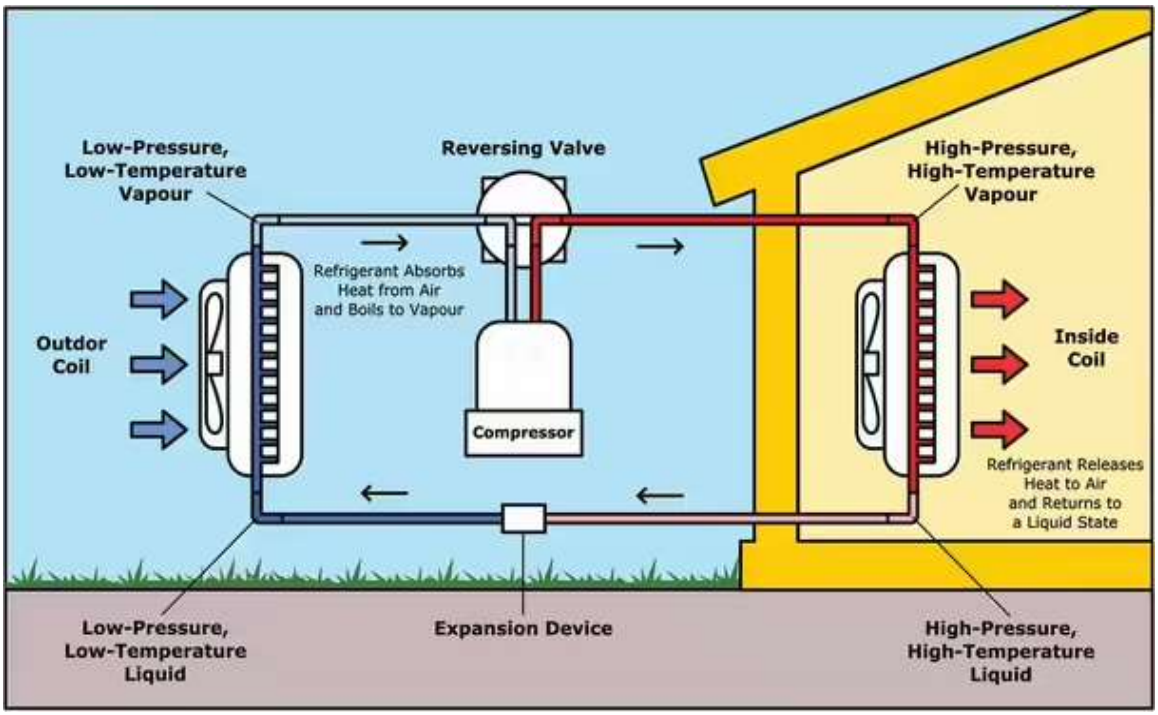


Figure 1: Illustration of how a heat pump works in air conditioning [9]

Figure 1 also shows four main components of the heat pump system, namely: condenser, compressor, expansion valve and evaporator. It also depicts how these components are integrated into each other to form a heat pump system.

2.3 Integration of renewable energy sources in heat pump systems

In the realm of energy-efficient solutions and sustainable heating and cooling, heat pumps play a pivotal role, often being seamlessly integrated with renewable sources of energy such as wind and solar. This integration extends to various applications such as space heating, domestic water heating, and air conditioning systems, contributing significantly to enhanced energy efficiency and the vital goal of decarbonization. Furthermore, the synergy between heat pump systems and thermal energy storage allows for the effective storing of excess energy during periods of low energy demand, further optimizing their performance.

For instance, Lei et al. [10] investigated the combination of vapour compression heat pump, absorption heat pump, solar collectors, wind turbine, and thermal energy storage for space heating. In this set-up, the vapour compression heat pump extracts heat from the surrounding air and then transfers it onto the absorption heat pump, which then uses its working fluid to transfer heat into the water in the storage tank. Lithium-Bromide is used as the working fluid in the absorption heat pump. The wind turbine and solar collectors produce electricity that is used to power the compressor in the vapour compression heat

pump and heat the water in the thermal storage tank. The usage of solar and wind for electricity generation, therefore, leads to a significant reduction of fossil fuel usage and hence a reduction in carbon emissions from the building sector.

Moreover, in the study by Leonforte and Miglioli [11], they conducted an in-depth analysis of a residential photovoltaic-thermal solar-assisted heat pump system, focusing on its design and performance monitoring. This prototype involves the seamless integration of photovoltaic-thermal solar collectors with an air-source heat pump. The integration serves a dual purpose: it assists in providing the necessary electrical energy required by the heat pump compressor while simultaneously harnessing solar energy through the collectors to enhance the working fluid's thermal energy. By leveraging solar radiation, this system effectively raises the COP of the heat pump, making it a highly efficient and sustainable energy solution.

The solar-assisted heat pump system can be classified into direct-expansion and indirect-expansion systems [12]. In a direct-expansion solar-assisted heat pump, the solar collector forms part of the evaporator, that is, the evaporator extracts heat from both the air and solar simultaneously. In an indirect-expansion system, the solar collector and evaporator are separated, and each has its own working fluid [13]. The choice between these systems depends on the specific application requirements and efficiency considerations.

There are a few parameters that affect the amount of thermal energy obtained in solar-assisted heat pump systems for space heating and domestic water heating. These include solar collector surface area exposed to the sun, angle of inclination of a solar collector, thermal storage tank volume, volume-area ratio, space heating, and domestic water heating loads as well as the control system [14]. Therefore, optimising these parameters is essential to maximise the overall performance and energy efficiency of solar-assisted heat pump systems for space heating and domestic water heating applications.

In all heat pump systems that are assisted by renewable sources of energy, the availability of both electrical and thermal energy from solar collectors, and electrical energy from a wind turbine depends on the climatic conditions, that is, solar irradiation and wind speed, respectively [15]. Therefore, thermal energy storage is required to store excess thermal energy in low-demand and high-supply periods to ensure sustainability and reliability in heating systems.

Sahin and Adiguzel [16] observed that heat pump system performance is largely affected by ambient temperature and humidity. Their experiment showed that the higher the relative humidity, the more exergy, and energy losses are incurred in heat pump systems, especially in the evaporators. Therefore, it becomes difficult to obtain cooling and heating from an air-source heat pump when there is a large amount of moisture in the air atmosphere hence resulting in high electrical energy consumption by the heat pump. This shows that the heat pump system efficiency is dependent on climatic conditions, particularly temperature and moisture.

Thermal energy storage consists of a heat exchanger that can be in various geometries, depending on the thermal properties of the phase change material to be used. Phase change materials are mostly used in thermal energy storage due to their ability to either absorb or release sufficient energy during phase change, providing required cooling or heating. However, phase change materials have a low heat transfer coefficient, that is; they require a lengthy period to absorb and release thermal energy [17]. Therefore, their relatively low heat transfer coefficient necessitates careful design and consideration of factors such as heat exchanger geometry and operating conditions to optimise their efficiency.

Ye and Li [18] also investigated air-source heat pump systems that are used for space heating and domestic water heating purposes, whereby the thermal energy storage tank is coupled with the condenser to increase the heat pump performance. This is achieved by wrapping around the thermal storage tank with condenser tubes to increase the heat transfer area between the condenser and the hot water storage tank. The performance of the heat pump water heater and hot water tank was then analysed by varying condenser coil pitch, diameter, and coil arrangement. From the analysis, varying the condenser coil pitch improved both the condenser heat transfer coefficient and heat pump COP. The findings in this study demonstrate the potential for enhancing the performance of air-source heat pump systems through innovative design modifications that optimise heat transfer within the thermal energy storage tank and condenser, ultimately leading to improved heat pump efficiency.

All the previous researchers examined the viability of incorporating heat pump systems with renewable energy sources and the utilization of thermal energy storage to create sustainable and cost-effective systems with minimized greenhouse gas emissions for space heating, domestic water heating, and HVAC systems. However, it can be inferred that while the integration of renewable energy sources into heat pump systems has a notable impact

on reducing carbon emissions, its feasibility largely hinges on prevailing climatic conditions. Consequently, this approach may not always represent a consistently sustainable and reliable solution, particularly in the case of large buildings that necessitate significant heat pump system capacities.

2.4 Variation on parameters of the heat pump system's components

Apart from the utilisation of renewable sources of energy, heat pump systems can also be optimised by modifying some parameters on their components. This includes: the evaporator, condenser, expansion valve, and compressor, all play a crucial role in heat pump performance. Thus, the choice of each of the components depends on the type of heat pump, whether it is an air-source or ground-source heat pump. The selection of heat pump components also depends on the main function of the system.

For instance, the usage of different working fluids can alter the performance of the heat pump, depending on the thermal properties of the refrigerant that is used. Investigations have shown that working fluids with low global warming potential (GWP) such as pentane, have better thermal properties than mostly used refrigerants with higher GWP such as R134a [19]. Thus, working fluids with better thermal properties improve the performance of the heat pump systems.

Optimization of heat pump systems can also be achieved by using carbon dioxide as a working fluid due to its low GWP, improved thermal properties such as low critical temperature as well as its zero-ozone potential depletion. However, when carbon dioxide is used as a working fluid in heat pump systems, the expansion valve is usually replaced with a Tesla turbine because the expansion valve tends to result in large exergy loss during the isentropic process. This is because the Tesla turbine can work with any two-phase fluid [20].

Energy consumption by heat pump systems can also be minimised by optimization of operating parameters such as obtaining the optimum compressor speed and airflow rate in the case of an air-source heat pump system. The higher the compressor speed, the more electrical energy is consumed and hence less operation time for the compressor. Various algorithms such as genetic algorithms, particle swarm optimization, and the methods of evolutionary computation can be used to optimise these operating parameters and hence improve the heat pump COP [21].

Condensers also play an important role in vapour compression heat pumps, and their energy consumption is significant. Research shows that a condenser consumes 20% of the heat pump's total electrical energy for a 1-degree Celsius change in temperature [22]. Stainless steel and plate heat exchangers are mostly used in heat pump condensers due to their high efficiency, tighter control of conditions, multi-stream and multi-pass configurations, and lower installation costs [23]. Moreover, plate heat exchangers are known to have a heat transfer coefficient that is averagely three to five times greater than that of the traditional tubular heat exchangers [24].

As for compressors, scroll types are mostly used in air-source heat pump systems, especially in air conditioning systems. These compressors are required to compress the refrigerant so that it leaves in the vapour state at a high temperature and high pressure, using the slide-bushing mechanism to control the working fluid. The scroll types are commonly used because of their high-performance coefficient, reduced vibrations, and minimum noise while in operation, hence consumption of electrical energy is reduced [25].

2.5 Heat exchangers

The selection of heat exchangers for use in the evaporators and condensers of a heat pump significantly influences the overall thermal performance of the system. These heat exchangers play a crucial role in heat pump systems because they are in direct contact with the environment, which varies across different applications of the system. It is, therefore, imperative to consider the specific context in which the heat exchanger will be integrated when designing it, as this consideration is paramount for enhancing efficiency.

Compact heat exchangers are the most suitable heat exchangers for use in heat pump systems because of their high ratio of heat transfer surface area to heat exchanger volume. The high ratio results in reduced volume and weight and therefore reduces the package sizing of heat pump systems. Moreover, compact heat exchanger results in less energy cost and high efficiency which is better than other heat exchanger types. Thus, compact heat exchanger sales increase by approximately 10% per year compared to 1% of sales from all other heat exchanger types [26].

2.5.1 Fin and tube heat exchanger

Finned tube heat exchangers are a commonly used type of heat exchanger in condensers and evaporators in heat pump systems which are used in HVAC systems. This is because of their configuration; tubes can be staggered and pass-through thin fins to improve heat

transfer surface area. Tubes are usually made of copper due to their high thermal conductivity and fins are made of aluminium. Air is blown through in between fins to the tube surface and allows for the transfer of heat either from or to the working fluid inside copper tubes during condensation and evaporation, respectively [27].

The tube shape of these types of heat exchangers may be circular, oval, or flat and these tubes usually sustain a maximum pressure of 3000 kPa without any leakage of the working fluid. Most studies discuss different fin patterns in fin and tube heat exchangers, and they include plain fin, wavy fin, louvre fin, and slit fin. These fin patterns play vital roles in the heat transfer and pressure drop characteristics of airside heat exchangers [28].

Even though fin and tube heat exchangers are commonly used, there are a few shortcomings associated with them. For instance, they tend to experience wake formation on the round tube, and tube expansion due to heat can lead to fin distortion, making them less efficient. Because of these, their fins have been modified to form louvres to increase their heat transfer area in a small space and hence increase the coefficient of heat transfer on the air side [29].

2.5.2 Enhancements on the heat exchangers

As an energy-saving initiative, a few modifications can be applied to the heat exchanger in both condensers and evaporators, and these are used in vapour compression heat pumps for space heating, domestic water heating, and air conditioning purposes. Most modifications are made to the geometry of the heat exchanger, especially on the air side of the heat exchanger as it contributes almost eighty percent or more of the total thermal resistance of the heat exchanger [30]. This is performed to increase the heat exchanger heat transfer surface area and hence reduce the electrical energy consumption by the system.

One of the modifications was to introduce the use of porous baffles and copper fins in the shell and tube heat exchanger, with and without helical fins on the tube side of the heat exchanger to improve heat transfer. This was modelled and analysed using ANSYS CFD fluent software. The results show that a heat exchanger with helical-shaped fins has higher effectiveness, heat transfer rates, and heat transfer coefficient [31]. This shows that the modified helical heat exchanger can result in a better system with reduced energy consumption than a conventional tubular heat exchanger, under the same air velocity when integrated into the evaporator used in the vapour compression heat pump system.

Apart from that modification, Bhuiyan et al. [32] discussed the performance analysis of the air side of a four-row plain fin and tube heat exchanger using CFD. The study carried out analysis on the impact of varying the geometrical parameters such as both the longitudinal and transverse pitch of the tube spacing and fin pitch and orientation (staggered and inline). The results from the analysis show that for all the Reynolds numbers between 400 to 2000, heat transfer rates and pressure drop on the air side increase with the decreasing tube spacing (longitudinal and transverse) and increase with the increasing fin pitch for both inline and staggered tube orientations. It was also observed that staggered tube arrangement results in enhanced heat transfer in inline arrangement due to better flow mixing in a staggered arrangement.

Furthermore, Ishaque and Kim [33] investigated the interaction of non-uniform air flow with a finned tube heat exchanger which is used in residential heat pump systems to optimise the system for better energy efficiency. The two-dimensional (2D) analysis was based on the thermohydraulic performance related to heat transfer rate and pressure drop correlations for all fluid states in a multirow finned tube heat exchanger using MATLAB and CFD. The results from the analysis suggested that the non-uniform air distribution across heat exchanger fins influences pressure drop, heat exchanger capacity, and the refrigerant outlet temperature and hence the COP of the heat pump.

Taler et al. [34] observed that airflow on the air side of the heat exchanger, either in a condenser or evaporator, has a significant impact on the thermal performance of the heat exchanger. His observations suggest that airflow is proportional to the heat transfer coefficient of the airside and hence the heat transfer rate. The higher the airflow, the more heat is transferred between the working fluid side and the air side, and hence higher COP may be achieved. However, heat exchangers' performance decreases rapidly when exposed to very high non-uniform airflow and high-pressure losses are incurred [35].

Also, Okbaz et al. [36] investigated the behaviour of heat exchangers by varying fin types, fin pitch, and tube row numbers, for optimization purposes. The comparison was made between wavy and louvred fins. The experimental results show that the pressure drop experienced on the louvred finned tube heat exchanger is higher than that with wavy fins. It was also observed that high fin pitch especially at low air velocity makes it difficult for air to be directed through in between fins, and hence results in low heat transfer rates. The thermal performance of the louvred finned tube heat exchanger was obtained to be higher than that of the wavy finned tubes. This means that a finned tube heat exchanger can be

optimised by using louvred fins, with a small fin pitch and more number or tube rows to increase the heat transfer area of the heat exchanger.

For further investigations, Saleem and Kim [37] performed numerical analysis on the thermal analysis of the multi-louvred fin and flat tube heat exchanger using different heat exchanger configurations with varying louvre angles and fin pitches at low Reynolds numbers (300 to 500) on the air side of the heat exchanger. The 36 heat exchanger configurations were obtained from the combinations of different louvres with different fin pitches at different Reynolds numbers on the air side of the heat exchanger. The analysis shows that the optimum thermal performance of the heat exchanger in terms of air-side heat transfer coefficient is obtained at a louvred fin angle of 19 degrees Celsius, at a flow depth of 16 mm, and fin pitch of 1 mm.

Moreover, Li et al. [38] experimentally investigated the improvement of the air-side heat transfer performance of the fin and tube heat exchanger by the implementation of the convex fin. The thermal transfer performance of the heat exchanger with four convex strips around each tube was compared with a plain fin heat exchanger and the results show that the heat exchanger with interrupted fins has an improved heat transfer coefficient of 25% higher than that with a plain fin. Also, heat exchangers with convex fins experienced a pressure increase of 16% more than that with plain fins. That means there could be significant savings in energy consumption by using the heat exchanger with convex fins instead of plain fins in either the condenser or evaporator of the heat pump system.

Louvred fins can also be modified by using delta winglet vortex generators to improve the air-side heat transfer performance of the heat exchanger. Huisseune et al. [39] numerically investigated the effect of this modification on the air side of the heat exchanger using CFD. The analysis shows that the introduction of delta winglet vortex generators has improved heat transfer by; reducing the tube boundary layer thickness, improving fluid mixing, and delaying the flow separation that occurs on the tube surface. However, the generated vortex results in increased pressure losses due to increased friction and flow obstruction.

The air side surface of the heat exchanger in heat pump systems can also be improved using wavy fins. Some researchers studied and compared the heat exchanger with herringbone wavy fins with the one with convex strips of herringbone wavy fins. The results show that the heat exchanger with herringbone wavy fins with convex strips, has a higher heat transfer rate, heat transfer coefficient, and high-pressure drops than the one

with herringbone wavy fins. Also, for both configurations, the thermal performance of the heat exchanger was observed to improve at higher frontal air velocity than at low velocities [40].

Moreover, Matos et al. [41] show that numerical and experimental investigations were made to compare the heat transfer performance between circular tubes and elliptical tubes in the flat-finned heat exchanger. The results proved that the fin and tube heat exchanger with elliptical tubes has a heat transfer rate that is approximately 19% higher than that with circular tubes. Thus, elliptical tubes provide much more heat transfer area for heat to be transferred between the tube side and air side and hence improve the energy efficiency of the system in which the heat exchanger is integrated, for instance, the heat pump system.

In addition, Tao et al. [42] investigated 3D analysis whereby variations were made on the geometry of fins and tubes and observed the effects of Reynolds number and fin thickness on the heat transfer of the heat exchanger. The tube shapes comparison was made between circular and elliptical tube shapes, and both tube shapes were finned with wavy fins. From the results, it was observed that the wavy fin and tube heat exchanger with elliptical tube arrangement has higher heat transfer and higher-pressure loss than in circular tube arrangement. It was also observed that as the Reynolds number increases, the temperature gradient drops, leading to a low heat transfer rate and hence poor heat exchanger performance for both tube arrangements. It was also observed from the analysis that as the fin thickness increases, there is an increase in the airflow velocity in between the fins, resulting in high-pressure drop and high thermal resistance, leading to low heat exchanger performance.

Hu et al. [43] also discussed the effects of using different fin materials on the performance of the heat exchanger. Three fin and tube heat exchangers with different fin materials were compared: aluminium fins, copper fins, and aluminium fins with a hydrophilic layer. The results from the analysis indicated that under identical ambient conditions, a heat exchanger that is made up of copper fins has the highest performance than others, followed by an aluminium fins heat exchanger while aluminium with hydrophilic layer fins became the worst. This is because copper is more conductive than aluminium and therefore fins made up of copper will absorb or release heat transfer at a higher rate than aluminium fins when exposed to the same ambient conditions.

Moreover, Tang et al. [44] also investigated the thermal performance of crimped spiral fin and tube heat exchangers when compared with plain fin and tube heat exchangers. This fin configuration improves the heat transfer surface area at the contact point between the fin and tube surface and hence increases the heat exchanger performance. The studies show that even though this configuration improves the heat transfer performance of the heat exchanger on the air side, there are higher pressure losses experienced than in plain fin and tube heat exchangers.

Pongsoi et al. [45] further investigated the effects of the number of tube rows on the air-side performance of the crimped spiral heat exchanger type. The study shows that at high Reynolds numbers (3,000-13,000), the effect of the number of tube rows on friction factors and hence on airside performance is negligible. This is caused by the good mixing of air streams on the tube surface due to turbulence eddies. The results also show that the average heat transfer rate and pressure losses on the airside of the crimped spiral fin and tube heat exchanger increase with the increase in the number of tube rows.

Tang et al. [46] compared the thermal performances of different fin patterns of the fin and tube heat exchanger at high Reynolds numbers (4,000-10,000). The study compared five fin patterns, namely: crimped spiral fin, plain fin, slit fin, fin with delta-wing longitudinal vortex generators as well as mixed fin with front 6-row vortex-generator fin and rear 6-row slit fin. The results show that the fin pattern with the highest heat transfer rate and pressure losses when compared with all other fin patterns, is the crimped spiral fin. Also, it was observed that the thermal performance of the heat exchanger with pattern fins of delta-wing longitudinal vortex generators can be improved by the larger angle of attack, higher length, and smaller height of vortex generators.

2.6 Gap in existing research

In pursuit of addressing the existing knowledge gap in heat pump heat exchanger analysis, this study offers a valuable advantage. One of the advantages of this project is to develop a 1-D analysis which is computationally less expensive when compared to complete modelling of a heat pump heat exchanger. Despite the considerable research conducted on numerical analysis of heat pump heat exchangers, there remains a knowledge gap. This gap pertains to the necessity for further investigation, specifically focusing on the incorporation of three-dimensional analysis. This analysis should be centred on the nonuniform airflow present on the air side of fin and tube heat exchangers. A key aspect of this inquiry involves

understanding how the nonuniform airflow distribution impacts the overall thermal efficiency of the heat exchanger. Such an understanding is vital as it directly influences the COP of the heat pump. Therefore, the purpose of this study is to determine the heat exchanger airside nonuniform impacts on the refrigerant pressure drop, overall heat transfer coefficient, heat transfer capacity and COP of the heat pump. Thus, each tube of the heat exchanger is discretized into several increments to investigate how both fluids' properties vary along the refrigerant tube length.

3. Methodology

3.1 Model development

This chapter details all the concepts applied as well as the research design followed to acquire all the findings discussed later in this document. This chapter is divided into two sections, that is, the first section describes the fin and tube heat exchanger model development using thermodynamics and heat transfer basics, while the second section describes the development of the heat pump cycle model in which the developed heat exchanger modelled in section one is integrated into the cycle model.

3.2 Heat exchanger model development

This section presents the approach in detail, followed to model and perform numerical analysis on the heat exchanger of known dimensions from the literature. The heat pump fin and tube heat exchanger is modelled using Python. The modelled heat exchanger in this study is used as an evaporator, that is, as an outdoor unit, used mainly to absorb or extract heat from the outside air being blown through the heat exchanger tubes to increase the heat transfer between the working fluid and air.

3.2.1 Heat exchanger geometry

The heat exchanger geometry is crucial in determining the heat transfer area as well as the heat transfer coefficient of the heat exchanger. That is, the larger the heat exchanger surface area, the larger the amount of heat is transferred between air and working fluid inside the piping and hence increased effectiveness of the heat exchanger. Figure 2 and 3 depict the physical model and schematic diagram of the finned tube heat exchanger, respectively.

Figure 2: Physical model of a typical finned tube heat exchanger[47]

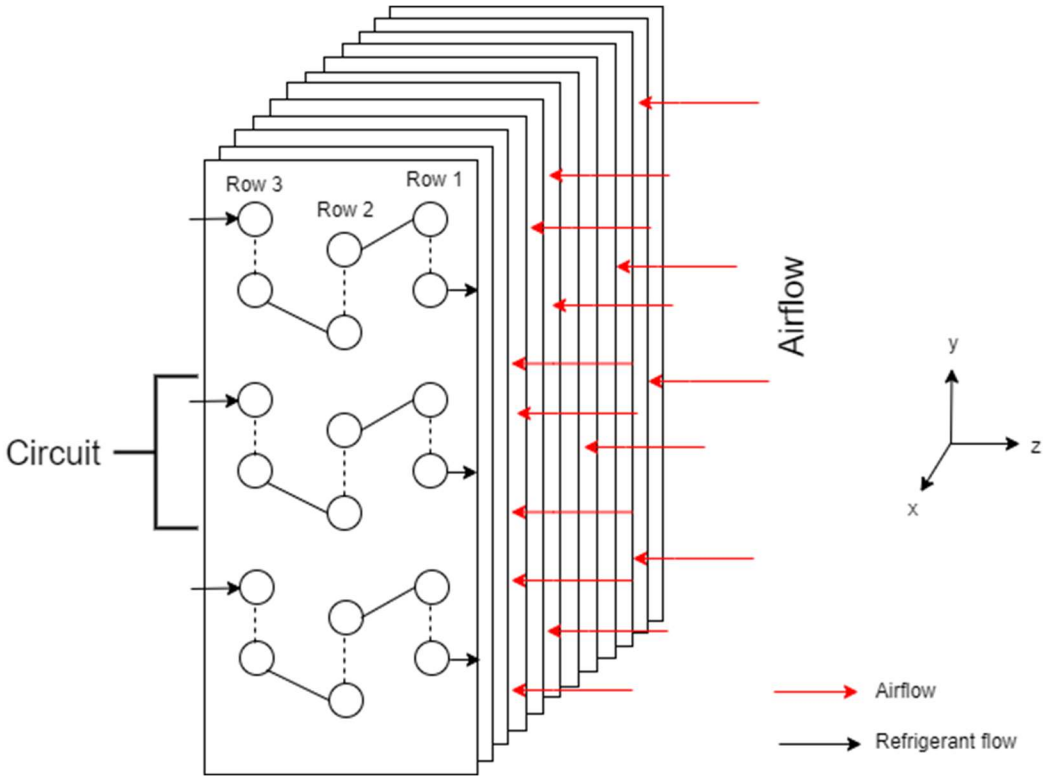


Figure 3: Schematic of the heat exchanger model

3.2.1.1 Tube and fins specifications

The finned tube heat exchanger investigated in this study is part of an outside unit of a heat pump system that is used for residential air conditioning. This particular heat exchanger geometry is characterized by three rows in depth, featuring thirty tube circuits with six tubes per circuit.

In this physical model, the heat transfer occurs as refrigerant R410a circulates within the copper tubes, while the aluminium fins are securely affixed to the exterior surface of these tubes. The primary objective of this geometric configuration is to substantially augment the efficiency of the heat transfer process through the substantial increase of available surface area for heat exchange[48].

By employing copper tubes with excellent thermal conductivity and aluminium fins that offer the advantages of being lightweight and corrosion-resistant, this design improves heat transfer efficiency while ensuring the longevity and durability necessary for residential heat pump applications. Table 1 and 2 depict some of the geometry parameters and specifications of the heat exchanger under investigation, respectively[49].

The use of louvred fins in this design is significant for enhancing heat exchanger performance. The louvred fin type creates a corrugated surface on the fins, which not only increases the surface area for heat exchange but also promotes better airflow and heat distribution[50]. This feature is particularly advantageous in residential settings, where maintaining consistent and efficient heat transfer is essential for achieving desired indoor temperatures.

Additionally, the staggered tube arrangement in this heat exchanger design plays a crucial role. By offsetting the tubes in adjacent rows, this arrangement further enhances heat transfer efficiency[51]. It minimizes the obstruction of airflow between tubes and ensures that each tube benefits from the convective heat transfer of the airflow.

Table 1: Outdoor heat exchanger tube specifications[33]

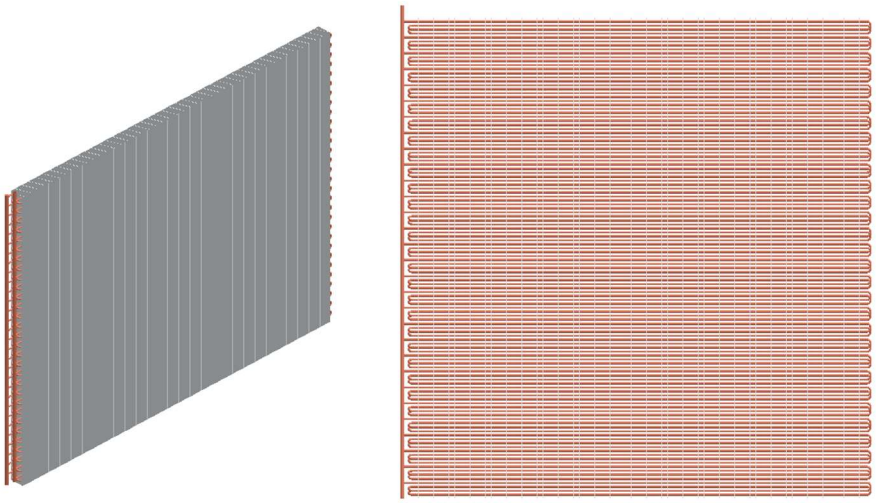
Tube material	Copper
Tube outer diameter (mm)	7
Tube inner diameter (mm)	6.3
Transverse tube pitch (mm)	25
Longitudinal tube pitch (mm)	22
Length of the finned tube (mm)	1720

Number of tube rows	3
Total number of tubes	180
Tube circuits	30
Number of tubes per circuit	6

Table 2: Outdoor heat exchanger fins specifications[33]

Fin material	Aluminium
Fin type	Louver
Fin thickness (mm)	0.092
Fin collar diameter (mm)	Tube outer diameter + 2 * Fin thickness
Fin pitch (mm)	1.7
Fin space (mm)	Transverse tube pitch – Fin thickness
Tube arrangement	Staggered

Figure 4 shows different views of a fin and heat exchanger modelled in SolidWorks. The heat exchanger consists of 180 copper circular tubes that are 1720 mm long and 0.35 mm thick. Refrigerant R410a was used as the working fluid for this study.



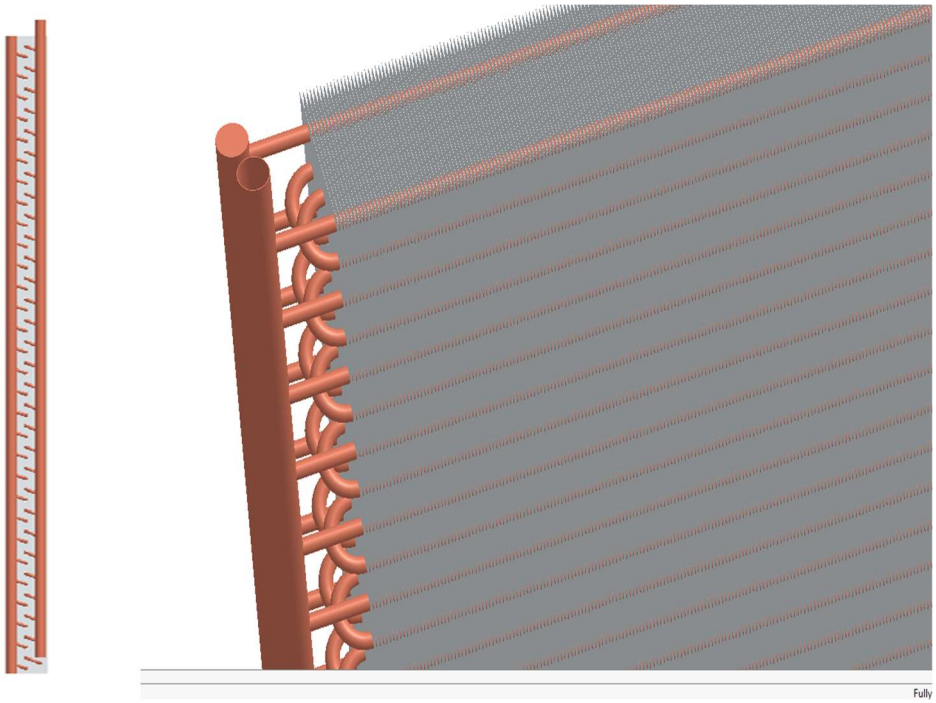


Figure 4: Finned heat exchanger as modelled in SolidWorks

3.2.1.2 Geometry calculations

Table 3 depicts formulae used to calculate the geometry parameters of a fin and tube heat exchanger:

Table 3: Heat exchanger geometry calculations

Parameter	**Equation**
Outer tube wall surface area	$A_o = \pi D_o L$
Inner tube wall surface area	$A_i = \pi D_i L$
Cross-sectional tube area	$A_c = \pi \frac{D_i{}^2}{4}$
Heat exchanger width	$W_{hx} = L$
Heat exchanger height	$H_{hx} = (N_{th} + 1)P_T$
Heat exchanger depth	$D_{hx} = N_{tr} P_L$
Frontal area	$A_f = W_{hx} H_{hx}$
Number of fins along each tube	$N_{fins/tube} = \frac{L}{P_f + t_f}$
Number of fins per	$N_{fins/length} = \frac{N_{fins/tube}}{L}$

Parameter	Equation
unit length	
Heat exchanger volume	$V_{hx} = A_f D_{hx}$
Refrigerant total heat transfer area	$A_{ht_{ref}} = \pi D_i L L_{tot}$
Refrigerant heat transfer area over core volume	$\alpha_{ref} = \frac{A_{ht_{ref}}}{V_{hx}}$
Surface area of fins	$A_{fins} = 2N_{fins/length} L((D_{hx} H_{hx}) - (N_{tubes} \frac{\pi {D_o}^2}{4}))$
Surface area of tubes	$A_{tubes} = (\pi D_o - t_f N_{fins/length}) L N_{tubes}$
Air side heat transfer area	$A_{ht_{air}} = A_{tubes} + A_{fins}$
Air side heat transfer area over core volume	$\alpha_{air} = \frac{A_{ht_{air}}}{V_{hx}}$
Heat transfer area ratio	$\frac{A_{ht_{ref}}}{A_{ht_{air}}} = \frac{\alpha_{ref}}{\alpha_{air}}$
Hydraulic diameter	$D_h = \frac{4A_f}{A_o}$

3.2.2 Operating conditions

This model is developed under the assumption that the heat exchanger being modelled is for a heat pump outdoor unit and it is operating under atmospheric pressure of 101.325 kPa. In this study, the operating conditions shown in Table 4 which were chosen based on the European standard EN14825[49], are used to develop and validate the model.

The model in this study only considered the conditions where the heat exchanger is used as an evaporator, that is, where it absorbs heat from atmospheric air blown on its surface area.

Table 4: Air and refrigerant sides operating conditions for a heat exchanger used as an evaporator[49]

Operating Condition	**Refrigerant mass flowrate (kg/s)**	**Inlet refrigerant temperature (ºC)**	**Refrigerant inlet pressure (kPa)**	**Dry –bulb/ Wet –bulb temperature (ºC)**	**Average face velocity (m/s)**
H1	0.06806	-13.7	503.39	-7/-8	1.1
H2	0.04417	-2.8	728.95	2/1	1.0
H3	0.02639	3	876.05	7/6	0.7
H4	0.01806	7.7	1013.34	12/11	0.5

3.2.3 Governing equations and boundary conditions

The mass and momentum equations shown below are the governing equations used for the three – dimensional turbulence airflow through a finned tube heat exchanger. The mass balance is given as:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \boldsymbol{u} = 0 \tag{3.1}$$

For incompressible airflow, $\frac{\partial \rho}{\partial t} = 0$,

$$\therefore \nabla \cdot \boldsymbol{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z} = 0 \tag{3.2}$$

The momentum balance (Navier-Stokes) equations for steady-state incompressible airflow are given as:

$$\nabla \cdot (\rho u_x \boldsymbol{u}) + \frac{\partial p}{\partial x} - \nabla \cdot \tau_x = \rho g_x + F_x \tag{3.3}$$

$$\nabla \cdot \left(\rho u_y \boldsymbol{u}\right) + \frac{\partial p}{\partial y} - \nabla \cdot \tau_y = \rho g_y + F_y \tag{3.4}$$

$$\nabla \cdot (\rho u_z \boldsymbol{u}) + \frac{\partial p}{\partial z} - \nabla \cdot \tau_z = \rho g_z + F_z \tag{3.5}$$

The boundary conditions applied are:

i) No slip boundary condition at the walls:

$$u_y = u_x = 0 \tag{3.6}$$

ii) The velocity inlet boundary condition at the front face of the heat exchanger:

$$u_z = u_{in} \tag{3.7}$$

$$u_y = u_x = 0 \tag{3.8}$$

iii) Zero pressure gradient at the outlet of the airflow domain.

3.2.4 Computational Fluid Dynamics modelling of 3D velocity profile

This section presents a detailed modelling scheme conducted for the non-uniform airflow on the heat exchanger calculated using CFD code, Ansys Fluent [52]. The Ansys Fluent modelling scheme for the finned heat exchanger encompasses several key steps, including geometry and meshing, flow domain setup, porous media modelling for the fins, solver selection, and convergence assessment. These steps collectively enable a detailed and accurate simulation of airflow within the heat exchanger, with particular consideration of turbulence effects.

3.2.4.1 Geometry and meshing

The modelling process started with the creation of a 3D geometry representation of the heat pump system in Ansys Fluent. This geometry serves as the foundation for the subsequent simulation. A crucial aspect of this step is mesh generation, which involves dividing the domain into discrete elements. In this particular case, a mesh with a total of 411,060 elements was successfully generated. The quality and resolution of this mesh are essential factors that influence the accuracy of the simulation results.

3.2.4.2 Flow domain setup

Within the defined geometry, the next step involves setting up the flow domain. This domain encompasses the entire heat pump system. To accurately capture the behaviour of the airflow, the simplest two-equation model, the basic k-ε turbulence model, has been employed to resolve turbulence in the fluid domain. By solving two different transport equations, this approach enables the independent determination of the turbulent velocity and length scales[53].

3.2.4.3 Porous media modelling

Performing thorough numerical analysis on the fin and tube heat exchanger can be very expensive, and time-consuming, and powerful computers may be needed for such analysis. Therefore, utilizing the known velocity vs. pressure loss characteristics as adopted by An and Kim[54], a porous media technique was utilized in this study. In other words, a 3D heat exchanger with many tubes and fins was reduced to a 1D thin membrane known as a porous jump[49]. This strategy, used by An and Kim[54], produced reliable outcomes without sacrificing accuracy.

3.2.4.4 Solver settings

The solver employed in this simulation is of the pressure-based type. This choice signifies that the simulation solves the governing equations of fluid flow with a focus on pressure corrections. Additionally, the simulation is configured to operate in a steady-state mode. This means that it does not consider transient effects over time but instead seeks to find a stable solution for the given conditions.

3.2.4.5 Convergence criteria

A critical aspect of the simulation process is assessing convergence. Convergence is considered achieved when certain criteria are met. Specifically, the scaled residuals for mass, momentum, turbulent kinetic energy, and turbulence dissipation rate should all be below 10^{-4}. Furthermore, energy residuals should register at levels below 10^{-6}. These criteria indicate that the solution has converged to a high degree of accuracy, ensuring reliable results for the heat pump system simulation.

The following assumptions are made on the Fluent numerical scheme:

- The finned tube heat exchanger makes one side of the heat pump casing,
- Steady-state and incompressible flow on both the tube side and air side of the heat exchanger,
- No heat generation within the fluid or solid.

The simplified representation of the heat pump numerical model as modelled in DesignModeler is shown in Figure 5. Moreover, Figure 6 depicts the representation of a meshed heat exchanger simplified model with a fluid flow region.

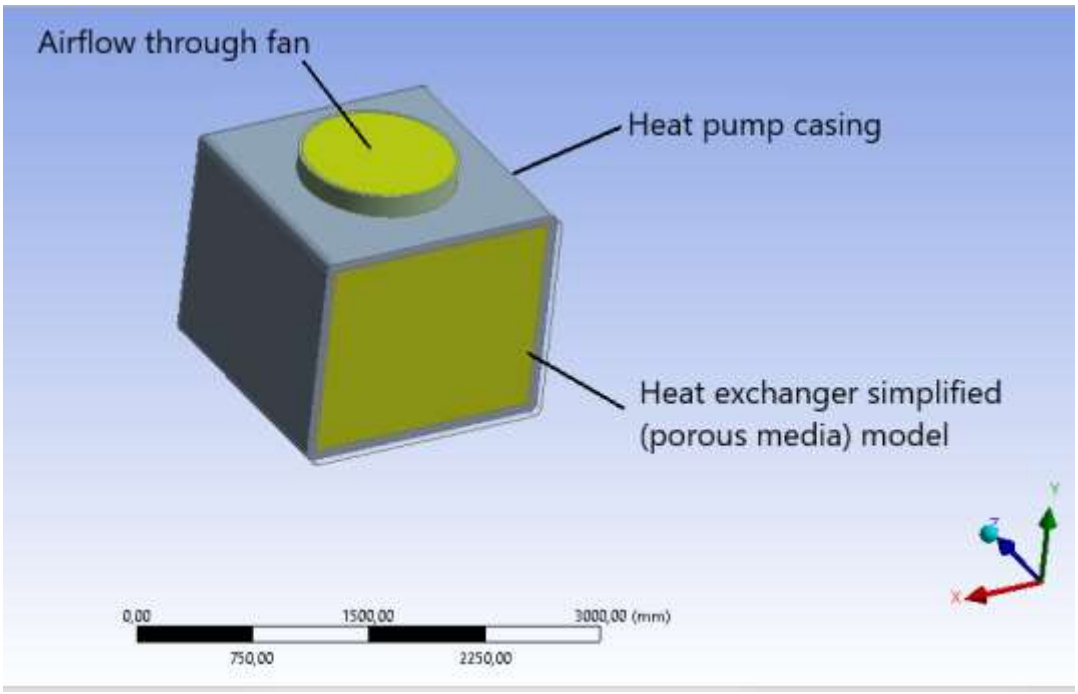


Figure 5: Simplified heat pump numerical model in Design Modeler

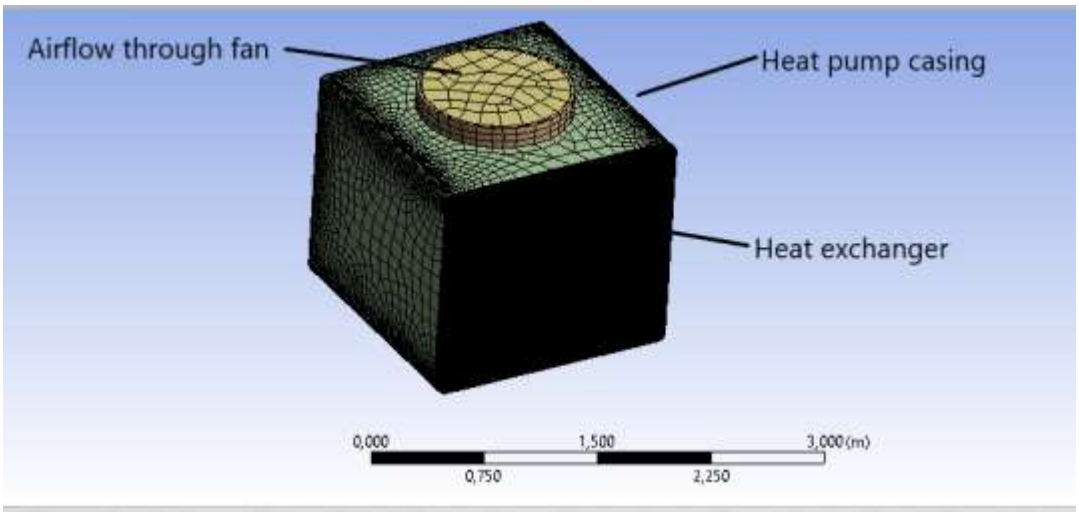


Figure 6: Meshed model of a simplified heat pump and airflow region

3.2.5 Heat transfer coefficients

Heat transfer coefficients for both the air side and tube side are calculated for each small increment into which the heat exchanger is discretized. The discretisation of the heat exchanger is explained in section 3.2.8.4. Different correlations have been used to calculate heat transfer coefficients for both air and tube sides.

3.2.5.1 Air side heat transfer coefficient

Theoretically, the airside of the evaporator has a lower convective heat transfer coefficient than the refrigerant side. The following procedure was followed in this study to calculate the heat transfer coefficient on the air side of the heat exchanger:

The Reynolds number of airflow for the staggered arrangement of tubes is calculated as:

$$Re_{air} = \frac{\rho_{air} v_{air\_max} D_o}{\mu_{air}} \tag{3.9}$$

Where the maximum velocity of air is given by:

$$v_{air\_max} = \frac{P_T}{P_T - D_o} v_{air} \tag{3.10}$$

Also, the Prandtl number is given by:

$$\mathrm{Pr}_{air} = \frac{c_{p\_air} \mu_{air}}{K_{air}} \tag{3.11}$$

And the Nusselt number:

$$Nu_{air} = j_H Re_{air} Pr_{air}^{\frac{1}{3}} \quad [55] \tag{3.12}$$

In this context, $\boldsymbol{j_H}$ is referred to as the Colburn j-factor and can be calculated using the chart depicted in Figure 45 in Appendix B: Heat exchanger. Consequently, we can establish a mathematical connection between the Reynolds number and the Colburn j-factor by representing it as a linear equation on a logarithmic graph. In simpler terms, if two points on the graph in Figure 45 in Appendix B: Heat exchanger are selected, the following relationship can be obtained [55]:

$$Log\ j_H = -0.394 log Re_{air} - 0.7773 \tag{3.13}$$

The heat transfer coefficient of air is, therefore calculated as:

$$htc_{air} = \frac{Nu_{air}K_{air}}{D_h} \tag{3.14}$$

Atmospheric air is not purely dry, that is, it has some amount of vapour in it. Therefore, both heat and mass transfer takes place between the tube's outer surface and air flow. Thus, using the following mass and heat transfer formulae, the overall convective heat transfer coefficient on the air side can now be calculated:

For the mass balance of the air and film of water on the outer surface of the tube:

$$\Delta\dot{m}_w = \dot{m}_{air}\Delta w_{air} \tag{3.15}$$

$$\Delta\dot{m}_w = htc_{air}\frac{A_{ht_{air}}}{c_{p_{air}}L_e}(w_{sw} - w_{air}) \tag{3.16}$$

And the energy balance of air and water film on the tube's outer surface:

$$\Delta\dot{Q}_w = -\dot{m}_{air}\Delta h_{air} \tag{3.17}$$

$$\Delta\dot{Q}_{air} = htc_{air}A_{ht_{air}}(T_{sw} - T_{air}) \tag{3.18}$$

Rearranging equations (3.15), (3.16), (3.17), and (3.18) yields equation (3.19), which reflects the total mass and heat transfer between airflow and the moisture coating on the tube surface.

$$\dot{Q}_{evap} = \dot{m}_{air}\Delta h_{air} = UA_{airside}((T_{sw} - T_{air}) + \frac{1}{c_{p_{air}}L_e}(w_{sw} - w_{air})h_{gw}) \tag{3.19}$$

Making $UA_{airside}$ the subject of the formula in equation 3.19 results in:

$$UA_{airside} = \dot{Q}_{evap}((T_{sw} - T_{air}) + \frac{1}{c_{p_{air}}L_e}(w_{sw} - w_{air})h_{gw})^{-1} \tag{3.20}$$

Therefore, the overall convective heat transfer coefficient on the airside is determined using equation 3.21 below:

$$\frac{1}{UA_{airside}} = \frac{1}{\eta_{air}htc_{air}A_{ht_{air}}} \tag{3.21}$$

Where η_{air} is the overall efficiency on the air side, and it is calculated by equation (3.21):

$$\eta_{air} = 1 - \frac{A_{fins}}{A_{ht_{air}}}(1 - \eta_{fin}) \tag{3.22}$$

The efficiency of the fins is calculated from the graph in Figure 46 in Appendix B: Heat exchanger.

3.2.5.2 Tube side heat transfer coefficient

The calculation of the heat transfer coefficient of the refrigerant side is dependent on the phase of refrigerant flowing inside, and therefore different correlations are used for different phases of the refrigerant.

Single-phase (supercooled or superheated) heat transfer coefficient correlations

The method employed to determine the convective heat transfer coefficient of air, as discussed in section 3.2.5.1, is also applied to estimate the heat transfer coefficient of the working fluid within the superheated or subcooled region. The only distinction lies in the Nusselt number calculation for superheated refrigerant, as indicated by the following Dittus-Boelter equation [55].

$$Nu_{ref} = 0.023 Re_{ref}^{0.8} \mathrm{Pr}_{ref}^{0.4} \quad (3.23)$$

$$htc_{ref\_sp} = \frac{Nu_{ref} K_{ref}}{D_i} \quad (3.24)$$

The overall heat transfer coefficient in the refrigerant side for the single-phase region is calculated using the effectiveness-NTU method as detailed below (equation (3.25) to (3.35)):

$$\dot{Q}_{evap\_sp} = \dot{m}_{air} \Delta h_{ref} \quad (3.25)$$

$$c_{air} = \dot{m}_{air} c_{p\_air} \quad (3.26)$$

$$c_{ref} = \dot{m}_{ref} c_{p\_ref} \quad (3.27)$$

$$c_{min} = \min\left(c_{p_{ref}} \dot{m}_{ref}, c_{p_{air}} \dot{m}_{air}\right) \quad (3.28)$$

$$c_{max} = \max\left(c_{p_{ref}} \dot{m}_{ref}, c_{p_{air}} \dot{m}_{air}\right) \quad (3.29)$$

$$c_r = \frac{c_{min}}{c_{max}} \quad (3.30)$$

The effectiveness is given as:

$$\varepsilon = \frac{\dot{Q}_{evap\_sp}}{c_{min}(T_{air} - T_{ref})} \quad (3.31)$$

The NTU is calculated from equations (3.32) and (3.33) below:

$$\text{If } c_r \neq 1,\ NTU = -\ln\left(1 + \frac{1}{c_r} \ln(1 - \varepsilon c_r)\right) \quad (3.32)$$

$$\text{Else } (c_r = 1),\ NTU = -\ln(1 - \varepsilon) \text{ [55]} \quad (3.33)$$

Hence, the overall convective heat transfer coefficient in the tube side during the superheated phase is computed using equation (3.34) below:

$$UA_{tube-side_{sp}} = NTUc_{min} \tag{3.34}$$

Which can be rewritten as:

$$\frac{1}{UA_{tube-side_{sp}}} = \frac{1}{\eta_{ref} htc_{ref} A_{ht_{ref}}} \tag{3.35}$$

The efficiency on the tube side η_{ref} is assumed to be 100%, that is, the tube is smooth, with no fins inside.

Two-phase (liquid and vapour mixture) correlations

The convective heat transfer coefficient of a refrigerant in the two-phase region can be determined using the Chato/Wattelet correlation, which asserts that the two-phase region heat transfer coefficient is given by[56]:

$$htc_{ref\_tp} = 0.023 Re_{ref}{}^{0.8} \mathrm{Pr}_{ref}{}^{0.4} \frac{K_{ref}}{D_i} (4.3 + 0.4(1000Bo)^{1.3}) \tag{3.36}$$

Where the Bond number, $Bo = \frac{\dot{Q}_{evap\_tp}}{\frac{\dot{m}_{ref}}{A_i} h_{fg} A_{ht_{ref}}}$ (3.37)

Therefore, the overall convective heat transfer coefficient in the tube side for two-phase is calculated using the following equation:

$$UA_{tube-side_{tp}} = \frac{\dot{Q}_{evap\_tp}}{(T_{sw} - T_{ref})} \tag{3.38}$$

$$\frac{1}{UA_{tube-side_{tp}}} = \frac{1}{\eta_{ref} htc_{ref\_tp} A_{ht_{ref}}} \tag{3.39}$$

Correlations for phase changes:

The change from the superheated region to a mixture of liquid and vapour phase region:

$$\dot{Q}_{evap} = \dot{Q}_{evap\_tp} + \dot{Q}_{evap\_sh} = \dot{m}_{ref}(h_g - h_i) + \dot{m}_{ref}(h_e - h_g) \tag{3.40}$$

$$\dot{Q}_{evap} = htc_{ref\_tp} A_{ht_{ref\_tp}} \Delta T + htc_{ref\_sh} A_{ht_{ref}} \Delta T \tag{3.41}$$

Rearranging Equations (3.40) and (3.41) yields the surface area ratio of two regions:

$\Delta h_{tp} = h_e - h_g \ \& \ \Delta h_{sh} = h_g - h_i$

$$A_{ht_{ref_{tp}}} = \frac{\Delta h_{tp} htc_{ref\_sh}}{\Delta h_{sh} htc_{ref\_tp}} A_{ht_{ref_{sh}}} \tag{3.42}$$

And $A_{ht_{ref_{tot}}} = A_{ht_{ref_{sh}}} + A_{ht_{ref_{tp}}}$

Now let $C_{tp} = \frac{\Delta h_{tp} htc_{ref\_sh}}{\Delta h_{sh} htc_{ref\_tp}}$ and $f_{tp} = \frac{A_{ht_{ref_{tp}}}}{A_{ht_{ref_{tot}}}}$

Rearranging the above equations results in:

$$f_{tp} = \frac{A_{ht_{ref_{tp}}}}{A_{ht_{ref_{tot}}}} = \frac{C_{tp}}{C_{tp}+1} \tag{3.43}$$

The change of refrigerant from the two-phase region to the subcooled region:

$$\dot{Q}_{evap} = \dot{Q}_{evap\_tp} + \dot{Q}_{evap\_sh} = \dot{m}_{ref}(h_f - h_i) + \dot{m}_{ref}(h_e - h_f) \tag{3.44}$$

$\Delta h_{tp} = h_e - h_f \ \& \ \Delta h_{sh} = h_f - h_i$

The equations (3.41), (3.42) & (3.43) can now be used again.

The change of refrigerant from the two-phase region to the superheated region:

$$\dot{Q}_{evap} = \dot{Q}_{evap\_tp} + \dot{Q}_{evap\_sh} = \dot{m}_{ref}(h_g - h_i) + \dot{m}_{ref}(h_e - h_g) \tag{3.45}$$

$\Delta h_{tp} = h_g - h_i \ \& \ \Delta h_{sh} = h_e - h_g$

The equations (3.41), (3.42) & (3.43) can now be used again.

The change from a subcooled region to a two-phase region

$$\dot{Q}_{evap} = \dot{Q}_{evap\_tp} + \dot{Q}_{evap\_sh} = \dot{m}_{ref}(h_f - h_i) + \dot{m}_{ref}(h_e - h_f) \quad (3.46)$$

$$\Delta h_{tp} = h_e - h_f \ \ \& \ \ \Delta h_{sh} = h_f - h_i$$

The equations (3.41), (3.42) & (3.43) can be used again.

As a result, the convective heat transfer coefficient of the refrigerant at any phase change boundary is computed using the equation (3.47) below [57]:

$$htc_{ref} = \frac{A_{ht_{ref_{tp}}} ht_{\ ref\_tp} + A_{ht_{ref_{sh}}} htc_{ref\_sh}}{A_{ht_{ref_{tot}}}} \quad (3.47)$$

Therefore, the overall heat transfer coefficient of the fin and tube heat exchanger is given by:

$$\frac{1}{UA_{tot}} = \frac{1}{A_{ht_{ref_{tp}}} htc_{ref\_tp} + A_{ht_{ref_{sh}}} htc_{ref\_sh}} + R_w + R_f + \frac{1}{\eta_{air} htc_{air} A_{ht_{air}}} \quad (3.48)$$

Where $R_w = \frac{ln\frac{R_o}{R_i}}{2\pi K_{Cu} L}$: thermal resistance due to tube thickness.

And $R_f = \frac{R_{f\_o}}{\eta_{air} A_{ht_{air}}} + \frac{R_{f\_i}}{A_{ht_{ref}}}$: Thermal resistance due to fouling of the tube surface.

3.2.6 Mass and heat transfer

As previously mentioned in the previous section, the air that is blown on the heat exchanger tubes contains some moisture, therefore, both mass and heat transfer occur in the air side of the heat exchanger. Heat transfer rates between air and refrigerant are computed by calculating mass and heat balance on both air and moisture as well as the refrigerant heat balance.

Letting i and e subscripts represent inlet and outlet conditions respectively, the mass balance of dry air is computed as $\dot{m}_{air\_i} = \dot{m}_{air\_e}$. That is, the dry air mass is conserved.

Moisture mass balance in atmospheric air is given as:

$$\dot{m}_{air\_e} w_{air\_e} = \dot{m}_{air\_i} w_{air\_i} + \Delta\dot{m}_w \tag{3.49}$$

Humid air energy balance:

$$\dot{m}_{air\_e} h_{air\_e} = \dot{m}_{air\_i} h_{air\_i} + \dot{Q}_{air} \tag{3.50}$$

Refrigerant mass balance:

$$\dot{m}_{ref\_i} = \dot{m}_{ref\_e} \tag{3.51}$$

Refrigerant energy balance:

$$\dot{m}_{ref\_e} h_{ref\_e} = \dot{m}_{ref\_i} h_{ref\_i} + \dot{Q}_{ref} \tag{3.52}$$

Water film on the tube surface mass balance:

$$\Delta\dot{m}_w = \dot{m}_{w\_i} - \dot{m}_{w\_}e \tag{3.53}$$

Water film energy balance:

$$h_{fw\_i} \dot{m}_{w\_i} = \dot{Q}_{ref} + \dot{Q}_{air\_sens} + \Delta\dot{m}_w h_{gw} + \dot{m}_{w\_e} h_{fw\_e} \tag{3.54}$$

Sensible heat transfer on the air side:

$$\dot{Q}_{air\_sens} = htc_{air} A_{ht_{air}} (\bar{T}_{sw} - \bar{T}_{air}) \tag{3.55}$$

Mass transfer on the air side:

If $\bar{w}_{sw} < \bar{w}_{air}$ then $\dot{Q}_{air\_latent} = \Delta\dot{m}_w h_{gw}$ (3.56)

And $\dot{Q}_{air\_latent}$ becomes zero otherwise.

Rearranging the foregoing equations from equation (3.41) to (3.48) yielded the following equations, which are used to calculate the following:

- Enthalpy of air at the exit,
- Humidity ratio of air at the exit,
- Refrigerant enthalpy at exit,
- Sensible and latent heat transfer,

- Heat transfer on the refrigerant side and,
- Tube surface temperature

Humidity ratio change of air:

$$\Delta w_{air} = \frac{\dot{Q}_{air\_latent}}{\dot{m}_{air} h_{gw}} \tag{3.57}$$

Refrigerant side heat transfer rate:

$$\dot{Q}_{ref} = -\dot{Q}_{air_{sens}} - \dot{Q}_{air\_latent} \frac{h_{fgw}}{h_{gw}} \tag{3.58}$$

Water film temperature on the tube surface:

$$\bar{T}_{sw} = \bar{T}_{ref} + \frac{\dot{Q}_{ref}}{A_{ht_{ref}} htc_{ref}} \tag{3.59}$$

3.2.7 Pressure losses

Both fluids encounter pressure drops while passing through the heat exchanger, and these pressure losses are predominantly influenced by the heat exchanger's physical characteristics and are also affected by the velocity of air. In other words, as the number of circuits increases, it becomes increasingly challenging for both the refrigerant and the air to pass through, leading to higher pressure on both sides.

Moreover, as the fluid's velocity increases, it undergoes a corresponding rise in pressure. In other words, it is essential to fine-tune the air supply velocity to prevent substantial pressure losses on the evaporator's airside. Consequently, the analysis of pressure loss assumes a critical role in the design of heat exchangers.

3.2.7.1 Air side pressure losses

In this study, the pressure drops incurred on the air side of the heat exchanger are computed from the Fanning friction factor, $f_F = \frac{f_{DW}}{4}$, where the Darcy-Weisbach friction factor (for turbulent flow) is computed by Swamee and Jain equation[58]:

$$f_{DW} = 0.25(\log(0.27\frac{\epsilon_r}{D_h} + \frac{5.74}{{Re_{air}}^{0.9}}))^{-2} \quad (3.60)$$

And for laminar flow:

$$f_{DW} = 64/Re_{air/ref} \quad (3.61)$$

$$\Delta p_{air} = f_{DW}\frac{L}{D_h}0.5\rho_{air}{V_{air\_max}}^2 = f_F\frac{\alpha L}{\sigma^3}\frac{{\dot{m}_{air}}^2}{2\rho_{air}{A_{af}}^2}\ [55] \quad (3.62)$$

3.2.7.2 Refrigerant side pressure losses

Refrigerant's pressure drops are computed using the Darcy-Weisbach friction factor:

$$\Delta p_{ref} = f_{DW}\frac{L}{D_{h\_i}}0.5\rho_{ref}{V_{ref}}^2 = f_{DW}\frac{\frac{Tubes\ high}{Circuits}Rows\ deep}{D_{h\_i}}0.5\rho_{ref}{V_{ref}}^2 \quad (3.63)$$

3.2.8 Modelling approach

This section presents in detail the Python heat exchanger model that was developed using the fundamentals of thermodynamics and thermofluids which are detailed in the preceding section. The model was developed in Jupyter Notebook, Python 3, on a 64-bit windows 10, core i3 Acer laptop. This model is made up of seven major sections, discussed below:

3.2.8.1 Undistributed air velocity on each tube of the heat exchanger

The model starts by assigning air velocity for each tube, in each row per circuit. That is, for each heat exchanger circuit, there is an array of velocity values showing a significant variation in velocity along the height as well as the depth of the heat exchanger. Undistributed air velocity values were extracted by modelling the heat exchanger in Ansys Fluent using computational fluid dynamics and that procedure is described in section 3.2.4.

3.2.8.2 Geometry parameter calculations

This is the section of the code where all the formulae depicted in section 3.2.1.2 are used to calculate all the heat exchanger parameters which include: heat exchanger air side and tube side heat transfer area, air side efficiency, air flow area, heat exchanger depth and height, frontal area, heat exchanger core volume, air side to tube side heat transfer area ratio.

3.2.8.3 Defining air and refrigerant properties on the heat exchanger inlets and outlets.

In this section, the model assigns dry bulb and wet bulb humid air temperatures, air humidity ratio, mass flow rates on both sides, refrigerant phase and air and refrigerant pressures on both the inlets and outlets for both flows on the heat exchanger. CoolProp Properties [59] as well as Fluid Properties[60] modules were imported into the model and used to calculate air and refrigerant properties.

3.2.8.4 Discretization of the heat exchanger and assigning of fluid properties per increment

This is the section where each tube is discretized into several increments for better resolution as fluid properties such as temperature and pressure vary along the tube length. All the properties for each fluid are defined per increment, that is, any fluid property for an i-th increment is assumed as:

$$\emptyset_i = \frac{\emptyset_{outlet} - \emptyset_{inlet}}{number\ of\ increments} \tag{3.64}$$

Where $\emptyset_i$ is any fluid property: it can be temperature, pressure, or any thermodynamics property. Film or tube surface temperature is initially assumed as the average temperature between the refrigerant and air temperatures per increment. Figure 7 shows how the heat exchanger has been discretized for improved resolution and better accuracy in thermal performance calculations.

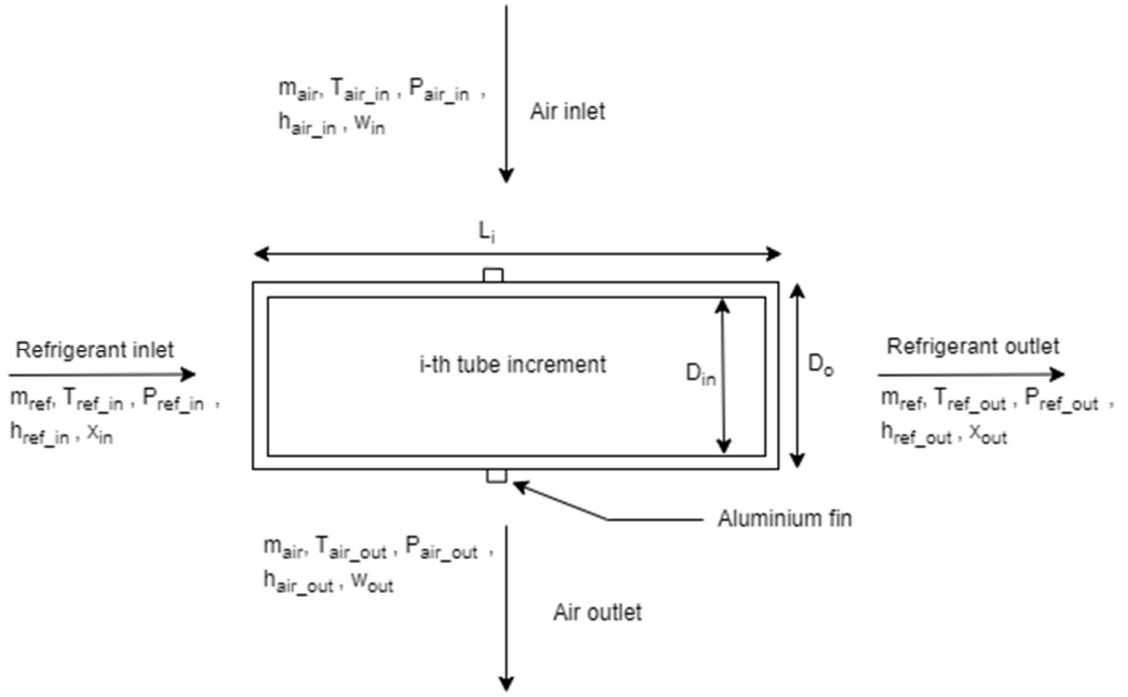


Figure 7: shows an i-th increment of the fin and tube heat exchanger

3.2.8.5 Heat transfer calculations using fluid properties

After discretization of the heat exchanger, each increment is treated as a control volume and the heat transfer rate on an i-th increment in the refrigerant side is given by equation (3.65):

$$\dot{Q}_{ref_i} = \dot{m}_{ref}(h_{ref_i} - h_{ref_{i-1}}) \tag{3.65}$$

And heat transfer on the air side compromises both sensible and latent heat, due to mass and heat transfer. From Figure 7, it can be observed that air and refrigerant are in crossflow, and the inlet of air flow becomes the outlet of the refrigerant on every increment. The sensible heat transfer rate is given in equation (3.65), while latent heat transfer is assumed as zero at the beginning of the model:

$$\dot{Q}_{sens_i} = \dot{m}_{air}(h_{ai\ \ i-1} - h_{air_i}) \tag{3.66}$$

Then the developed model uses these heat transfer rates calculated from assumed properties per increment to check convergence as detailed in section 3.2.8.7.

3.2.8.6 Pressure losses

The model calculates the pressure losses as per equations described in section 3.2.7 for each increment on each side of the heat exchanger. The code is modelled such that it first checks if the refrigerant is single phase or two phase so that it can use the correct correlations for pressure losses.

3.2.8.7 Generation of the loop for convergence

This section of the model recalculates the heat transfer rates by using the overall heat transfer coefficient of the heat exchanger. That is, the model makes use of the correlations stipulated in section 3.2.5 to calculate the heat transfer coefficients for each increment of the heat exchanger.

This is where the convergence of the model is tested by comparing the heat transferred calculated from section 3.2.8.5 with the heat transfer rates calculated using the heat transfer coefficients (discussed in section 3.2.5). That is, if the difference in these heat transfer rates is greater than 1 x 10^{-4}, the model updates the assumed fluid properties which were initially assumed in section 3.2.8.5 with the calculated properties using heat transfer rates which resulted from the overall heat transfer coefficient. The model simplifies the calculations performed on each increment by use of matrices.

For enthalpy of refrigerant for n increments, the following equations can be rearranged:

$$\dot{Q}_{ref_0} = \dot{m}_{ref} h_{ref_0}$$

$$\dot{Q}_{ref_1} = \dot{m}_{ref} (h_{ref_1} - h_{ref_0})$$

$$\dot{Q}_{ref_2} = \dot{m}_{ref} (h_{ref_2} - h_{ref_1})$$

…

$$\dot{Q}_{ref_n} = \dot{m}_{ref} (h_{ref_n} - h_{ref_{n-1}}) \tag{3.67}$$

The model generates matrices using the above equations to solve for refrigerant enthalpy at each increment.

$$\begin{pmatrix} \dot{m}_{ref} & 0 & 0 & 0 & \dots \\ -\dot{m}_{ref} & \dot{m}_{ref} & 0 & 0 & \dots \\ 0 & -\dot{m}_{ref} & \dot{m}_{ref} & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -\dot{m}_{ref} & \dot{m}_{ref} \end{pmatrix} \begin{pmatrix} h_{ref_0} \\ h_{ref_1} \\ h_{ref_2} \\ \vdots \\ h_{ref_n} \end{pmatrix} = \begin{pmatrix} \dot{Q}_{ref_0} \\ \dot{Q}_{ref_1} \\ \dot{Q}_{ref_2} \\ \vdots \\ \dot{Q}_{ref_n} \end{pmatrix} \tag{3.68}$$

The same procedure is followed to calculate refrigerant pressure per increment:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots \\ 1 & -1 & 0 & 0 & \dots \\ 0 & 1 & -1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} P_{ref_0} \\ P_{ref_1} \\ P_{re\ _2} \\ \vdots \\ P_{ref_n} \end{pmatrix} = \begin{pmatrix} \Delta P_{ref_0} \\ \Delta P_{ref_1} \\ \Delta P_{ref_2} \\ \vdots \\ \Delta P_{ref_n} \end{pmatrix} \tag{3.69}$$

On the air side, since the refrigerant outlet on each increment becomes an air inlet and vice versa, the air pressure at each increment is calculated from the following matrix:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots \\ -1 & 1 & 0 & 0 & \dots \\ 0 & -1 & 1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_{air_0} \\ P_{air_1} \\ P_{air_2} \\ \vdots \\ P_{ai\ n} \end{pmatrix} = \begin{pmatrix} \Delta P_{ai\ 0} \\ \Delta P_{ai\ 1} \\ \Delta P_{air_2} \\ \vdots \\ \Delta P_{air_n} \end{pmatrix} \tag{3.70}$$

The air humidity ratio and enthalpy per increment are also solved from matrices (3.71) and (3.72) respectively:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & \dots \\ 1 & -1 & 0 & 0 & \dots \\ 0 & 1 & -1 & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} w_{air_0} \\ w_{air_1} \\ w_{air_2} \\ \vdots \\ w_{air_n} \end{pmatrix} = \begin{pmatrix} \Delta w_{air_0} \\ \Delta w_{air_1} \\ \Delta w_{air_2} \\ \vdots \\ \Delta w_{air_n} \end{pmatrix} \tag{3.71}$$

$$\begin{pmatrix} \dot{m}_{air} & 0 & 0 & 0 & \dots \\ -\dot{m}_{air} & \dot{m}_{air} & 0 & 0 & \dots \\ 0 & -\dot{m}_{air} & \dot{m}_{air} & 0 & \dots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -\dot{m}_{air} & \dot{m}_{air} \end{pmatrix} \begin{pmatrix} h_{air_0} \\ h_{air_1} \\ h_{air_2} \\ \vdots \\ h_{air_n} \end{pmatrix} = \begin{pmatrix} \dot{Q}_{air_0} \\ \dot{Q}_{ai\ 1} \\ \dot{Q}_{ai\ 2} \\ \vdots \\ \dot{Q}_{air_n} \end{pmatrix} \tag{3.72}$$

Enthalpy, humidity ratio, and pressure from the above equations are solved in the model using the mathematical fact that stipulates that:

If X, A and B are matrices and A^{-1} is the inverse of matrix A, then XA = B can be solved by multiplying both sides of the equation with A^{-1} to give $X = A^{-1}B$.

The convergence test continues until the maximum error in these heat transfer rates is less than 1×10^{-4}. When the model converges, it calculates new fluid properties per increment on each tube of the heat exchanger and then recalculates heat transfer rates and therefore prints all the results. The model is developed to perform all these calculations per circuit, on each increment.

Figure 8 below summarizes the evaporator model developed in Python code using a flow chart.

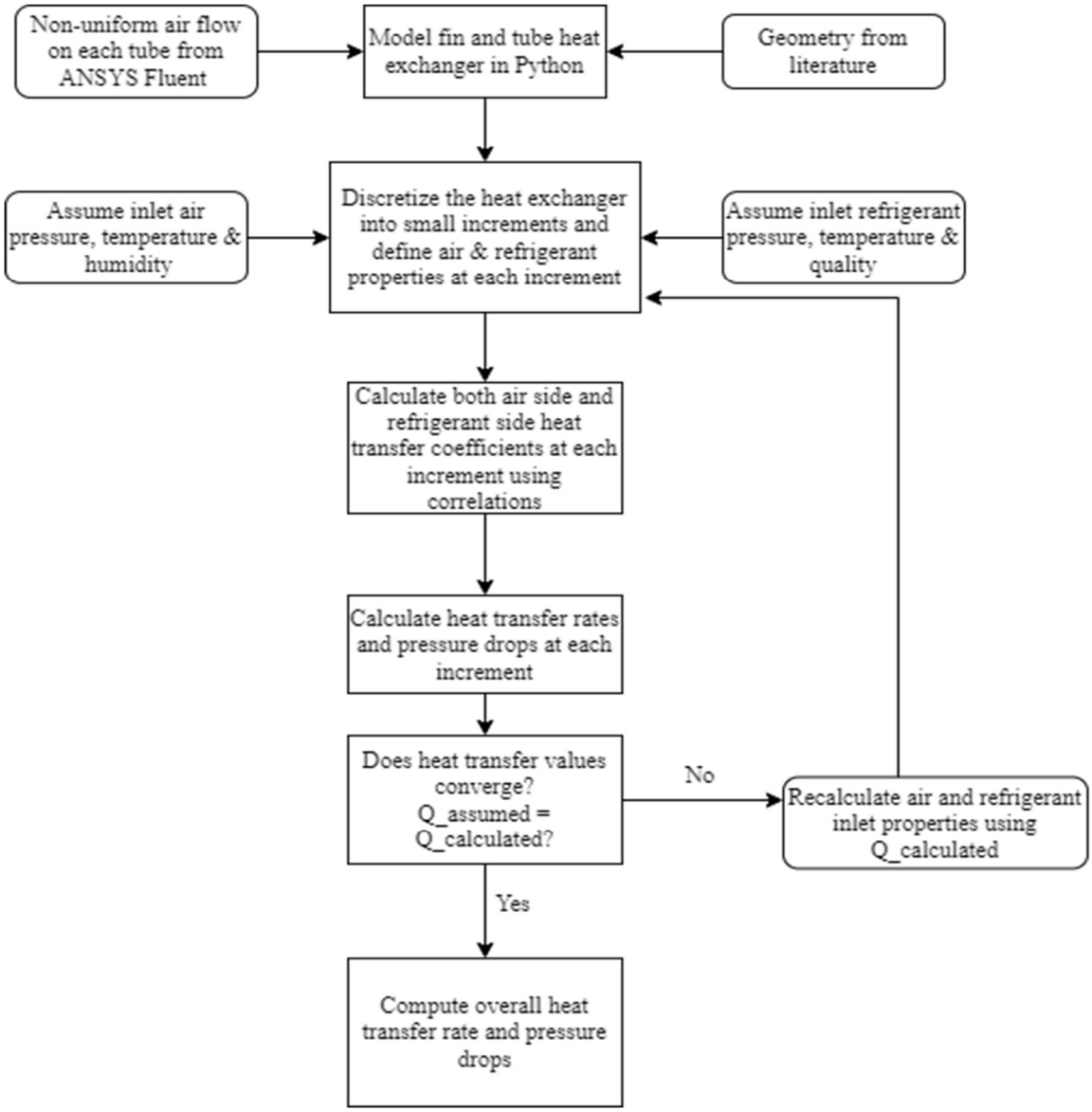


Figure 8: Model development flow chart

3.3 Heat pump system model

This section of Chapter 3 describes a high-level analysis of the heat pump model developed in Jupyter Notebook, Python 3, using Thermal Engineering Systems in Python (TESPy) documentation[61]. This section illustrates the methodology used to integrate the modelled fin and tube heat exchanger into the heat pump cycle to investigate how changes in heat exchanger parameters impact the thermal performance of the heat pump system.

The heat pump modelled in this study has four main components, namely: a compressor, a condenser, an expansion valve, and an evaporator. Even though the heat pump is usually designed to either heat the indoor environment (in winter) or cool the indoor space (in

summer), the study only focuses on the heat pump that is used to heat the indoor space, that is, the integrated heat exchanger from section 3.2 is used in this cycle as an evaporator installed outside.

Figure 9 shows a heat pump cycle developed in a Python model, with all the components and how they are connected to each other. R410a has been used as the working fluid for this cycle.

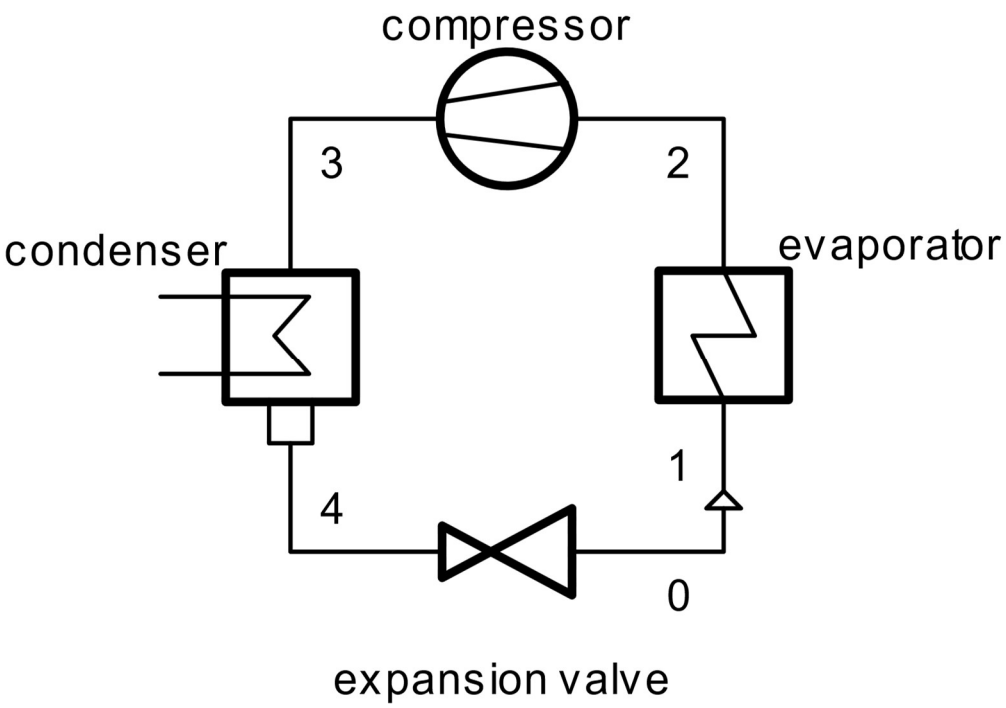


Figure 9: Heat pump system network [62]

Table 5 below also depicts each component specification that has been used in the model to generate a heat pump cycle.

Table 5: Heat pump components specifications[63]

Components	**Specifications**
Expansion valve	Capillary, diameter = 4.76 mm and total length = 240 mm
Evaporator	From the developed model in section 3.2
Condenser	Outer tube diameter = 7 mm, 3 rows, 56 steps, 168 tubes and tube length = 1040 mm
Compressor	Scroll type (inverter), diameter = 7 mm

The developed code which models the heat pump systems is made up of the following steps:

- Import the working fluid, and assign fluid properties units to be used in the model,
- Set up the components which make up the heat pump system and establish connections between the components as shown in Figure 9,
- Set component parameters and calculate to get the results.

There are various system parameters that the TESPy module can accept, however, for the purposes of this study, the developed model parameters used for each component are shown in Table 6.

Table 6: Input parameters for the model

Component	Parameters
Condenser	Pressure ratio, quality, and temperature of the refrigerant at the outlet
Compressor	Efficiency
Evaporator	Overall heat transfer coefficient, pressure ratio and quality and temperature of the refrigerant at the outlet
All components	Mass flow rate

3.3.1 Heat pump performance criteria

Heat transfer rate, often denoted as Q, is a fundamental parameter in the analysis of heat pump performance. It represents the quantity of thermal energy transferred between a heat exchanger and its surroundings or between two fluid streams within the exchanger per unit of time. The heat transfer rate is a crucial criterion for evaluating heat exchanger performance because it directly impacts the efficiency of heat transfer processes, which, in turn, influences the overall effectiveness of the heat exchanger in maintaining desired temperatures or cooling systems.

Moreover, another important parameter used to describe the performance of the heat pump system is COP, which is calculated as the ratio of the indoor unit capacity to the compressor input power as shown in equation (3.70). Thus, the higher the COP implies higher efficiency of the heat pump, and hence lower electrical energy consumption by the compressor.

$$COP = \frac{Indoor\ unit\ capacity}{Compressor\ input\ power} \tag{3.73}$$

3.4 Summary

This chapter discussed in detail, the development of the heat exchanger Python model and the integration of that heat exchanger into the heat pump system as an evaporator using heat and mass transfer correlations, fluid mechanics and CFD. Thus, it described the research design used herein this document to generate the models which accurately represent the performance of both the heat exchanger and heat pump system used for the residential space heating.

4. Model Validation

4.1 Preview

This chapter validates the heat exchanger and heat pump system models developed in Python by comparing the generated results against the known results in the literature. That is, it is in this chapter where the study verifies if the developed models accurately represent the performance of the heat exchanger and heat pump system under study.

For each of the models, different conditions were applied, and a few assumptions were also made, to check the performance of the model and verify if the results agree with those that come from the literature.

4.2 Assumptions made in the models:

- The refrigerant's mass flow rate is equally distributed in all tube circuits of the heat exchanger,
- All refrigerant inlets are kept at one side and are receiving refrigerant at the same temperature, pressure, and quality,
- The refrigerant pressure losses due to tube bends are assumed to be insignificant and therefore ignored,
- Each tube is assumed to have the same surface roughness, both inside and outside,
- Airflow pressure is assumed to be at atmospheric pressure, that is, 101.325 kPa,
- The refrigerant flow inside the copper tubes is fully developed.

4.3 Heat exchanger model

For this model, the four operating conditions tabulated in Chapter 3, Table 4 were applied to compare the generated results with the experimental results reported in the literature. In all these four operating conditions, the heat exchanger was used as an outdoor evaporator, with an air fan installed on the top of the tube circuits. All these conditions, as well as the geometrical parameters, were kept the same as the ones used in [49].

4.3.1 Heat transfer rates

Table 7 below shows the comparison between the heat transfer rates calculated from the model versus experimental heat transfer rates for four operational conditions. This model was set up such that it discretizes the heat exchanger tube by tube, as per literature and that means for validation purposes, heat transfer was calculated per tube and then all the tube

heat transfer rates were algebraically summed to find the total heat transfer by the heat exchanger.

Table 7: Comparison between model heat transfer rate vs heat transfer rates from literature

Operating Condition	**Heat transfer rate from the model (kW)**	**Heat transfer rate from literature (kW) [49]**	**Percentage error**
H1	10.287	9.90	3.91
H2	8.602	8.75	1.69
H3	5.403	5.00	8.06
H4	3.471	3.40	2.09

The comparison in heat transfer rates can also be represented in a graphical way as shown in Figure 10. The maximum percentage error in heat transfer rates calculations was calculated as 8.06 %. Figure 11 shows the heat transferred per circuit for each operating condition. The circuits are numbered from the top of the heat exchanger where the fan is installed.

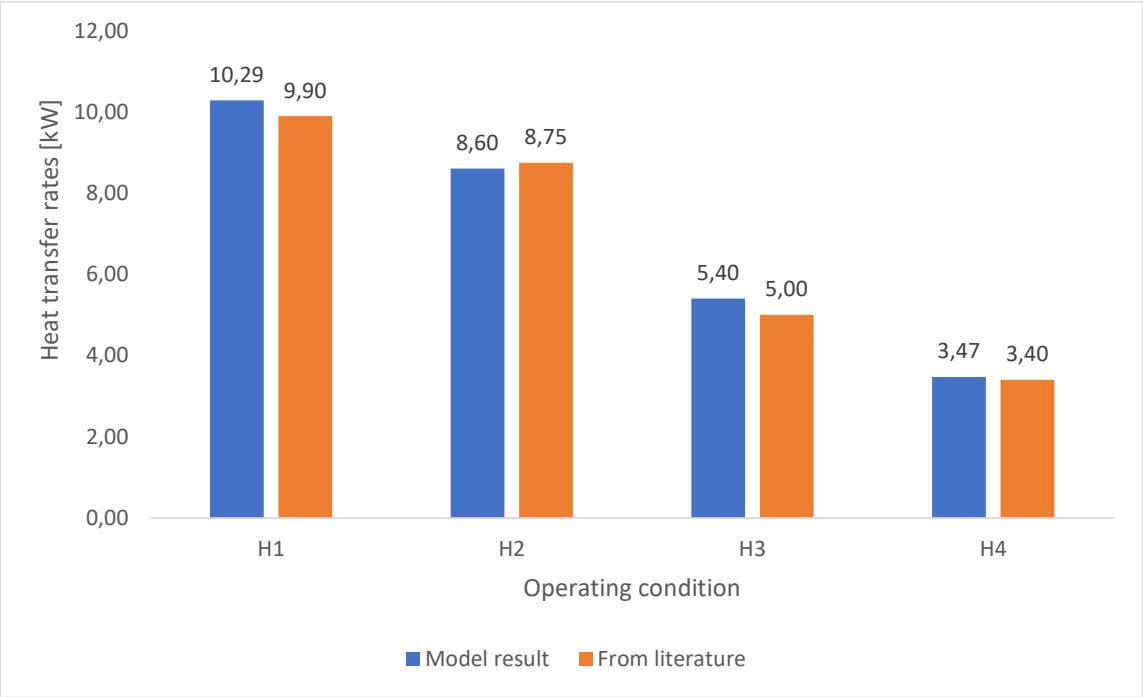


Figure 10:heat transfer rates (model vs literature/experimental) for different models

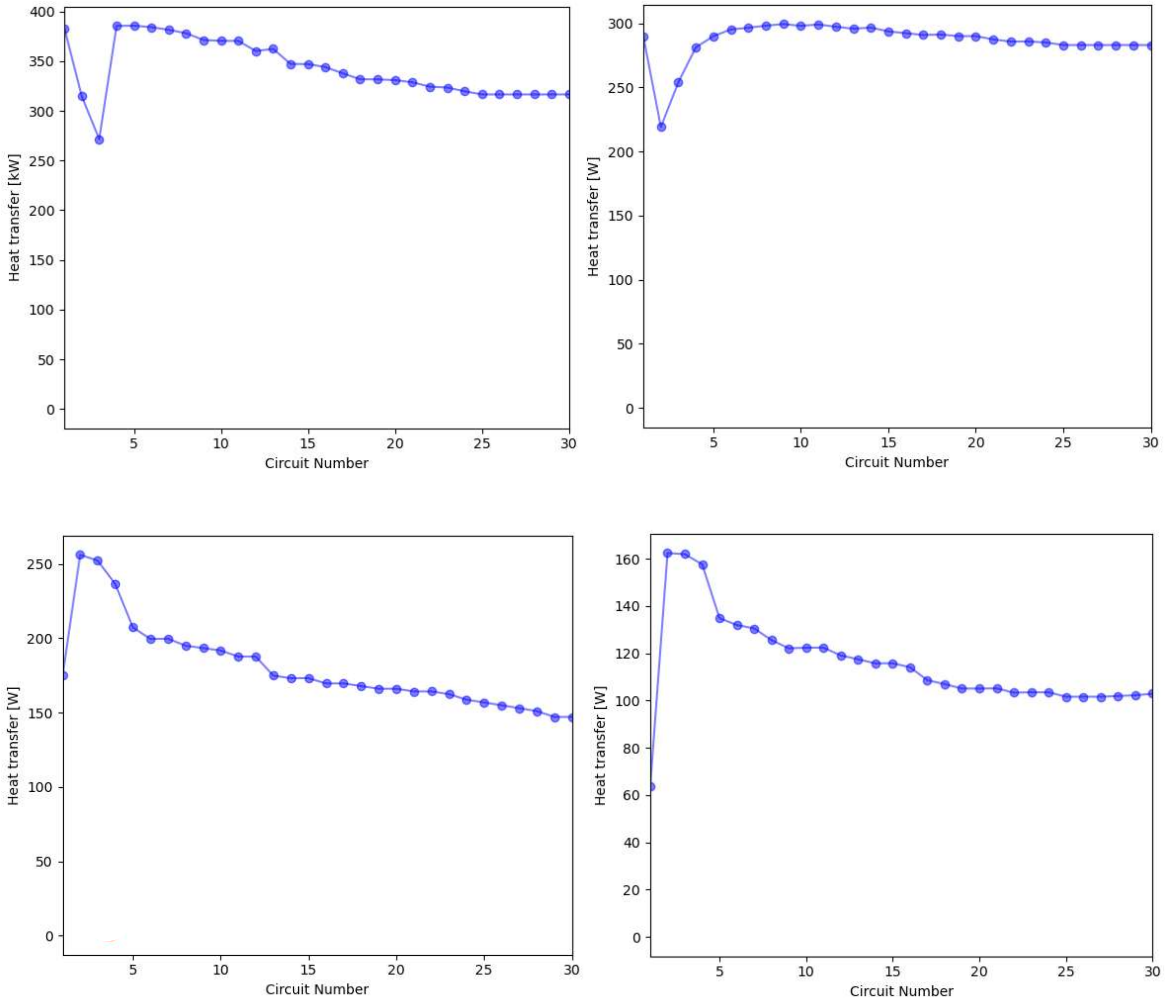


Figure 11: Calculated heat transferred per circuit for each operating condition (top left - H1, top right - H2, bottom left - H3 & bottom right - H4)

4.4 Heat pump model

The heat pump system model that is developed in Python was validated using three heat pump examples given in the Fundamentals of Engineering Thermodynamics textbook [64]. The validations were performed using refrigerant 12 as the working fluid and all pressure losses in both the evaporator and condenser were ignored. Table 8 shows the heat pump system operating conditions that were used as inputs to the developed heat exchanger Python model.

Table 8: Operating conditions of a heat pump system used to validate the model

	Operating Condition 1	**Operating Condition 2**	**Operating Condition 3**
Evaporator exit	Saturated vapour at 20 ºC	Saturated vapour at 12 ºC	Saturated vapour at 12 ºC
Condenser exit	Compressed liquid at 40 ºC	Compressed liquid at 1.4 MPa	Compressed liquid at 1.4 MPa
Compressor	100% isentropic efficiency	100% isentropic efficiency	80% isentropic efficiency
Refrigerant mass flow rate	0.008 kg/s	0.008 kg/s	0.008 kg/s

Table 9 shows the calculated compressor work and evaporator heat transfer rate from both the textbook and the developed model. The maximum variation in percentage based on evaporator heat transfer rate is 7.226%, while the maximum error based on compressor work input is calculated as 1.242%.

Table 9: Model vs literature heat pump system calculations

		Operating Condition 1	**Operating Condition 2**	**Operating Condition 3**	**% max. error**
Compressor work (W)	Model	74.9	163	203	1.242
	Textbook	75	161	202	
Evaporator heat transfer rate (W)	Model	977.0	813.0	813.0	7.226
	Textbook	969.52	808.8	876.32	

The heat pump system model was also validated using the same operating conditions used in the heat exchanger model. Table 6 from Chapter 3 shows the component specifications used for the model validations. Using the operating conditions tabulated in Table 4, the heat pump model was validated based on the heat transfer on the evaporator and therefore the maximum error between the calculated and experimental values is found to be 9.09 %.

Table 10: Operating conditions of the heat pump systems used for validations[49]

Operating Condition	Evaporator Exit Temperature (ºC)	Condenser Exit Temperature (ºC)	Evaporator Pressure Ratio	Calculated Evaporator Heat transfer (kW)	Evaporator Heat transfer from literature (kW)
H1	-9	26.30	0.7989	9.00	9.90
H2	4.5	24.20	0.9743	8.14	8.75
H3	5	22.30	0.9829	4.95	5.00
H4	14.65	21.30	0.9921	3.46	3.40

Figure 12 compares heat transfer rates from the literature against the calculated values, for each condition detailed in Table 4.

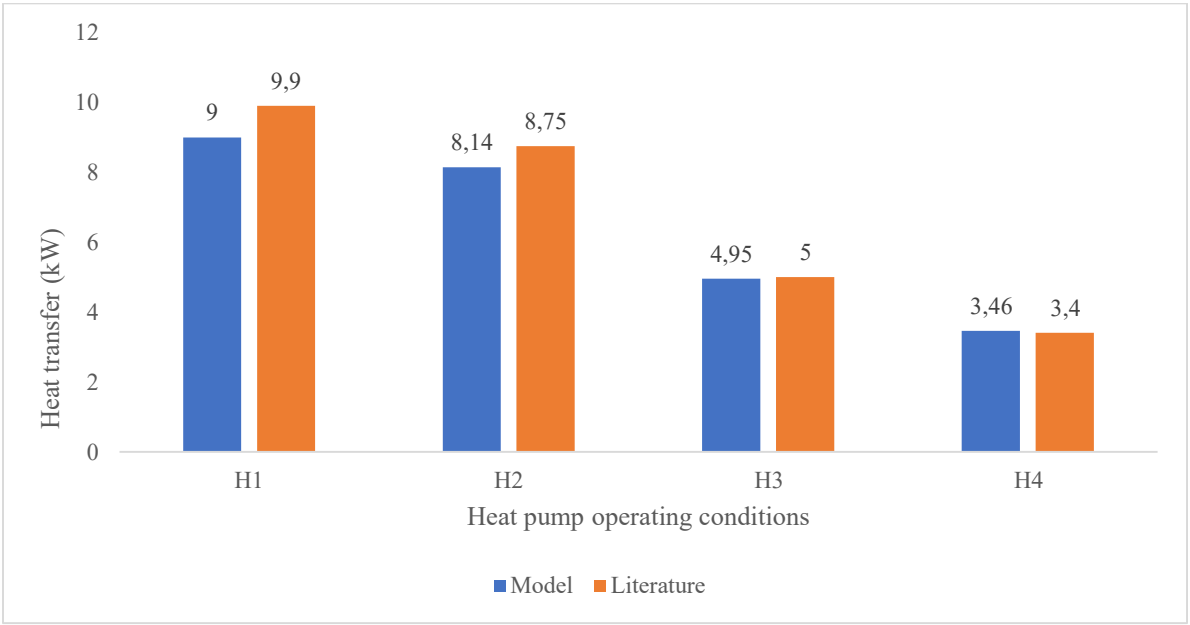


Figure 12: Comparison between calculated evaporator heat transfer and the one from the literature

4.5 Summary

For both models developed, the maximum differences between experimental and calculated values based on heat transfer as well as on compressor work input are all below 9.1 %. This error might have been brought due to different heat transfer coefficient correlations that

have been used and therefore this might have slightly underpredicted calculated heat transfer rates.

The maximum deviations based on heat transfer fall within the 10% margin, and it can therefore be concluded that the developed Python models can represent the performance of the heat exchanger and heat pump system respectively, with less than 10% error. Detailed analysis performed in Chapter 5 provides confidence on the validity of the models.

5. Results and discussions

5.1 Preview

This chapter presents findings from the numerical analysis of the heat exchanger model that was analysed in 3D. Thus, the analysis is made for 3D undistributed air velocity on the discretised heat exchanger and how that nonuniformity of airflow affects the performance of the evaporator, and hence the heat pump system for different operating conditions.

5.2 CFD model

This section describes airflow through a simplified representation of a fin and tube heat exchanger used for a heat pump system. Table 11 depicts the mesh analysis on a representation of the heat exchanger and fluid flow domain, while Table 12 shows details of the setup and solution in Ansys Fluent.

Table 11: Details of heat exchanger representation and fluid flow domain mesh

Physical preference	CFD
Solver preference	Fluent
Element order	Linear
Number of nodes	471,276
Number of elements	411,060

Table 12: Details of the setup and solution

Solver type	Pressure-based
Solver time	Steady-state
Velocity formulation	Absolute
Convergence residuals (all except energy equation)	0.001
Energy equation residual	1e-06

5.2.1 Model convergence

Figure 13 shows the convergence scheme of the continuity equation, velocity components, energy equation as well as k and epsilon. The solution converges on the 10^{th} iteration for all inlet velocities that were used as inputs as the inlet boundary condition. The overall

behaviour of the curves suggests that the energy equation, epsilon, and velocity in the z-axis converged faster than k, other velocity components as well as continuity equation.

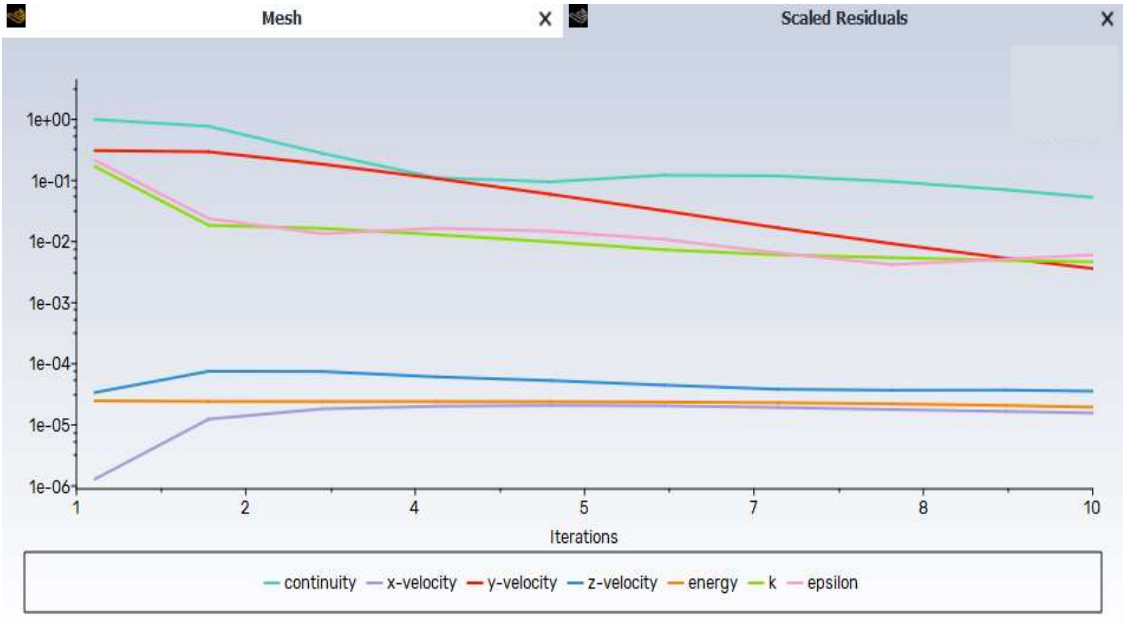


Figure 13: Scaled residuals showing convergence scheme

5.2.2 Undistributed airflow contours

Figure 14 - 19 show velocity contours of 3D undistributed airflow velocity distribution for each of the four operating conditions while keeping all other geometrical parameters and flow conditions the same as discussed in Chapter 3. For each condition, velocity contours were captured for each tube row.

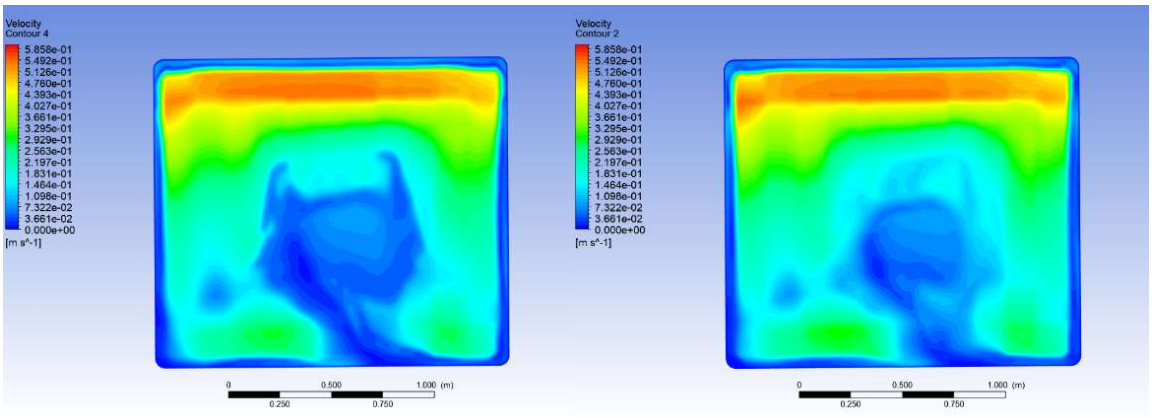


Figure 14: Condition H4, first row (left) and second row (right)

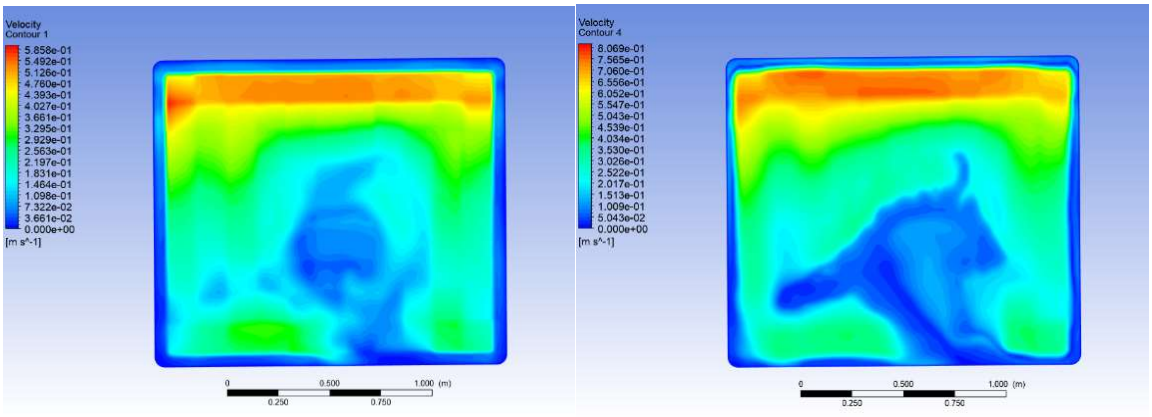


Figure 15: Condition H4, third row (left) H3 and first row (right)

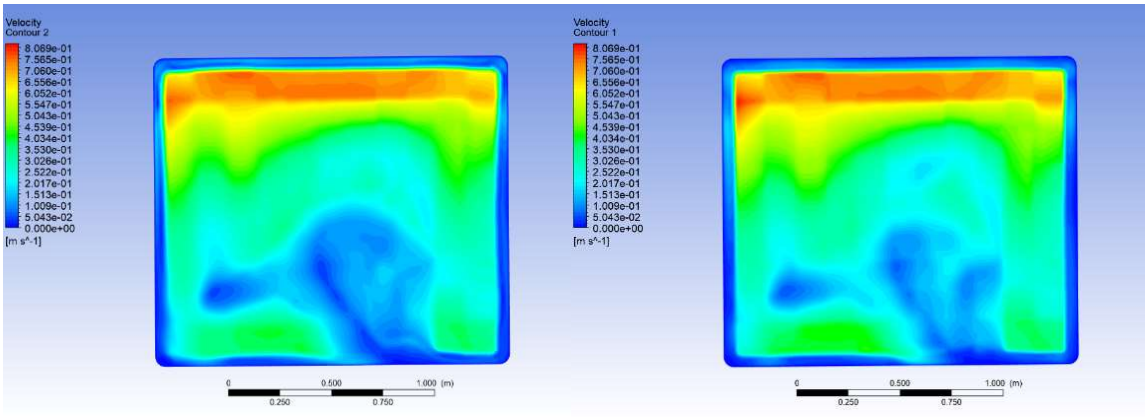


Figure 16: Condition H3, second row (left) & third row (right)

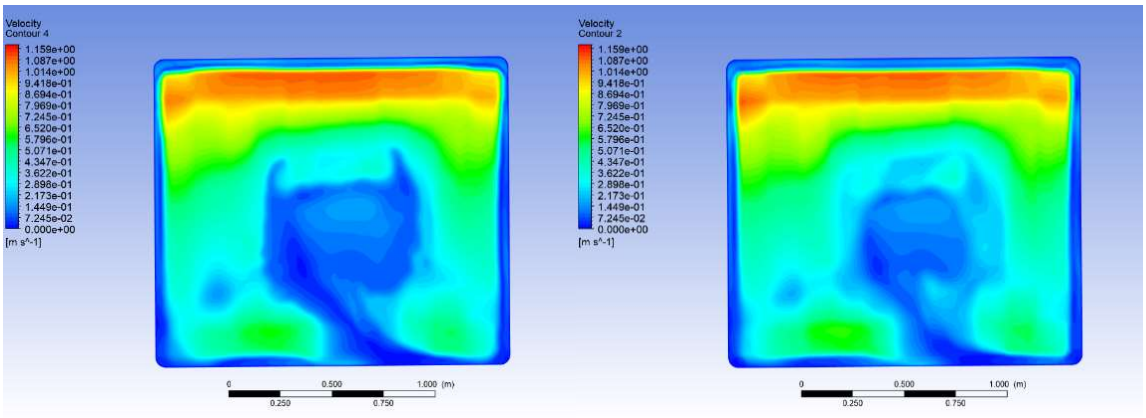


Figure 17: Condition H2, first row (left) & second row (right)

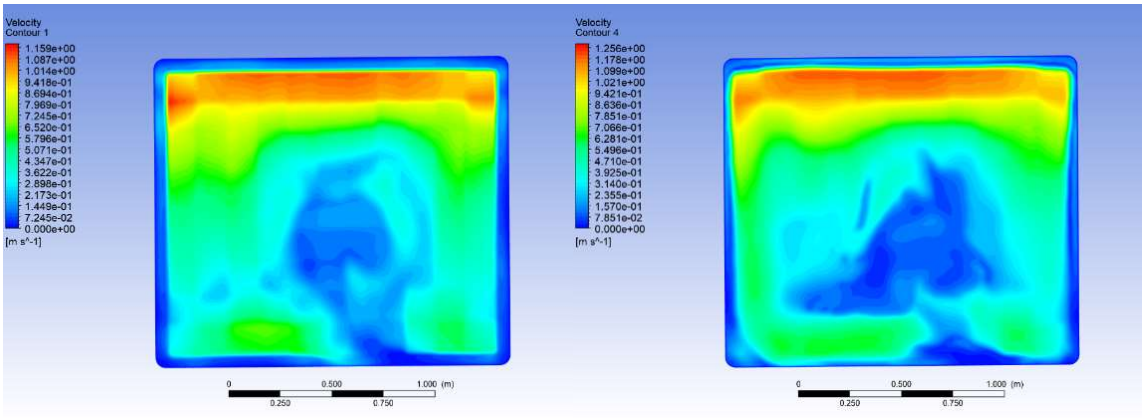


Figure 18: Condition H2, third row (left) & H1, first row (right)

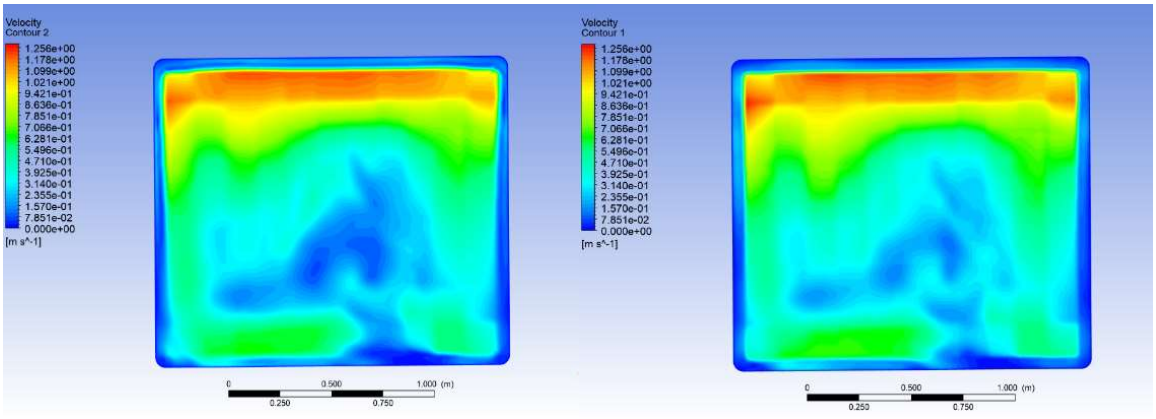


Figure 19: Condition H1, second row (left) & third row (right)

All velocity contours suggest that the velocity varies with not only height but also with the width and length of the heat exchanger. That is, Figure 14 - 19 show velocity contours changing along the x, y and z axes, showing the highest velocity at the top of the heat exchanger, which then gradually decreases with heat exchanger height, and this is because the suction fan is installed on the top of the heat exchanger.

For all these contours, velocity becomes zero on both ends of the heat exchanger width, this is because of the no-slip boundary condition that has been assumed on the walls of the heat exchanger. This boundary condition has therefore resulted in a lower velocity of air towards both ends of the heat exchanger tubes than velocity at the centre, for the same tube.

Also, there is an indication of a significant change in airflow velocity between tube rows. Thus, the first row (from the external side of the heat exchanger), for all operating conditions, has higher velocity magnitudes at the upper part of the heat exchanger than in all other rows. The contours also show that the airflow is more undistributed on row 1 than in row 2 or row 3 for all operating conditions. Additionally, the rate of decrease in velocity magnitude of airflow along the heat exchanger height is highest in row 1, followed by row 2 and lowest in row 3. This variation in uniformity of airflow between tube rows might be because of variation in air pressure losses experienced between the inlet and outlet of airflow in the heat pump.

Moreover, the contours suggest that assuming uniform average face velocity on the heat exchanger slightly underpredicts the air velocity on the top of the heat exchanger and overpredicts it on the lower part of the heat exchanger. For instance, Figure 14 & 15 depict velocity contours for the operating condition whereby the average face velocity of air is 0.5 m/s. The contours suggest that the air velocity is 0.5590 m/s at the top, which is slightly above the average velocity, while the velocity at the bottom of the heat exchanger can be read as approximately 0.4392 m/s, which is slightly less than the average face velocity. A similar trend is observed for all other velocity contours which belong to other operating conditions of the heat pump.

5.3 Heat exchanger Python model

This section details and analyses the results obtained from the heat exchanger model developed in Jupyter Notebook. That is, the section describes the effects of undistributed airflow on the parameters such as temperature and pressure of the refrigerant. The model is developed such that it analyses every circuit as an independent control volume.

5.3.1 Geometry parameters

Table 13 presents calculated parameters of the fin and tube exchanger under study and these parameters are calculated in the Python model. These parameters were kept constant for all four operating conditions described in Chapter 3.

Table 13: Calculated geometrical parameters of heat exchanger

Parameter	Value	Units
Dimensions (height x width x depth)	1.5 x 1.72 x 0.066	m
Frontal area	0.0860	m^2
Core volume	0.005676	m^3
Tube side total heat transfer surface area	0.2043	m^2
Airside total heat transfer surface area	6.107	m^2
Tube side/ air side total heat transfer area ratio	0.03345	-
Overall airside efficiency	95.37	%

The overall heat transfer coefficient was calculated per circuit of tubes as it differs from circuit to circuit due to undistributed air flow along the heat exchanger height, width, and depth, as suggested by velocity contours in section 5.2.2. The Python code results appended in Appendices B, depict the variations in overall heat transfer coefficient (UA) values with circuit number and the significance of increasing discretization increments on the overall UA values.

The UA values are higher in circuits exposed to higher velocity towards the top of the heat exchanger than the UA values for lower tube circuits. This is because the UA value of the heat exchanger is dependent on the airside heat transfer coefficient, which in turn is dependent on the air velocity. Thus, the higher the velocity of airflow on the refrigerant tubes significantly increases the ability to transfer heat from air to refrigerant inside the tubes. Therefore, the calculated overall UA values, shown in Appendices B suggest that assuming average velocity over the whole heat exchanger overpredicts the overall heat transfer coefficients of lower circuits (far from the fan location) while underpredicting the overall heat transfer coefficients of the upper tubes near the fan.

5.3.2 Non-uniform versus uniform airflow heat transfer capacity along heat exchanger height

The impacts of undistributed airflow velocity along the heat exchanger height (2D) on the heat transfer capacity are shown in Figure 20 & 21 below. The figures depicted four operating conditions.

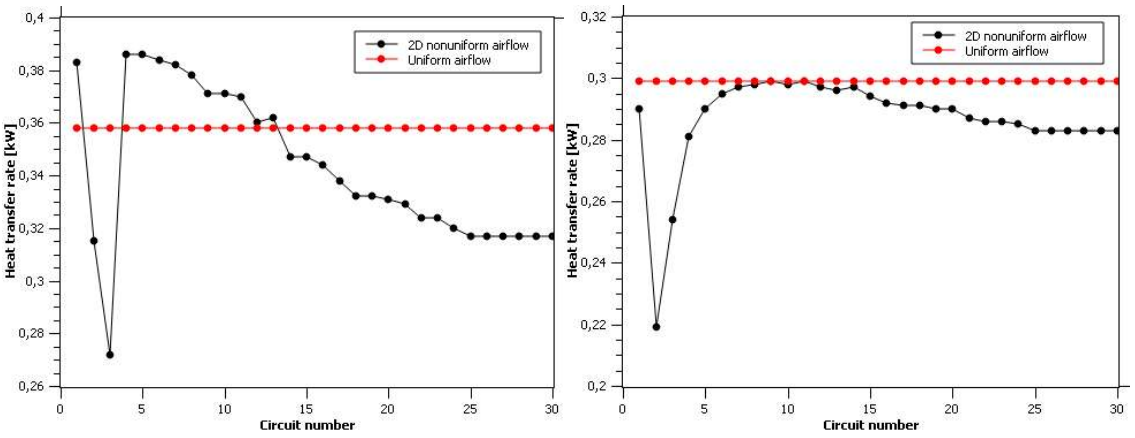


Figure 20: Condition 1(left) & 2 (right) – 2D Heat transfer capacity

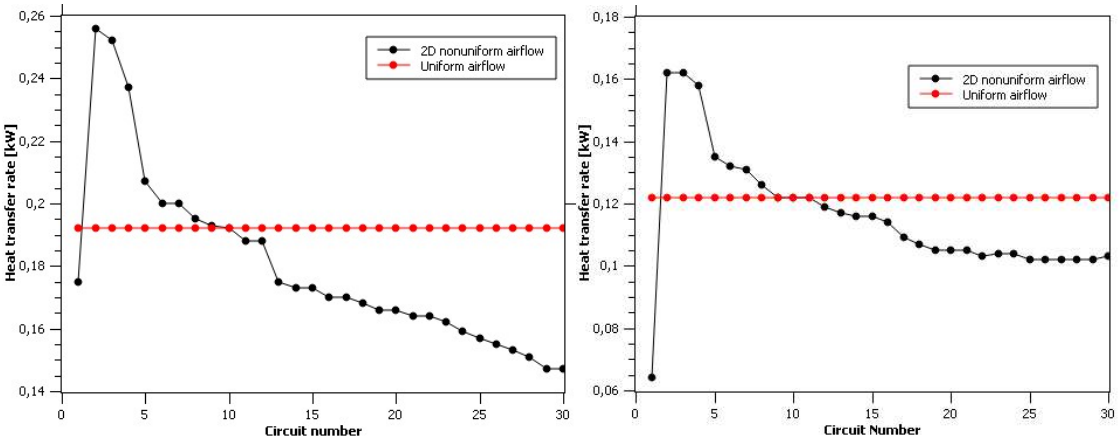


Figure 21: Condition 3 (left) & 4 (right) – 2D Heat transfer capacity

Figure 20 & 21 suggest that assuming uniform airflow along heat exchanger height underpredicts the actual heat transfer rates from the top circuits closer to the fan while overestimating the heat transfer rates from the lower tube circuits away from the fan where velocity has decreased. This is because the heat transfer rate is dependent on the velocity of air, that is, the higher the flow rate of air, the more heat is transferred to the refrigerant inside the tube. Therefore, higher heat transfer rates at higher tube circuits result from increased velocity and this heat transfer rate generally decreases gradually with heat exchanger height due to a slight decrease in velocity of air.

However, Figure 20 & 21, depict that assuming uniform airflow along heat exchanger height overestimates the actual heat transfer rates at higher tube circuits which are closer to the fans. This is observed by low peak heat transfer rates that drop below the estimated heat transfer rates for uniform air flow, at upper tube circuits. This might be because operating conditions 1 and 2 have inlet air at dry-bulb temperatures of -7 and 2 degrees

Celsius, respectively. Therefore, if airflow has a very low temperature, the rate at which heat is transferred to the refrigerant will decrease even if the velocity of air is high.

5.3.3 Effects of non-uniform airflow on refrigerant entropy

Figure 22 & 23 show refrigerant entropy at the exit of every circuit for four operating conditions of the heat exchanger. As the second law of thermodynamics states, entropy (state of disorder of fluid molecules) can either increase or remain the same. Thus, entropy increases for irreversible processes where there are irreversibilities such as friction, mixing of two fluids and heat transfer across finite temperature differences [65].

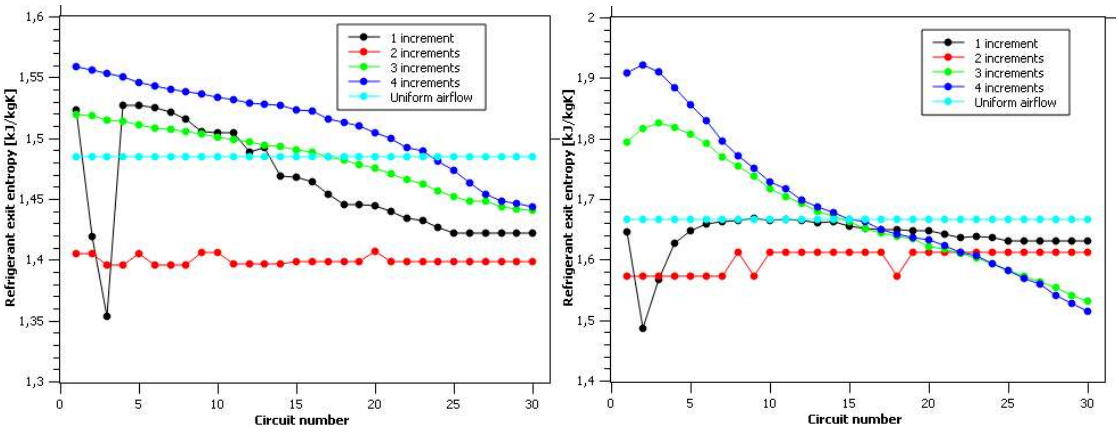


Figure 22: Refrigerant exit entropy H1 (left) and H2 (right)

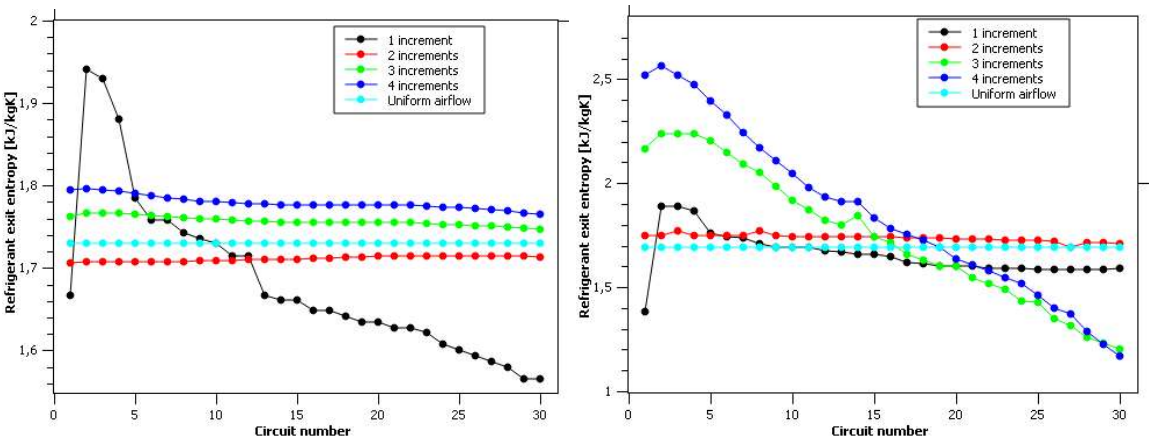


Figure 23: Refrigerant exit entropy H3 (left) and H4 (right)

Figure 22 & 23 show that when discretising every finned tube of the heat exchanger into several increments, refrigerant entropy varies at each tube circuit of the heat exchanger. That is, for each circuit, the entropy at the inlet was kept the same as 0.932 kJ/kgK, 0.985 kJ/kgK, 1.014 kJ/kgK and 1.036 kJ/kgK for operating conditions 1, 2, 3 and 4 respectively. The maximum variation between uniform airflow and discretising the tube into four

increments is calculated to be 4.75%, 13.22 %, 3.67 %, and 44.82 % for operating conditions 1, 2, 3 and 4 respectively.

The refrigerant entropy plots for each heat exchanger circuit exit suggest that assuming uniform airflow underpredicts the actual entropy for higher tube circuits while overpredicting refrigerant entropy for lower tube circuits along the heat exchanger height. This observation might be because at higher tube circuits, the higher the velocity of airflow, the higher the rate of heat transfer between air and refrigerant, that is, across finite temperature differences and hence increasing irreversibilities in the refrigerant flow. Therefore, lower entropy on lower circuits results due to reduced airflow velocity which results in reduced irreversibilities.

5.3.4 Influence of tube discretization on heat transfer capacity

To further investigate the impacts of nonuniform airflow on the heat exchanger, the study analysed the heat transfer rates per each small increment of each tube in each circuit. This is because the airflow varies not only with heat exchanger height, but it varies also with the width and depth of the heat exchanger (3D) (as depicted in section 5.2.2), which consequently affects the heat transfer rates. Figure 24 & 25 therefore, depict the heat transfer rate calculated for different numbers of increments per tube during discretizing, for better resolution.

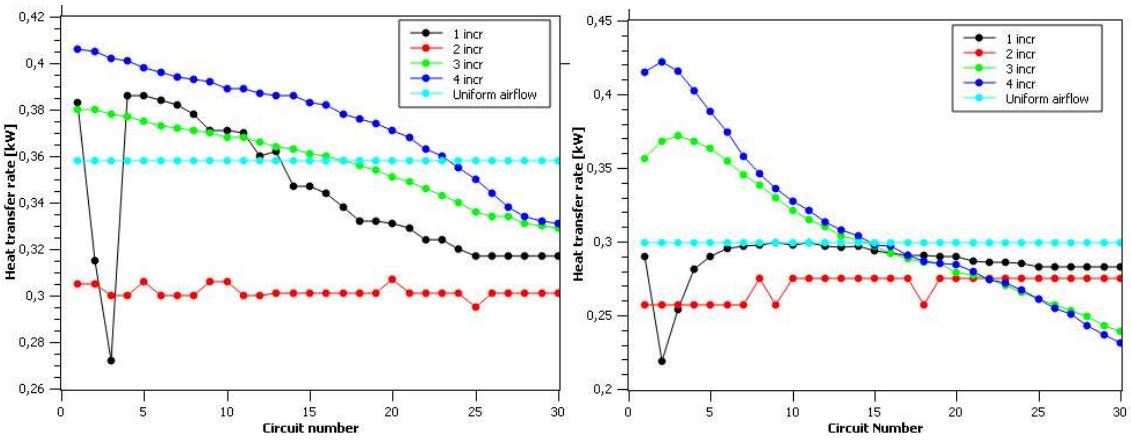


Figure 24: Condition 1 (left) & 2 (right) - 3D heat transfer rates

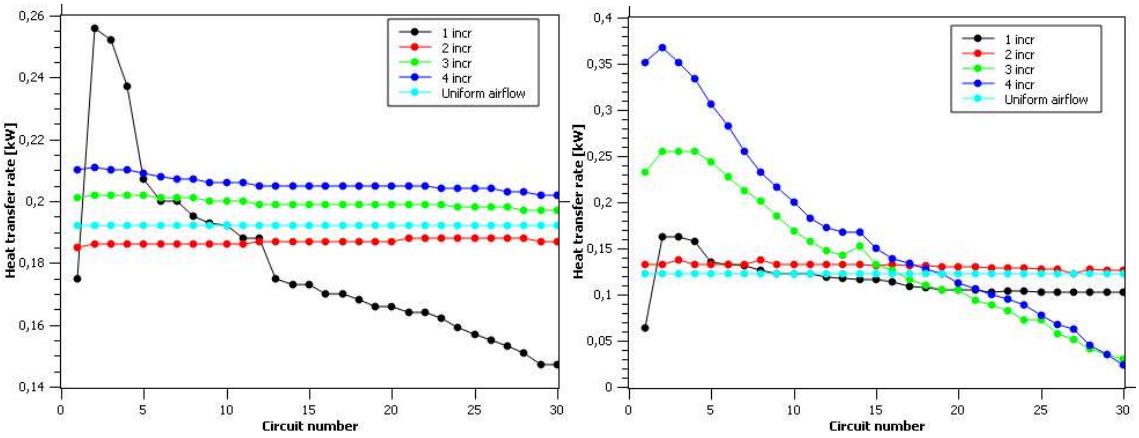


Figure 25: Condition 3 (left) & 4 (right) - 3D heat transfer rates

In general, Figure 24 & 25 show a gradual decrease in heat transfer rate from higher to lower tube circuits due to the decreasing velocity. Other than that, all graphs suggest that the heat transfer rate calculated when discretizing each tube into four increments results in higher heat transfer rates than for lower increments, particularly on higher tube circuits next to the fan. Thus, the higher the discretization increment number per tube, the higher the heat transfer rates on higher circuits. This shows that assuming a uniform air velocity over a refrigerant tube underpredicts the actual heat transfer by the tubes on circuits next to the fan while overpredicting the actual heat transfer by the tubes far away from the fan, especially from circuit number 20 to 30.

Figure 25 shows a significant difference in calculated heat transfer rates between discretising a tube in four increments and assuming uniform air velocity, especially for the upper circuits and lower circuits. The variation in heat transfer capacity calculated from discretisation (four increments per pipe) from the one calculated with uniform air velocity over a tube is observed to be 27.85 % for condition 4. The higher variation in heat transfer

capacity might have been caused by higher airflow velocity, and therefore the higher air velocity results in a higher heat transfer rate.

In other words, the significant variation in heat transfer capacity calculated from the discretization of each tube in four increments from the one calculated with uniform air velocity over a tube suggests an inadequate prediction of the heat transfer capacity with the assumption of uniform airflow over the refrigerant tubes.

5.3.5 Influence of tube discretization on refrigerant pressure losses

Figure 26 & 27 show that nonuniform airflow also affect the refrigerant pressure losses inside the pipes. It is evident from the plots that assuming uniform airflow underpredicts the actual pressure losses for most tube circuits. Also, the pressure losses decrease for lower circuits, indicating that the refrigerant pressure losses decrease with airflow velocity. That is the higher magnitude of airflow velocity results in higher friction between the refrigerant and the tube's inner surface.

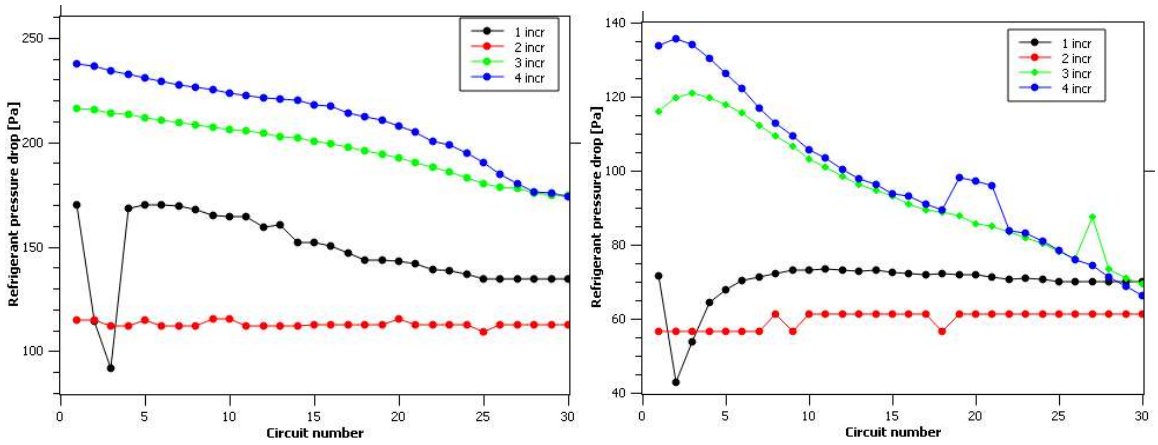


Figure 26: Condition 1 (left) & 2 (right) - Refrigerant pressure losses

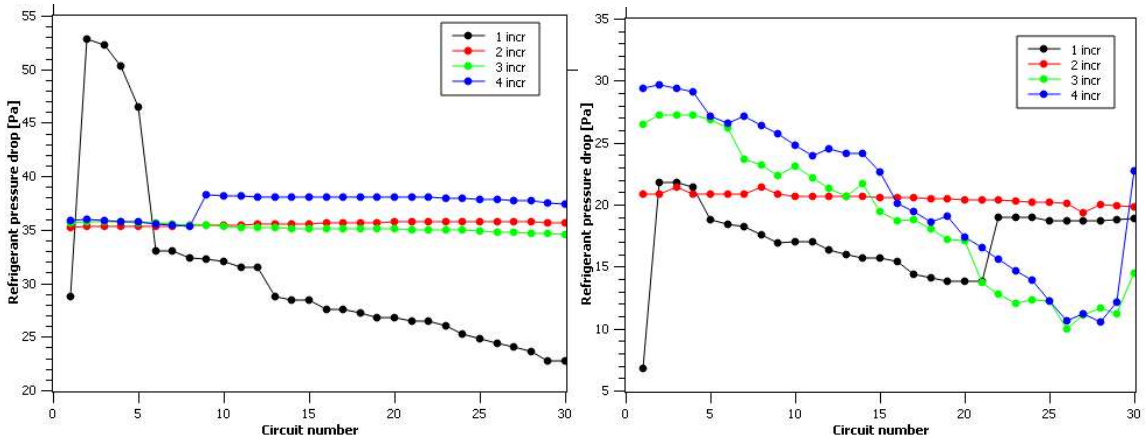


Figure 27: Condition 3 (left) & 4 (right) - Refrigerant pressure losses

Similar to heat transfer capacity, it can be observed herein these plots that assuming uniform airflow over the refrigerant tubes underpredict the actual pressure losses for most tubes of the heat exchanger. The percentage variations between the refrigerant pressure losses calculated by discretizing each tube in four increments and the pressure losses calculated by assuming uniform airflow over each tube are 30.14%, 29.62%, 17.52%, 18.26% for conditions 1, 2, 3 and 4 respectively. The maximum variation in pressure losses is observed from condition 1 where the inlet airflow velocity is higher than in other conditions. These percentage variations are major and therefore highlight the importance of analysing the nonuniform airflow on heat exchangers in 3D instead of assuming uniform airflow on each tube.

The plots depict some discrepancies in refrigerant pressure losses for some tube circuits along the heat exchanger height. These might be caused by the varying air velocity magnitudes along the circuits. Also, the airflow velocity contours depicted in Figure 14 & 15 for operating condition 4 suggest that the velocity variation is more significant than in

other operating conditions and therefore this has caused significant variations in refrigerant pressure losses over the circuits on condition 4 than in any other operating condition, as shown in Figure 27.

Apart from that, the plots show that the refrigerant experiences the highest-pressure losses while the heat exchanger is working under condition 1, followed by condition 2, then condition 3, and lastly condition 4, with the least pressure losses. This indicates that the refrigerant pressure loss is also affected by the refrigerant mass flow rate, thus the higher the mass flow rate, the higher the pressure losses experienced by the refrigerant.

5.3.6 Influence of tube discretization on refrigerant exit vapour quality

The refrigerant vapour quality determination is crucial in heat exchanger performance analysis as it helps in identifying flow regimes inside the refrigerant tubes. Vapour quality is the measure of the amount of vapor available in the refrigerant and it is denoted by the letter x. That is, subcooled refrigerant theoretically has a vapour quality of zero ($x = 0.0$), while any vapour quality between zero and one ($0.0 < x < 1.0$) indicates two–phase fluid, that is, a mixture of vapour and liquid refrigerant. Vapor quality, $x = 1.0$ denotes superheated fluid region.

In evaporators, it is expected to have superheated refrigerant at the outlet of every circuit so that maximum heat transfer can be transferred between airflow and refrigerant. Figure 28 to 29 show plots of exit refrigerant vapour quality per circuit for different operating conditions in 3D undistributed airflow velocity. In general, there is a gradual decrease in exit refrigerant vapour quality for lower tube circuits (except for condition 3) due to decreasing airflow velocity.

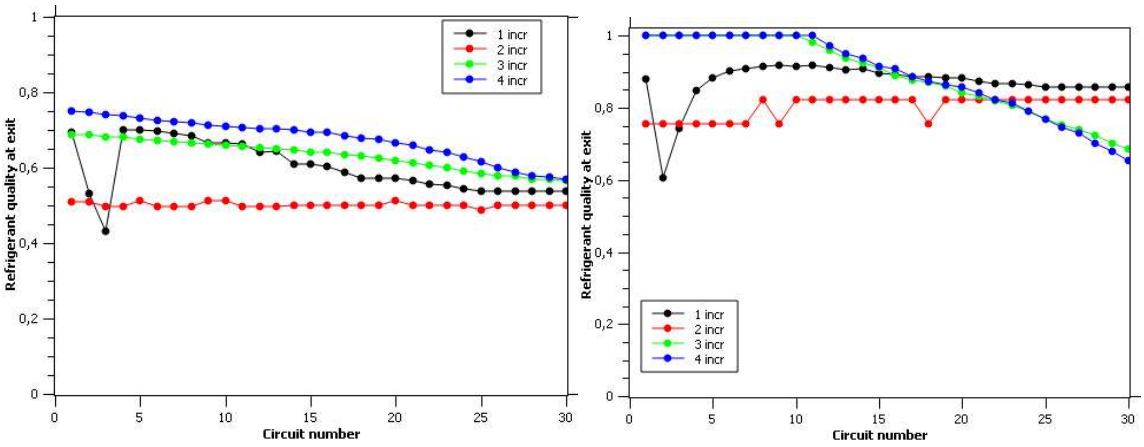


Figure 28: Condition 1 (left) & 2 (right) - Refrigerant exit vapour quality

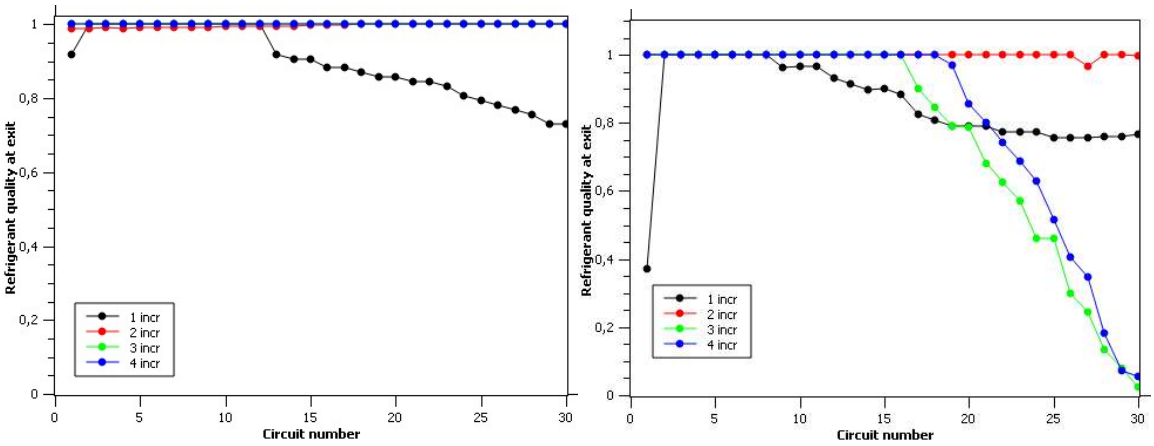


Figure 29: Condition 3 (left) & 4 (right) - Refrigerant exit vapour quality

Moreover, Figure 28 suggests that estimating uniform airflow over each tube underpredicts the actual exit quality calculated by discretizing each tube for all circuits, while Figure 29 depicts the same behaviour on the lower circuits (circuit number 12 to 30). This means that the higher airflow velocity results in increased refrigerant vapour in the tubes because more heat is being transferred.

Apart from that, Figure 28 & 29 show that assuming uniform airflow velocity per tube overestimates the exit quality of the refrigerant on lower circuits (circuit number 20 to 30). For these analyses, the refrigerant inlet vapour quality was kept the same. The substantial difference in exit vapour quality when assuming uniform airflow and when discretizing continues to demonstrate the importance of performing 3D nonuniform airflow analysis.

The rapid decrease in refrigerant exit vapour quality for operating condition four shown in Figure 29, suggests that the heat absorbed by the refrigerant from airflow is inadequate and less than the latent heat of vaporisation and therefore cannot change the flow regime from

a two-phase state to superheated phase at lower circuits, resulting in reduced efficiency of the heat exchanger.

5.3.7 Refrigerant saturation temperature

Figure 30 shows refrigerant saturation temperatures for different operating conditions of a heat exchanger. The saturation temperature also known as the boiling point, of a refrigerant is the temperature at which the refrigerant changes from liquid to gas. The temperature does not change during this process. From the plot, operating condition four has a higher saturation temperature, that is, the refrigerant under this operation boils at a higher temperature than in other operating conditions. The refrigerant saturation temperatures are 39.9 ºC, 34.6 ºC, 28.1 ºC and 15.9 ºC for operating conditions four, three, two and one, respectively.

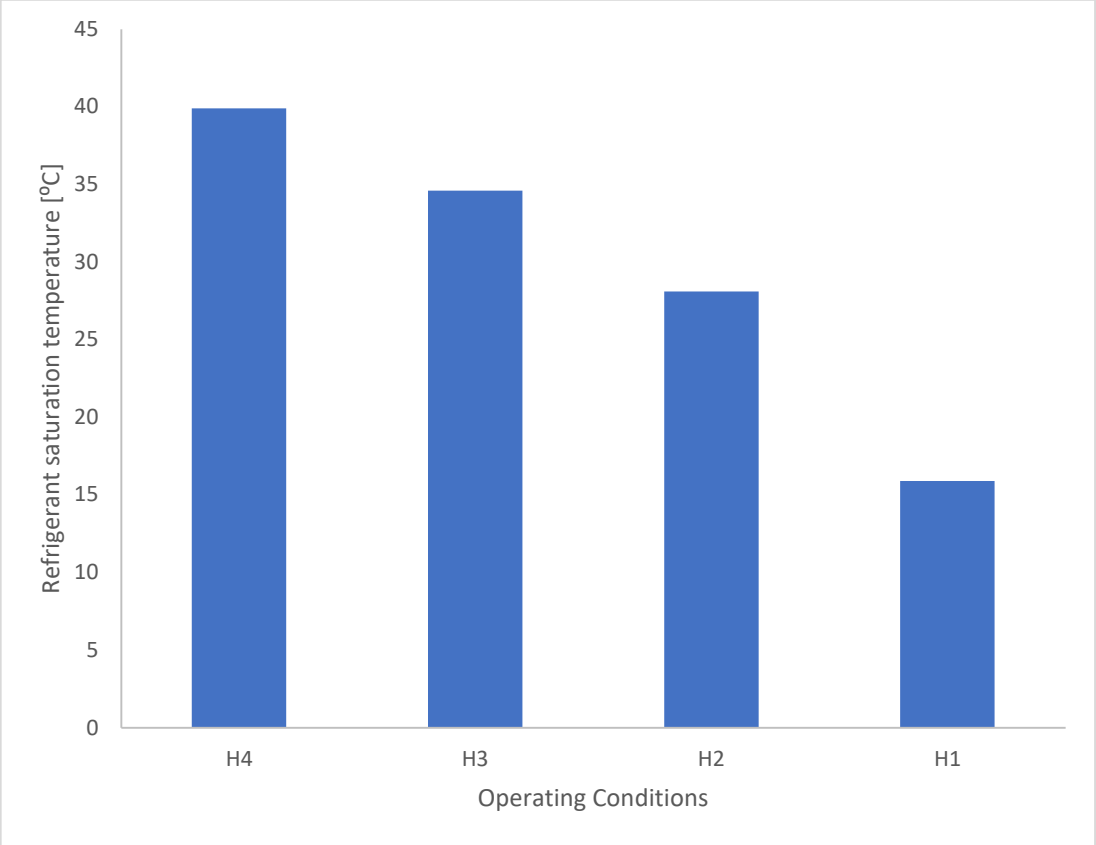


Figure 30: Refrigerant saturation temperatures for different operating conditions

The variation in these saturation temperatures deduces that the boiling point of the refrigerant inside the tubes depends on the refrigerant pressure, mass flow rate on both sides as well as the inlet temperature of the refrigerant. Thus, low airflow velocity on the refrigerant tube results in less heat transfer between air and the refrigerant and hence a much higher refrigerant temperature is required to change liquid refrigerant into vapour.

5.4 Heat pump system model

This section presents the results obtained from the heat pump system model in which the modelled heat exchanger is integrated. That is, the section analyses the influence of nonuniform airflow velocity on the thermal performance of the heat pump system which is used for residential air conditioning purposes.

Figure 31 to 34 illustrate the heat pump system representation in a p-h diagram for different operating conditions. The vapour compression heat pump system has four main processes: compression, condensation, expansion, and evaporation. Furthermore, it can be deduced that there is a smaller difference in refrigerant enthalpy on condition one when compared with other operating conditions and therefore this means that the COP from this operating condition is less than in other conditions.

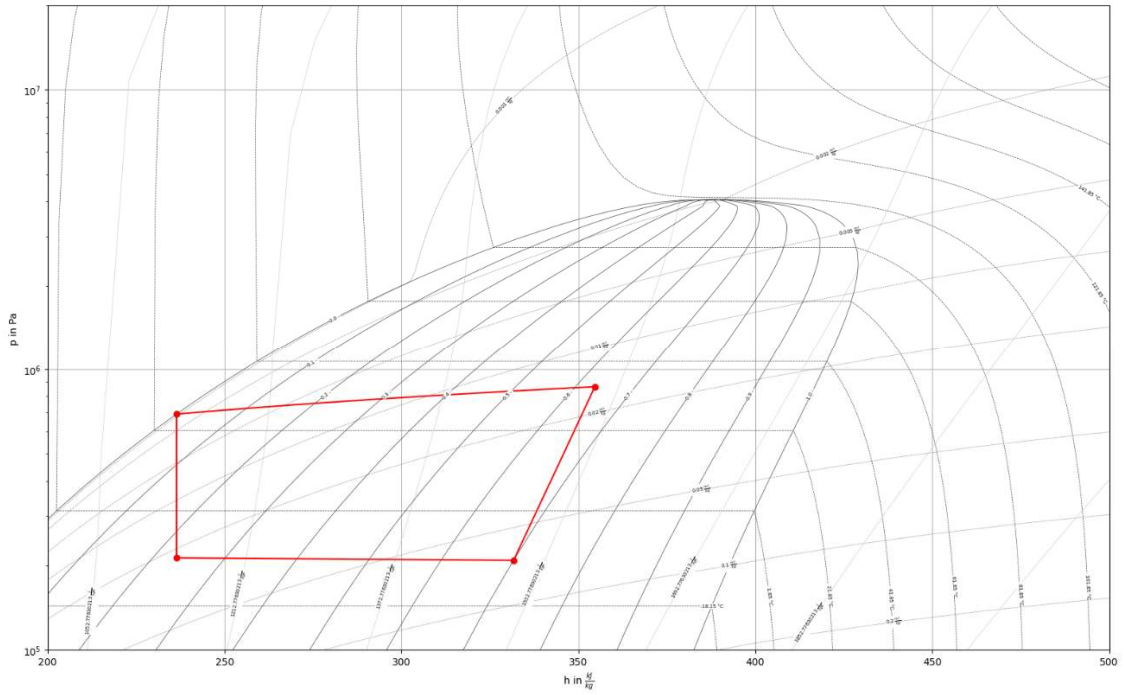


Figure 31: The heat pump system illustrated in a p-H diagram - H1

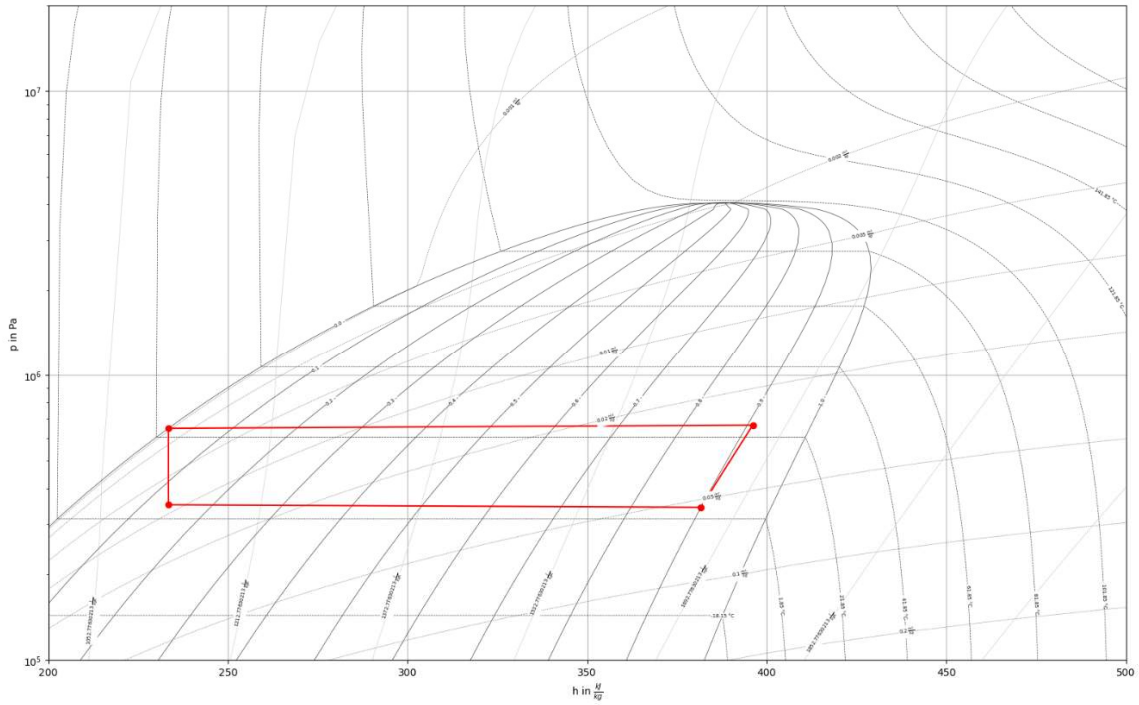


Figure 32: The heat pump system illustrated in a p-H diagram – H2

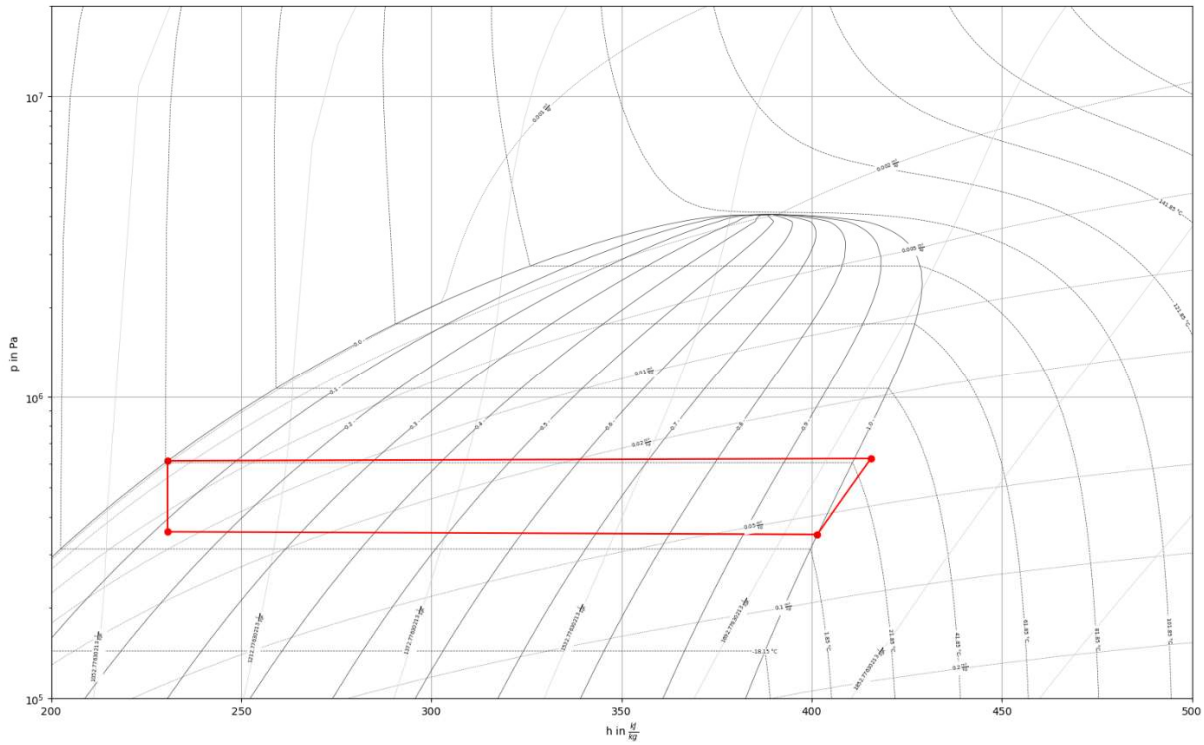


Figure 33: The heat pump system illustrated in a p-H diagram – H3

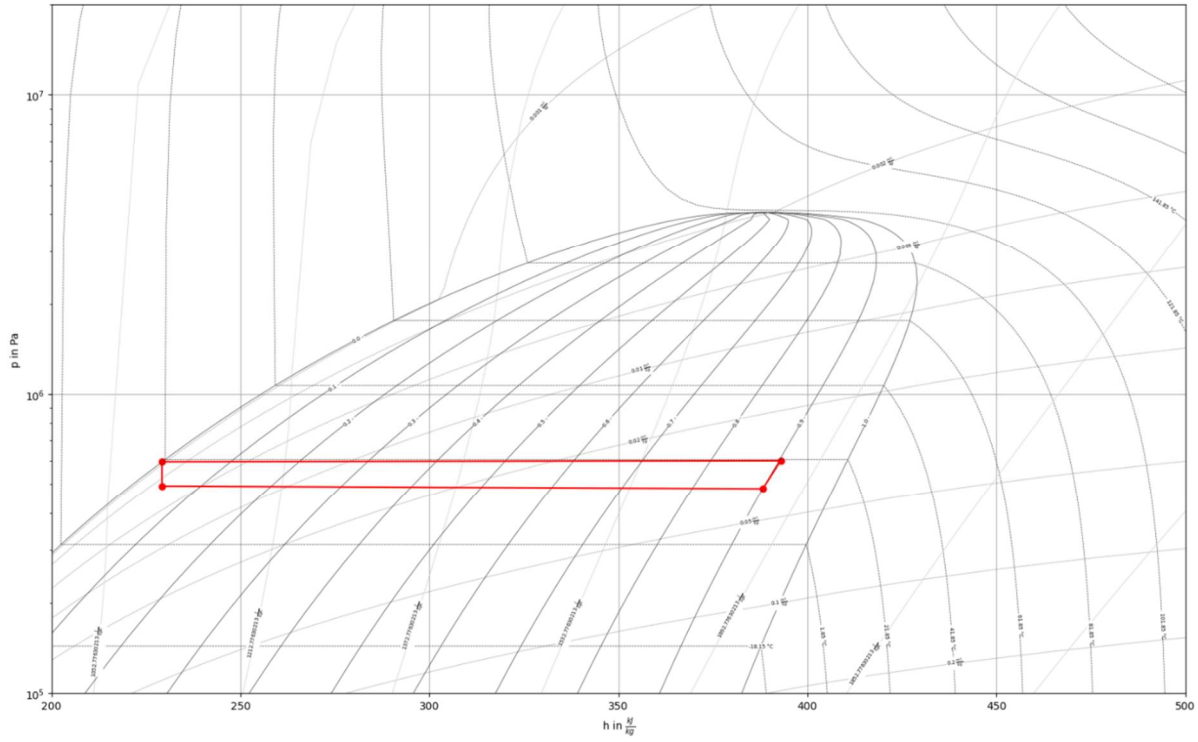


Figure 34: The heat pump system illustrated in a p-H diagram – H4

5.4.1 Effects of 3D undistributed airflow velocity on the COP

The results of the heat pump model showing how the nonuniform airflow velocity influences COP are shown in Figure 35. In general, Figure 35 depicts that assuming that

airflow velocity is uniform throughout each tube length underestimates the actual COP of the heat pump. This is due to the underpredicting and overpredicting of airflow velocity on the higher tubes and lower tubes of the heat exchanger, respectively, as discussed in the previous section.

In other words, the graph suggests that analysing the heat exchanger performance by discretizing each tube of the heat exchanger to a higher number of increments results in a significantly different COP from that of assuming uniform airflow, especially for operating conditions 2 and 4. That is, assuming uniform airflow velocity results in COP variations of 2.73 %, 10.68%, 4.03% and 11.07% for operating conditions 1, 2, 3 and 4, respectively. This might be because the lower velocity of airflow assumed by assuming uniform airflow resulted in lower heat transfer between air and refrigerant in the heat exchanger, and this leads to lower heat exchanger capacity and therefore lower heat pump system COP results.

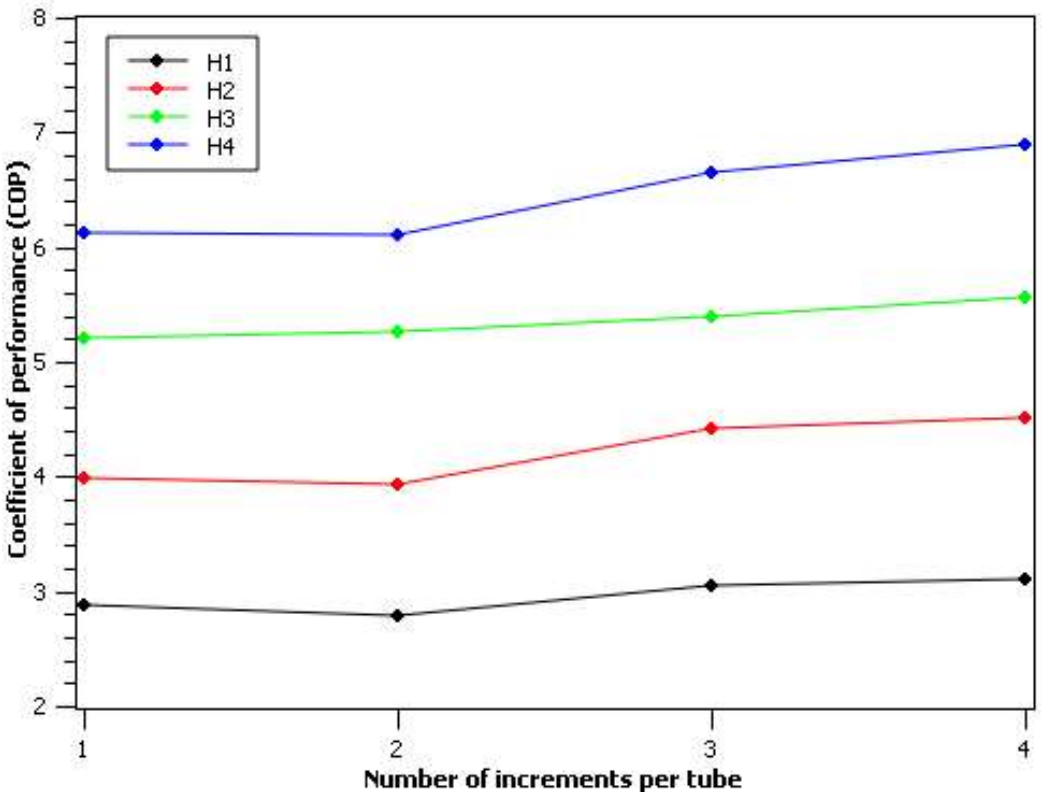


Figure 35: Tube discretization increments vs COP

Furthermore, Figure 35 shows that the heat pump system under study operates best under condition 4. This is because of the increased ambient temperature which leads to a higher temperature gradient between airflow and the refrigerant inside the heat exchanger tubes. Thus, the higher temperature gradient results in higher heat exchanger capacity and hence higher COP of the heat pump system.

5.5 Off design parameters

This section examines the performance of the modelled finned heat exchanger when subjected to off-design parameters. Specifically, variations were introduced on the inlet refrigerant pressure and temperature, to assess how these changes affect the heat exchanger's performance. These parameter variations were conducted while maintaining a fixed geometry.

5.5.1 Effects of variation in inlet refrigerant pressure on pressure drop

Figure 36 & 37 below show the effects of variation of inlet refrigerant pressure on the total pressure losses in the heat exchanger tubes. The inlet refrigerant pressure was varied between 250 kPa and 1500 kPa for all four operating conditions of the heat exchanger and all other parameters were kept constant. For non-uniform airflow, the model considered four increments per tube.

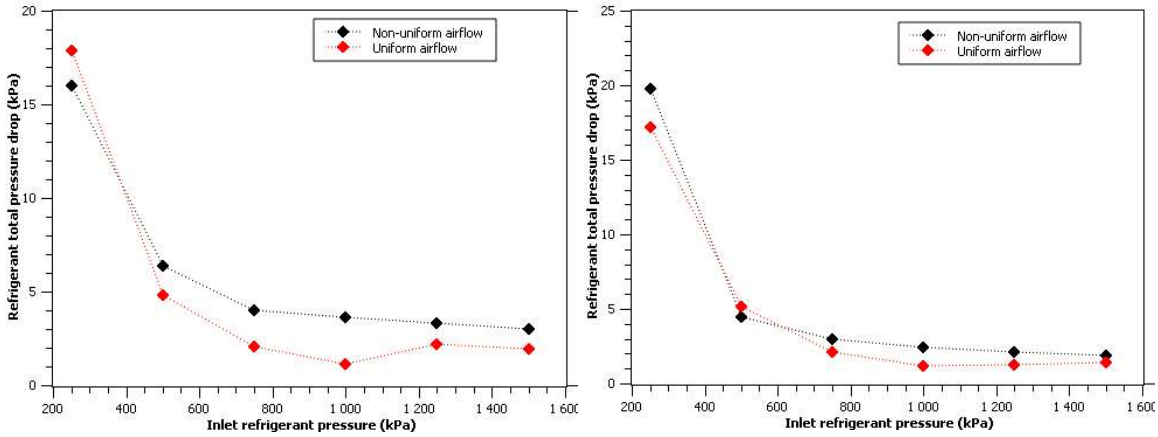


Figure 36: Inlet refrigerant pressure vs pressure drop H1 (left) and H2 (right)

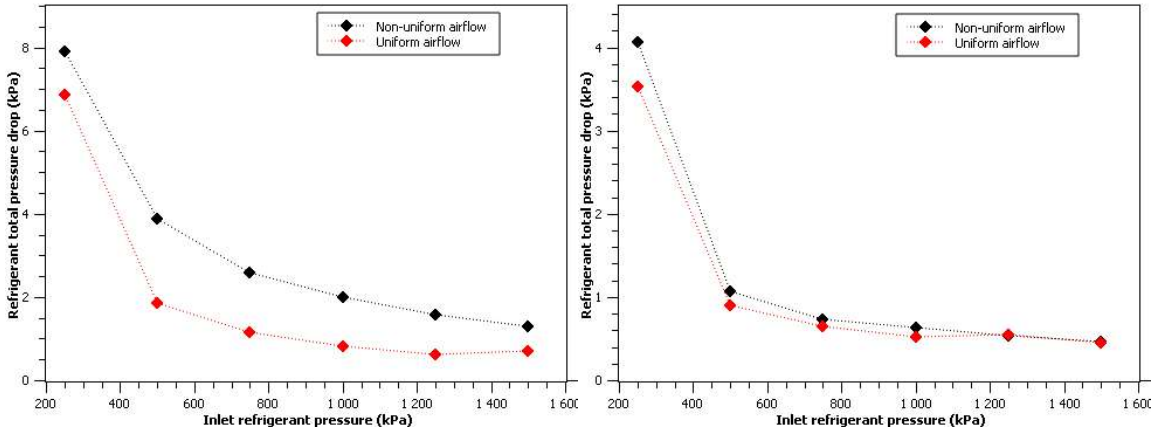


Figure 37: Inlet refrigerant pressure vs pressure drop H3 (left) and H4 (right)

In general, Figure 36 & 37 depict that the higher the inlet refrigerant pressure, the lower the refrigerant pressure losses in the tubes. This might be because increasing the inlet pressure of the refrigerant leads to an increase in the saturation temperature, that is, the temperature at which boiling occurs. An increase in saturation temperature means that the vapour state of the refrigerant approaches slower, that is, it requires a lot more heat to vaporise all the refrigerant liquid into vapour. As a result, refrigerant at higher inlet pressure takes more time to vaporise than at lower pressure and therefore experiences less pressure drop than refrigerant at lower pressure. This is because pressure drop is dependent on velocity and vapour experiences more pressure drop than a liquid. After all, the liquid is denser [66].

Moreover, Figure 36 & 37 suggest that assuming uniform airflow throughout the heat exchanger underpredicts the actual total pressure losses for all conditions in the inlet pressure range plotted in Figure 36 & 37. The variation in refrigerant pressure drop between uniform and non-uniform airflow reduces at higher inlet refrigerant pressure. That

is, the assumption of uniform airflow can estimate the actual refrigerant pressure drop at higher pressure drop with better accuracy than at lower refrigerant pressures.

5.5.2 Effects of variation in inlet refrigerant temperature

The inlet refrigerant temperature in a heat exchanger has a significant impact on the heat exchanger's performance, particularly on the heat transfer capacity and refrigerant drop. Figure 38 & 39 depict the impacts of varying inlet refrigerant temperature on the heat transfer rate of the heat exchanger while Figure 40 & 41 illustrate the impacts of varying inlet refrigerant temperature on the total refrigerant pressure losses. Different ranges of the inlet refrigerant temperature are considered for each operating condition of the heat exchanger and all other parameters were kept constant. For non-uniform airflow, the model considered four increments per tube.

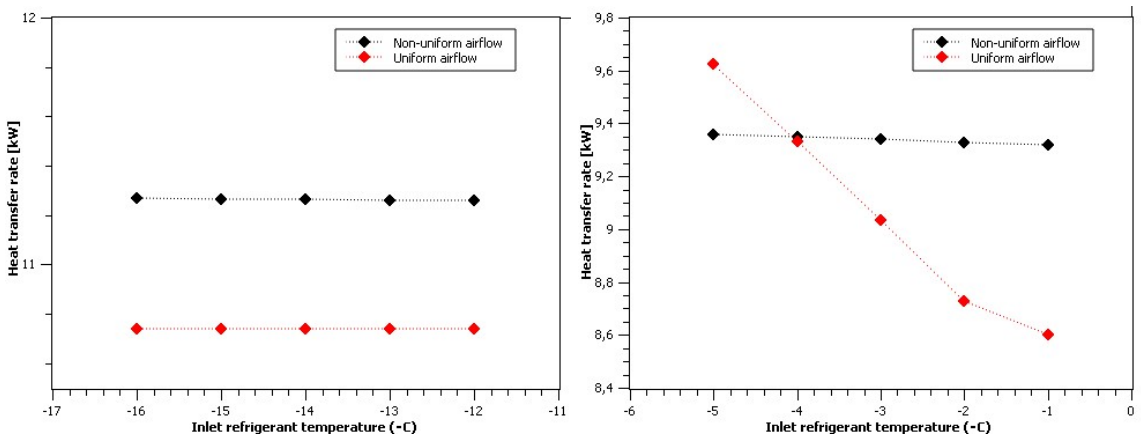


Figure 38:Inlet refrigerant temperature vs heat transfer capacity H1 (left) and H2 (right)

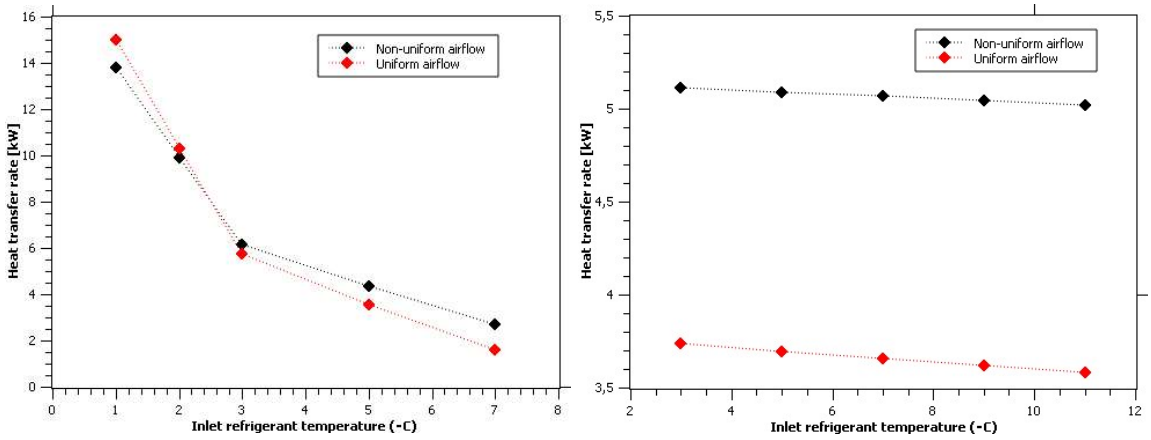


Figure 39: Inlet refrigerant temperature vs heat transfer capacity H3 (left) and H4 (right)

As previously discussed, the calculated heat transfer rate under the uniform airflow assumption is observed to be lower than the actual heat transfer rate calculated with the

inclusion of non-uniform airflow. This holds even under different inlet temperature ranges, as depicted in Figure 38 & 39.

Heat transfer is fundamentally dependent on the temperature difference between two fluid streams. In this context, as the temperature gradient between the incoming airflow and the refrigerant narrows, the amount of heat transferred between these two fluid streams decreases, as illustrated in Figure 38 & 39. This phenomenon is a direct consequence of the fact that the heat transfer rate decreases with the decrease in temperature gradient between two fluid streams.

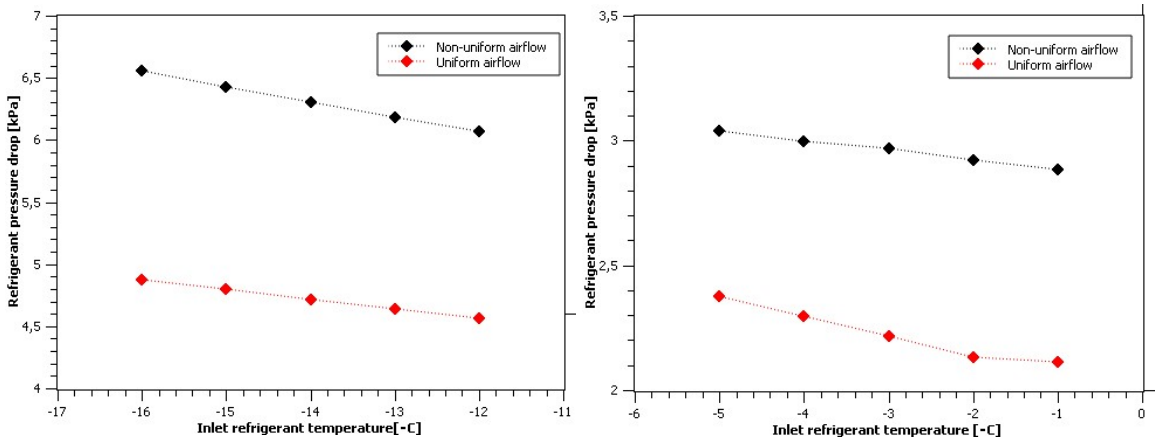


Figure 40:Inlet refrigerant temperature vs pressure drop H1 (left) and H2 (right)

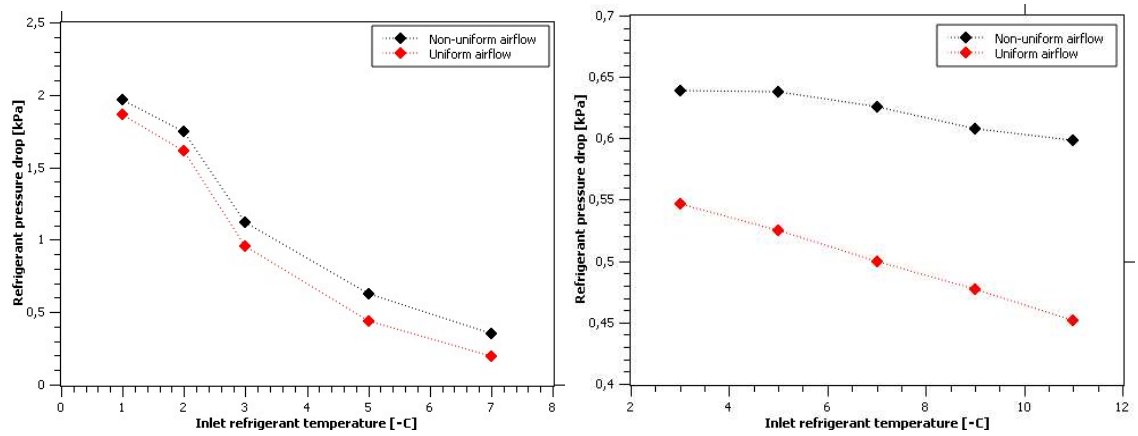


Figure 41: Inlet refrigerant temperature vs pressure drop H3 (left) and H4 (right)

Furthermore, this reduction in heat transfer rate as a result of decreasing temperature gradient between two fluid streams has a cascading effect on the refrigerant pressure drop. As the heat transfer rate decreases, the vapour quality of the refrigerant tends to decrease as well. In this case, as the heat transfer rate between airflow and refrigerant decreases, more of the refrigerant remains in its liquid phase. Liquids generally experience less pressure drop than vapour within the refrigerant tube, because liquid is denser than vapour.

Consequently, as a greater proportion of the refrigerant remains in its liquid state due to reduced heat transfer, the overall refrigerant pressure losses in the system decrease, as depicted in Figure 40 & 41.

5.5.3 Assessment of heat exchanger performance using different refrigerants

The thermal performance of the heat exchanger used in the heat pump system can also be influenced by the properties of the refrigerant being used as a working fluid. In this section, the comparisons in heat transfer rates and tube side pressure losses were made, using different refrigerants; R410a, R32 and R407C. These refrigerants are commonly used in air conditioning systems due to their low flammability and low toxicity as classified by the American Society of Heating, Refrigeration, and Air conditioning Engineers (ASHRAE) [67].

Figure 42 depicts heat transfer rates calculated per tube circuit from the developed heat exchanger model when using three different refrigerants. All other parameters were kept the same for three different refrigerants and the model discretised each tube into four increments. This assessment was done using operating condition H4.

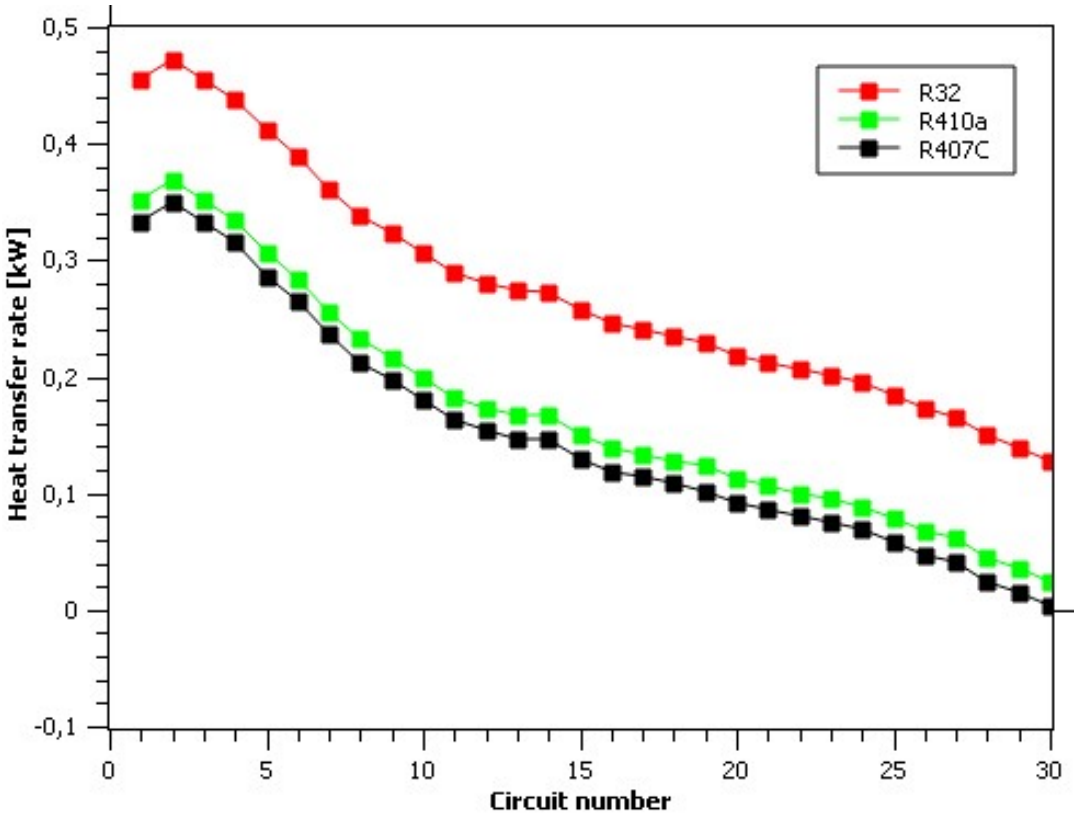


Figure 42: Heat transfer rates calculated per tube circuit for R32, R410a and R407C

The heat transfer rates were determined to be higher when utilising refrigerant R32, followed by R410a, and finally R407C. These findings align with the results presented by Dalila and Benelmir [68]. Specifically, the total heat transfer capacity of the heat exchanger was computed to be 38.46% greater when employing refrigerant R32 in comparison to R410a. Additionally, the heat transfer calculated from R410a exceeded that of R407C by 11.75%. This disparity can be attributed to the fact that refrigerant R32 exhibits a higher convective heat transfer coefficient than both R410a and R407C. Consequently, the higher heat transfer coefficient results in a more pronounced rate of heat transfer between the airflow and the refrigerant.

Furthermore, R32 has a higher latent heat of vaporisation when compared to both R410a and R407C, with R410a ranking second and R407C third in this regard. This signifies that R32 demands a greater amount of heat energy to transition from a liquid to a vapour phase. As a result, it is probable that R32 will exhibit a higher heat transfer rate due to its enhanced heat absorption characteristics [69].

Figure 43 also shows the tube side pressure losses per tube circuit for the same refrigerants, under operating condition H4.

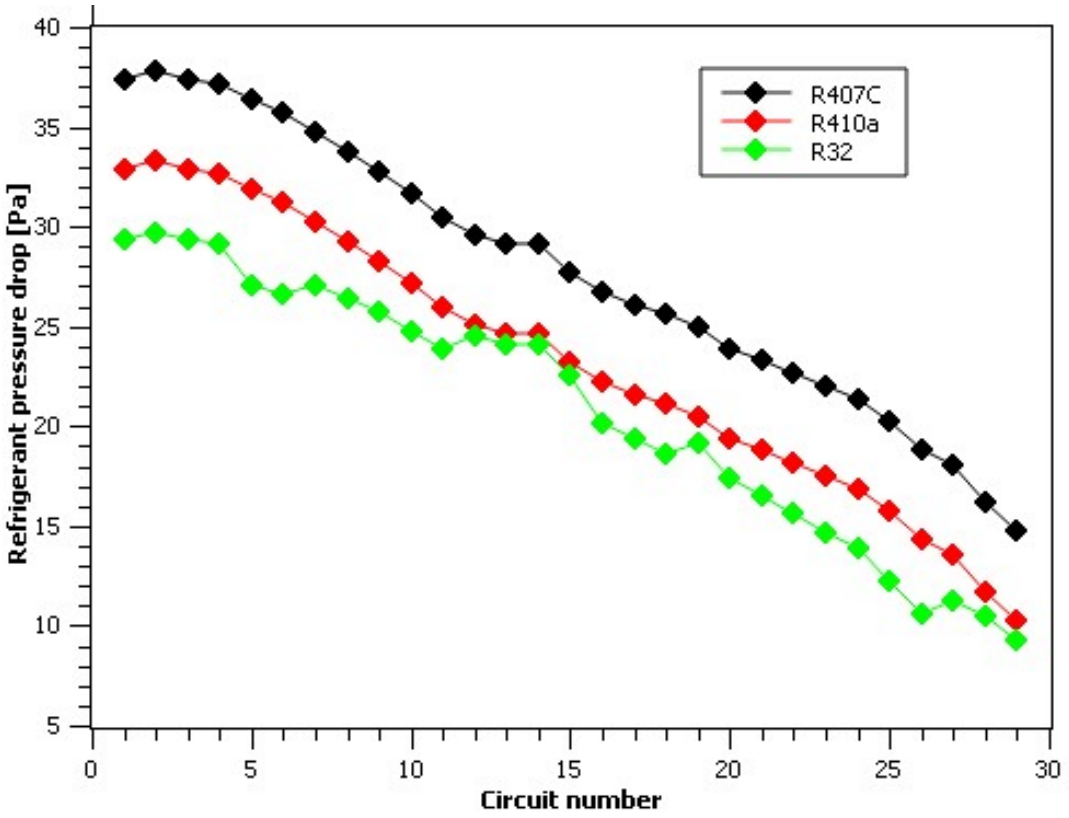


Figure 43:Refrigerant pressure losses calculated per tube circuit for R32, R410a and R407C

The graphical representation in Figure 43 suggests that the highest-pressure losses are associated with the utilisation of refrigerant R407C, followed by R410a, while the least pressure losses are observed when R32 is employed. Specifically, the total refrigerant pressure loss calculated when using R407C exceeds that of R410a by 25.34%. In contrast, the total refrigerant pressure loss computed for R410a is 11.87% higher than that for R32. This discrepancy arises from the fact that higher latent heat of vaporisation necessitates a greater amount of heat energy to transition all the liquid into vapour. Consequently, the liquid phase of R32 requires a lengthier duration and more heat to vaporise compared to R410a and R407C. Therefore, a higher proportion of liquid relative to vapour leads to reduced pressure losses, as liquids inherently experience lower pressures than vapour due to their higher density, as previously discussed.

Additionally, Figure 44 displays calculated COP values for different refrigerants under operating condition H4. The chart shows that the heat pump system achieves the highest COP when using R32 as a refrigerant, followed by R410a, and the lowest COP when R407C is employed. Thus, the calculated COP values are 11.22, 6.902 and 6.091 when utilising refrigerant R32, R410a and R407C, respectively.

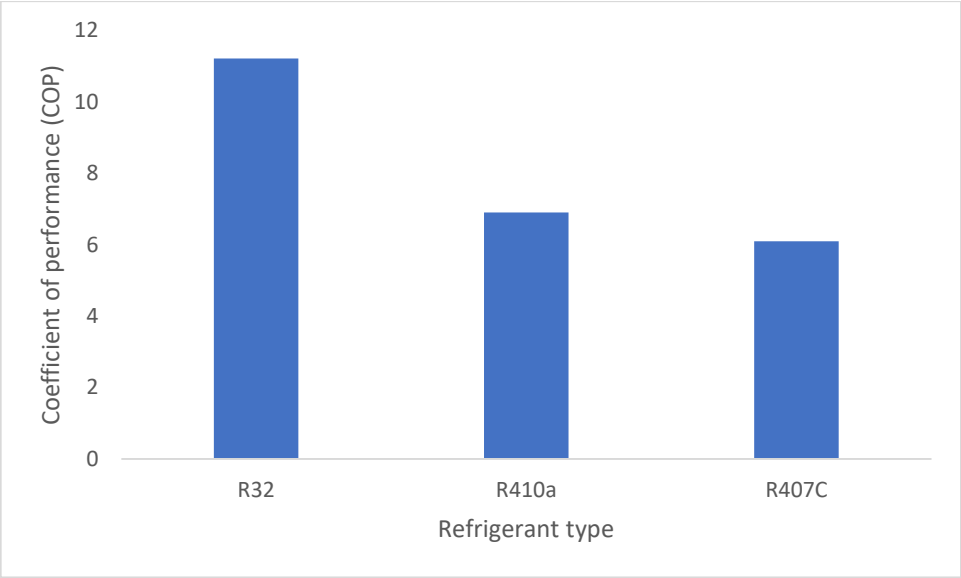


Figure 44: Calculated COP values for different refrigerants

The highest COP value calculated for refrigerant R32 indicates that, for a specified compressor power input, the evaporator extracts more heat compared to other refrigerants under investigation. This could be attributed to the fact that R32 takes a longer time and more heat to vaporise in comparison to R410a and R407C, as discussed in the previous section. Consequently, this extended vaporisation process enables R32 to absorb more heat before transitioning into the superheated phase within the heat exchanger, distinguishing it from other refrigerants in this regard.

5.6 Summary

This chapter has presented a comprehensive assessment of the non-uniform airflow patterns within a fin and tube heat exchanger under investigation. Furthermore, it has delved into the consequential effects of this airflow non-uniformity on various aspects, including heat transfer rates, entropy variations, refrigerant pressure losses, vapour quality, and their cumulative impact on the COP of the integrated heat pump system. Additionally, within this chapter, a thorough evaluation of the developed heat exchanger model has been conducted under conditions that deviate from the standard operating parameters. These variations encompass alterations in the inlet refrigerant temperature and pressure, as well as the utilisation of different refrigerants. Considering these findings, it can be ascertained that the objectives outlined in Chapter 1 have been successfully attained.

6. Conclusions and recommendations

The aim of this study was to model and analyse a fin and tube heat exchanger using Ansys Fluent and Python and integrate it into a modelled heat pump system in Python. These analyses were conducted to identify how the thermal performance of the vapour compression heat pump systems can be improved for better efficiency during the design stage. These models were developed and validated according to the methodology described in the preceding chapters. This chapter, therefore, draws conclusions on the findings from the study and highlights how the work done in this study can be improved for future related studies.

6.1 Conclusions

The following conclusions may be drawn from the analyses made on the modelled fin and tube heat exchanger and heat pump system:

- The airflow velocity in a fin and tube heat exchanger where a fan is installed on top varies significantly in 3D. That is, there is a noticeable change in airflow velocity magnitude along the heat exchanger height, tube length, and across the tube rows. Also, the change of air velocity is higher along the heat exchanger height than across tube rows or along tube length.
- Assuming uniform airflow on the heat exchanger underestimates the actual heat transfer rates and refrigerant pressure losses on the upper part of the heat exchanger while overpredicting heat transfer as well as refrigerant pressure losses on the lower tube circuits. The maximum variations between actual refrigerant pressure losses and heat transfer calculated when discretizing heat exchanger tubes (in 3D) compared with the ones calculated under uniform airflow velocity are 30.14% and 27.85 %, respectively for four operating conditions of the heat pump mentioned in the previous chapters.
- Assuming uniform airflow on the heat exchanger also underpredicts refrigerant vapour quality on the upper tube circuits while overpredicting the vapour quality on the lower circuits.
- Performing calculations on heat transfer rates, refrigerant pressure losses and vapour quality by discretizing each tube in a heat exchanger into several increments improves the certainty of the calculations because the properties of both air and refrigerant are seen to vary significantly with small tube increments.

- Under all four operating conditions of the heat pump, analysing the heat exchanger performance by discretizing each tube of the heat exchanger into small increments results in significantly different COP from that of assuming uniform airflow. That is, assuming uniform airflow velocity results in COP maximum variation of 11.07% when comparing with the results whereby nonuniform airflow was considered.

Therefore, based on the findings from the study, the thermal performance of a residential vapour compression heat pump system analysed in this study can be improved by performing all the calculations by discretizing each tube circuit into small increments, that is, at least four increments per tube. This means each increment is taken as a control volume. This will therefore ensure that the chosen compressor input power is calculated with better accuracy.

Moreover, reducing the height of the heat exchanger by adding more tube rows and fewer tube circuits can enhance the heat exchanger performance and hence increase heat pump COP. This is because airflow variations are more along the heat exchanger height than across the tube rows. This will therefore increase the heat transfer capacity and hence reduce the dissipation of electrical energy by the heat pump compressor. Therefore, with the presence of these unique concepts that the study introduces, it is expected that the currently developed model will enhance productivity in the conceptualization and enhancement of finned tube heat exchangers.

Therefore, the analyses conducted on the modelled fin and tube heat exchanger and heat pump system provide valuable insights into the intricacies of airflow velocity distribution, heat transfer rates, refrigerant pressure losses, and vapor quality within the system. The findings highlight the significant variations in airflow velocity magnitude along the height of the heat exchanger, tube length, and across tube rows, challenging the assumption of uniform airflow. By discretizing each tube into small increments, the study demonstrates improved accuracy in calculations of heat transfer rates, refrigerant pressure losses, and vapor quality, leading to more precise predictions of the heat pump's COP.

Overall, the study's findings provide valuable contributions to the body of knowledge in thermal engineering, offering insights that can inform the design of heat exchangers for enhanced energy efficiency and performance.

6.2 Recommendations

The recommendations on the current study for future related work are:

- The heat exchanger model in Ansys Fluent is just a representation of the actual fin and tube heat exchanger. That is, the model is simplified to a one-dimensional thin membrane known as a porous jump, using known velocity vs airflow pressure losses on the heat exchanger airside. Even though this simplification produces reliable outcomes without compromising accuracy, the next step may be to model the actual heat exchanger in Ansys Fluent to investigate velocity profiles on each tube surface and how the airflow variation around each tube surface affects the heat transfer capacity of the heat exchanger.
- In this study, all refrigerant inlets at every tube circuit are assumed to have the same properties. However, there might be heat or pressure loss on the main pipe refrigerant which travels a longer distance in the main tube to far away circuit inlets from the main refrigerant inlet. Therefore, future work may explore this as it was not in the scope of the current study.
- Future work may consider the refrigerant pressure losses due to tube bends and how they affect the heat exchanger's thermal performance.

References

[1] M. Schwikowski, "Coal, Africa' s conundrum as energy demands increases," *ESI Africa,* 2021.

[2] E. Atasoy, "Experimental-based optimization of an energy-efficient heat pump integrated water heater for household appliances.," *Energy,* no. 245, 2022.

[3] I. Mohammed, "Operational vs embodied emissions in buildings - A review of current trends," *Energy and Buildings,* pp. 232-245, 2013.

[4] L. Perez-Lombard, "A review on buildings energy consumption information," *Energy and Buildings,* no. 40, pp. 394-398, 2008.

[5] "Department of Economic and Social Affairs." United Nations. https://sdgs.un.org/2030agenda (accessed 15 October, 2022).

[6] C. Jingyong, "Performance optimization of solar-air composite source multi-functional," *Sustainable Energy Technologies and Assessments,* vol. 50, 2022.

[7] N. Andrey, M. Deymi-Dashtebayaz, S. Muraveinikov, and V. Nikitina, "Comparative study of air source and ground source heat pumps in 10 coldest Russian cities based on energy-exergy-economic-environmental analysis," *Journal of Cleaner Production,* vol. 321, 2021.

[8] M. Yu, S. Li, and X. Zhang, "Techno-economic analysis of air source heat pump combined with latent thermal energy storage applied for space heating in China," *Applied Thermal Engineering,* vol. 185, 2021.

[9] R. Brecha. "Electric Heat Pumps: Existing Tech for Energy-efficient Future." How stuff works. https://home.howstuffworks.com/home-improvement/heating-and-cooling/electric-heat-pumps-existing-tech-news.htm (accessed 04 September, 2023).

[10] B. Lei, Z. Duan, and Y. T. Wu, "Thermodynamic investigations on an integrated heat pump with thermal," *Energy Conversion and Management,* vol. 254, 2022.

[11] F. Leonforte and A. Miglioli, "Design and performance monitoring of a novel photovoltaic-thermal solar-assisted heat pump system for residential applications," *Applied Thermal Engineering,* vol. 210, 2022.

[12] K. Sezen and A. D. Tuncer, "Effects of ambient conditions on solar assisted heat pump systems:a review," *Science of the Total Environment,* vol. 778, 2021.

[13] M. Mohanraj, Y. Belyayev, S. Jayaraj, and A. Kaltayev, "Research and developments on solar assisted compression heat pump systems – A comprehensive review (Part

A: Modeling and modifications)," *Renewable and Sustainable Energy Reviews,* vol. 83, pp. 90-123, 2018.

[14] S. Deng, Y. Dai, and R. Wang, "Performance optimization and analysis of solar combi-system with carbon dioxide heat pump " *Solar Energy,* vol. 98, pp. 212-225, 2013.

[15] K. Moltem and A. Mustafaa, "Experimental investigation of a novel thermal energy storage unit in the heat pump system," *Journal of Cleaner Production,* vol. 311, 2021.

[16] E. Sahin and N. Adiguzel, "Experimental analysis of the effects of climate conditions on heat pump system performance," *Energy,* vol. 243, 2022.

[17] Z. A. Qureshi, H. M. Ali, and S. Khushnood, "Recent advances on thermal conductivity enhancement of phase change materials for energy storage system: A review," *International Journal of Heat and Mass Transfer,* vol. 127, pp. 838-856, 2018.

[18] Q. Ye and S. Li, "Investigation on the performance and optimization of heat pump water heater with wrap-around condenser coil " *International Journal of Heat and Mass Transfer,* vol. 143, 2019.

[19] Z. Marshall and J. Duquette, "A techno-economic evaluation of low global warming potential heat pump assisted organic Rankine cycle systems for data center waste heat recovery," *Energy,* vol. 242, 2022.

[20] A. Aghagoli and M. Sorin, "CFD modelling and exergy analysis of a heat pump cycle with Tesla turbine using CO2 as a working fluid," *Applied Thermal Energy,* vol. 178, 2020.

[21] E. Atasoy, B. Çetin, and O. Bayer, "Experiment-based optimization of an energy-efficient heat pump integrated water heater for household appliances," *Energy,* vol. 245, 2022.

[22] Z. H. Ayub, "Plate heat exchanger literature survey and new heat transfer and pressure drop correlations for refrigerant evaporators," *Texas Heat Transfer Engineering,* vol. 24, no. 5, pp. 3-16, 2003.

[23] D. Kim, I. Moretti, and H. Huber, "Heat exchangers and the perfomance of heat pumps - Analysis of a heat pump database," *Applied Thermal engineering,* vol. 31, pp. 911-920, 2011.

[24] "HVAC Systems and Equipment," in *ASHRAE Handbook*, 2008.

[25] S. Kiyoshi, Y. Shuichi, M. Shigeru, Y. Michio, and M. Hiroshi, "Low-pressure type scroll compressor for air conditioners," *National Technical Report,* vol. 35, no. 6, pp. 80-86, 1989.

[26] D. Reay, "Compact heat exchangers, enhancement and heat pumps," *International Journal of Refrigeration,* vol. 25, pp. 460-470, 2002.

[27] V. Malapure, S. K. Mitra, and A. Bhattacharya, "Numerical investigation of fluid flow and heat transfer over louvered fins in compact heat exchanger," *International Journal of Thermal Sciences,* vol. 46, no. 2, pp. 199-211, 2007.

[28] A. Sadeghianjahromi and C. C. Wang, "Heat transfer enhancement in fin-and-tube heat exchangers – A review on different mechanisms," *Renewable and Sustainable Energy Reviews,* vol. 137, 2021.

[29] M.-H. Kim and C. W. Bullard, "Air-side thermal hydraulic performance of multi-louvered fin aluminum heat exchangers," *International Journal of Refrigeration,* vol. 25, pp. 390-400, 2002.

[30] V. Malapure, S. K. Mitra, and A. Bhattacharya, "Numerical investigation of fluid flow and heat transfer over louvered fins in compact heat exchanger," *International Journal of Thermal Sciences,* vol. 46, pp. 199-211, 2007.

[31] R. K. Pasupuleti, M. Bedhapudi, S. R. Jonnala, and A. R. Kandimalla, "Computational Analysis of Conventional and Helical Finned Shell and Tube Heat Exchanger Using ANSYS-CFD," *International Journal of Heat and Technology,* vol. 39, no. 6, pp. 1755-1762, 2021.

[32] A. A. Bhuiyan, M. R. Amin, and A. S. Islam, "Three-dimensional performance analysis of plain fin tube heat exchangers in transitional regime," *Applied Thermal Engineering,* vol. 50, no. 1, pp. 445-454, 2013.

[33] S. Ishaque and M.-H. Kim, "Numerical modeling of an outdoor unit heat exchanger for residential heat pump systems with nonuniform airflow and refrigerant distribution," *International Journal of Heat and Mass Transfer,* vol. 175, 2021.

[34] D. Taler, J. Taler, and K. Wrona, "Transient response of a plate-fin-and-tube heat exchanger considering different heat transfer coefficients in individual tube rows," *Energy,* vol. 195, 2020.

[35] J. P. Chiou, "Thermal Performance Deterioration in Crossflow Heat Exchanger due to the Flow Nonuniformity," *Heat Transfer,* vol. 100, no. 4, pp. 580-587, 1978.

[36] A. Okbaz, A. Pınarbaşı, and A. B. Olcay, "Experimental investigation of effect of different tube row-numbers, fin pitches and operating conditions on thermal and

hydraulic performances of louvered and wavy finned heat exchangers," *International Journal of Thermal Sciences,* vol. 151, 2020.

[37] A. Saleem and M. H. Kim, "CFD Analysis on the Air-Side Thermal-Hydraulic Performance of Multi-Louvered Fin Heat Exchangers at Low Reynolds Numbers " *Energies,* vol. 10, 2017.

[38] X. Y. Li, Z. H. Li, and W. Q. Tao, "Experimental study on heat transfer and pressure drop characteristics of fin-and-tube surface with four convex-strips around each tube," *International Journal of Heat and Mass Transfer,* vol. 116, pp. 1085-1095, 2018.

[39] H. Huisseune, C. T'Joen, P. D. Jaeger, B. Ameel, S. D. Schampheleire, and M. D. Paepe, "Performance enhancement of a louvered fin heat exchanger by using delta winglet vortex generators," *International Journal of Heat and Mass Transfer,* vol. 56, no. 1-2, pp. 475-487, 2013.

[40] K. Zhang, M. J. Li, H. Liu, J. G. Xiong, and Y. L. He, "Experimental and numerical study and comparison of performance for herringbone wavy fin and enhanced fin with convex-strips in fin-and-tube heat exchanger," *International Journal of Heat and Mass Transfer,* vol. 175, 2021.

[41] R. Matos, T. Laursen, T. Laursen, and A. Bejan, "Three-dimensional optimization of staggered finned circular and elliptic tubes in forced convection " *International Journal of Thermal Sciences,* vol. 43, no. 5, pp. 477-487, 2004.

[42] Y. Tao, Y. He, Z. Wu, and W. Tao, "Three-dimensional numerical study and field synergy principle analysis of wavy fin heat exchangers with elliptic tubes," *International Journal of Heat and Fluid Flow,* vol. 28, no. 6, pp. 1531-1544, 2007.

[43] W. L. Hu, A. J. Ma, Y. Guan, Z. J. Cui, Y. B. Zhang, and J. Wang, "Experimental Study of the Air Side Performance of Fin-and-Tube Heat Exchanger with Different Fin Material in Dehumidifying Conditions," *Energies,* vol. 14, no. 21, 2021.

[44] L. Tang, M. Zeng, and Q. Wang, "Experimental and numerical investigation on air-side performance of fin-and-tube heat exchangers with various fin patterns," *Experimental Thermal and Fluid Science,* vol. 33, no. 5, pp. 818-827, 2009.

[45] P. Pongsoi, S. Pikulkajorn, C.-C. Wang, and S. Wongwises, "Effect of number of tube rows on the air-side performance of crimped spiral fin-and-tube heat exchanger with a multipass parallel and counter cross-flow configuration," *International Journal of Heat and Mass Transfer,* vol. 55, no. 4, pp. 1403-1411, 2012.

[46] L. Tang, M. Zeng, and Q. Wang, "Experimental and numerical investigation on air-side performance of fin-and-tube heat exchangers with various fin patterns," *Experimental Thermal and Fluid Science,* vol. 33, no. 5, pp. 818-827, 2009.

[47] "Heat Exchangers in the HVAC/R Sector." CAREL INDUSTRIES S.p.A. https://www.carel.com/blog/-/blogs/heat-exchangers-in-the-hvac-r-sector (accessed 29 August, 2023).

[48] J. E. O'Brien and M. S. Sohal, "Heat transfer enhancement for finned-tube heat exchnagers with winglets," *International Journal of Heat Transfer,* 2000.

[49] S. Ishaque, "Numerical modeling of an outdoor unit heat exchanger for residential," *International Journal of Heat and Mass Transfer,* 2021.

[50] R. Brown, "Experimental investigation of geometry effects on the performance of a compact louvered heat exchanger.," *ASME Journal of Heat Transfer,* vol. 31, no. 16, pp. 3337-3346, 2011.

[51] L. Chen, "Staggered Tube Arrangement Effects on Heat Exchanger Performance.," *Heat and Mass Transfer Journal,* vol. 28, no. 4, 2020.

[52] "FLUENT user's guide release 16.0," in *ANSYS FLUENT*, 2015.

[53] S.Ishaque and M. H. Kim, "Seasonal performance investigation for residential heat pump system with different outdoor heat exchanger designs.," *Energies,* no. 12, 2019.

[54] C. S. An and M. H. Kim, "Thermo-hydraulic analysis of multi-row cross-flow heat exchangers," *International Journal of Heat and Mass Transfer,* no. 120, pp. 534-539, 2018.

[55] F.Incropera, "Heat exchangers," in *Fundamentals of mass and heat transfer*. United States of America: John Wiley & Sons, 2007, pp. 667-707.

[56] D.Admiraal, "Heat transfer Correlations," *Heat Transfer in Refrigerator Condensers and Evaporators,* pp. 3-6, 1993.

[57] P.Rousseau, "Air conditioning system info pack," University of Cape Town, 2020.

[58] D. K. Swamee and A. K. Jain, "Explicit Equations for Pipe Flow Problems," *Journal of the Hydraulics Division,* no. 102, pp. 657-664, 1976.

[59] I. Bell, J. Wronski, S. Quoilin, and V. Lemort, "Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp," *Ind. Eng. Chem. Res.,* vol. 53, no. 6, pp. 2498–2508, 2014.

[60] P. Rousseau, "FluidProperties.py," University of Cape Town, Cape Town, 2019.

[61] W. Francesco and T. Ilja, "TESPy: Thermal Engineering Systems in Python," *Journal of Open Source Software,* vol. 5, no. 49, pp. 1-2178, 2020.

[62] W. Francesco and T. Ilja, "TESPy 0.6.3 dev Documentation," ed: The Journal of Open Source Software, 2020.

[63] I. Shehryar, M. I. H. Siddiqui, and K. Man-Hoe, "Effect of heat exchanger design on seasonal performance of heat pump systems," *International Journal of Heat and Mass Transfer,* vol. 151, 2020.

[64] M. Moran, N. Howard, and Shapiro, "Refrigeration and Heat pump Systems," in *Fundamentals of Engineering Thermodynamics*. West Sussex: John Wiley & Sons, Inc., 2006, pp. 454-479.

[65] Y. A. Cengel and M. A. Boles, *Thermodynamics: An Engineering Approach*. McGraw-Hill, 2006.

[66] "Refrigeration handbook." SWEP. https://www.swep.net/refrigerant-handbook/refrigerant-handbook/ (accessed 12 September, 2023).

[67] A. Kapicioglu, "Theoretical examination of altenative refrigerants for R410a in a ground source heat pump according to ASHRAE classification.," *ADYU Mühendislik Bilimleri Dergisi,* no. 16, pp. 129-147, 2022.

[68] A. Dalila and R. Benelmir, "Comparative assessment of heat pump using refrigerants R407C, R410A and R32 as working fluids with regards to exergy and exergoeconomic performance.," *International Journal of Energy, Environment and Economics,* vol. 27, no. 2, 2021.

[69] "Honeywell Continues Transition To Next-Generation Refrigerant In Collaboration With Trane Technologies." https://www.honeywell.com/us/en/press/2021/06/honeywell-continues-transition-to-next-generation-refrigerant-in-collaboration-with-trane-technologies (accessed 26 September, 2023).

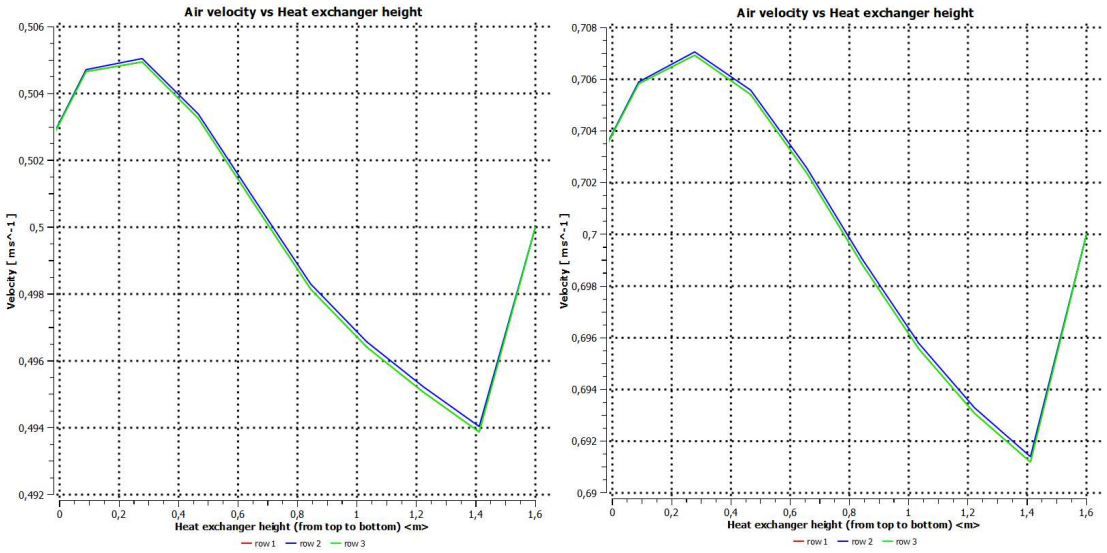
Appendices

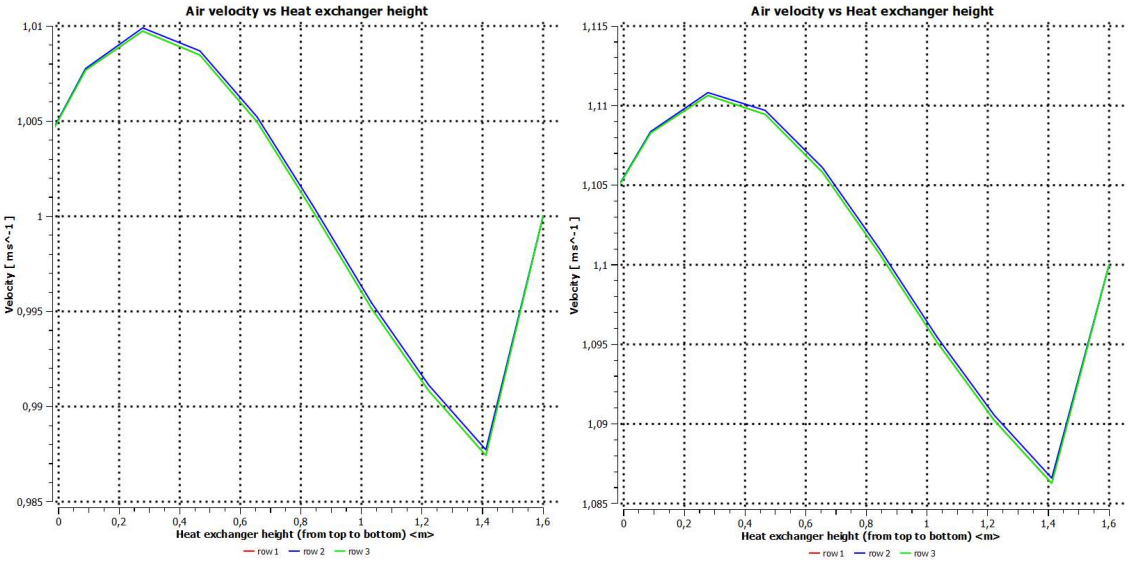
Appendix A: Velocity profiles

This section shows velocity graphs along the heat exchanger height for each tube row. These velocity values were calculated from the 1D simplified heat exchanger representation in Ansys Fluent. The graphs depict velocities on the mid-plane which divides tubes into two equal lengths, that is, at x = 860 mm. The following Forchheimer formula was used to solve for permeability (K) and inertia resistance factor (C), using second-order polynomial curve fitting of the known face velocity vs pressure drop curve.

$$\frac{\Delta p}{l} = -(\frac{\mu}{K} + \frac{1}{2}\rho C u^2)$$

The graphs show velocity vs heat exchanger height for operating conditions 4, 3, 2 and 1, respectively.




Air velocity vs Heat exchanger height
Velocity [ms^-1]
1,01
1,005
1
0,995
0,99
0,985
0
0,2
0,4
0,6
0,8
1
1,2
1,4
1,6
Heat exchanger height (from top to bottom) <m>
row 1
row 2
row 3
Air velocity vs Heat exchanger height
Velocity [ms^-1]
1,115
1,11
1,105
1,1
1,095
1,09
1,085
0
0,2
0,4
0,6
0,8
1
1,2
1,4
1,6
Heat exchanger height (from top to bottom) <m>
row 1
row 2
row 3

Appendix B: Heat exchanger

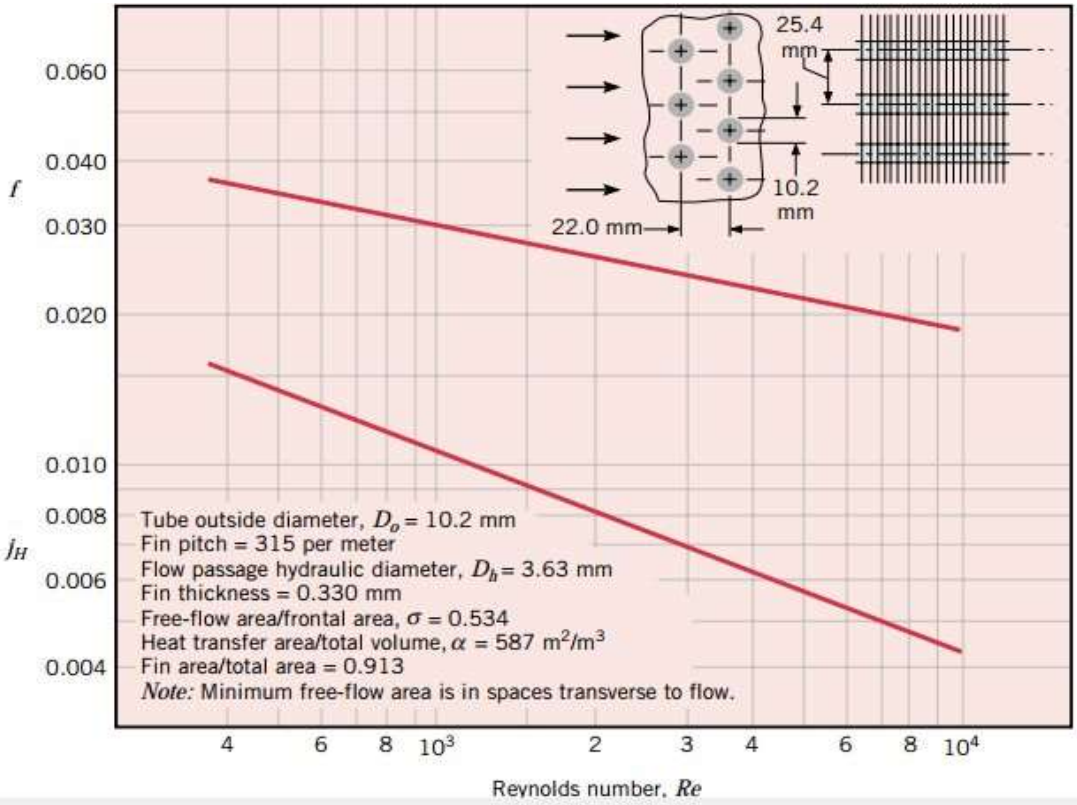


Figure 45: shows a continuous fin heat exchanger characteristic[55]

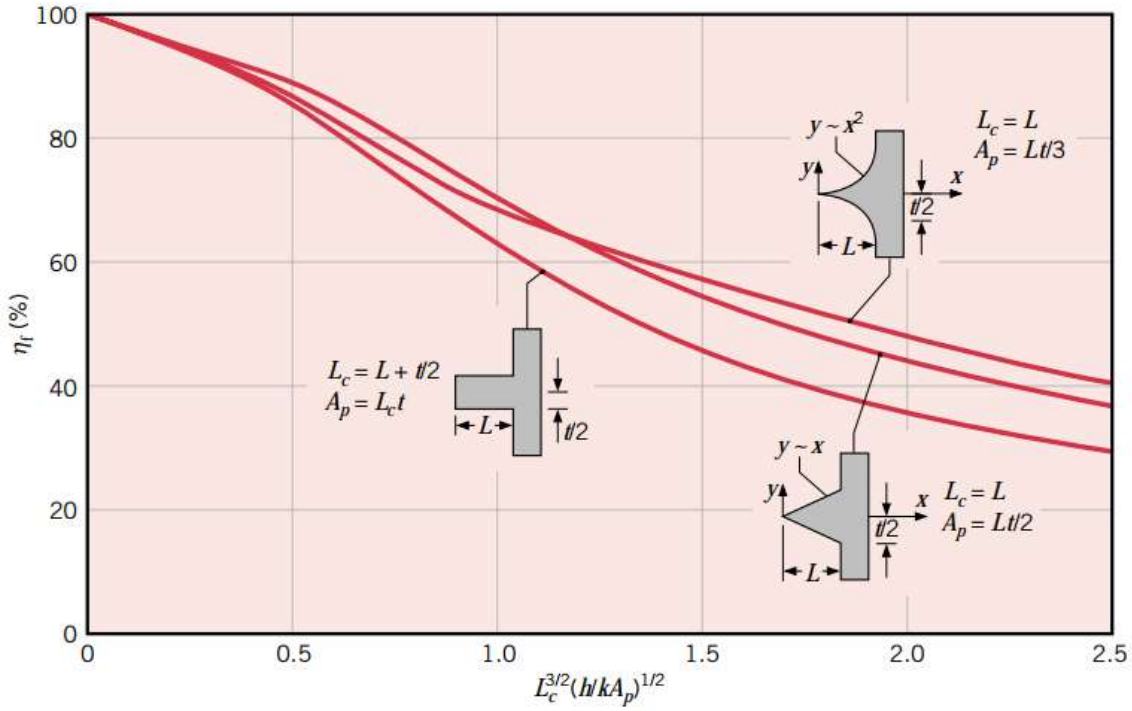


Figure 46: shows the fin efficiency vs fins' geometry[55]

Heat exchanger python model

This section shows the Python model that represents a multi-row fin and tube heat exchanger analysed in this study. The code was used for all operating conditions of the heat exchanger and the one shown in this section has inlet refrigerant and airflow for operating condition 3.

Discretizing each tube into four increments

```
"""
Multi-row fin and tube heat exchanger model- 4 increments.ipynb
Written by
Sehobai Sehobai
August 2023
"""
"""
This code is built with the use of an open source library CoolProp.
CoolProp is free to all users.
The developers simply ask that if you use CoolProp in your academic work,
that you make a reference to CoolProp by siting the following publication:
I. Bell, J. Wronski, S. Quoilin, and V. Lemort,
“Pure and pseudo-pure fluid thermophysical property evaluation
and the open-source thermophysical property library CoolProp,”
Ind. Eng. Chem. Res., vol. 53, no. 6, pp. 2498–2508, 2014.
and the domain:
www.coolprop.org.
"""
"""
The module FluidProperties that has been used herein this code has been built by:
Prof Pieter Rousseau
Nov 2019
"""

import FluidProperties as fp
import numpy as np
import math
import matplotlib.pyplot as plt
from numpy import inf

p = 101325 #atmospheric pressure

# 1. Air velocity in 3D from Ansys Fluent (CFD modelling)

N_circ = 0.0 #initialising circuit number
Q_dot_s_hx = 0.0 #initialising total heat transfer on the air side
Q_dot_p_hx = 0.0 #initialising total heat transfer on the refrigeration side
Q_dot_ref = [] #initialising total heat transfer per circuit for plotting
N_Circuits = [] #initialising circuits numbers for plotting
deltaPress_r_tot = 0.0 #initialising total pressure losses on refrigerant

while N_circ < 30: #assigning velocity values for tube increment per circuit
    N_circ = 1 + N_circ
    if N_circ == 1:
        V_y =
np.array([[[0.6957,0.6908,0.6907,0.6955],[0.6964,0.6910,0.6910,0.6962]],[[0.8053,0.8001,0.8001,0.80
51],[0.8057,0.8002,0.8002,0.8055]],[[0.6957,0.6908,0.6908,0.6955],[0.6964,0.6910,0.6910,0.6962]]])
    elif N_circ == 2:
        V_y =
np.array([[[0.6968,0.6912,0.6912,0.6966],[0.6971,0.6915,0.6915,0.6969]],[[0.8059,0.8003,0.8003,0.80
57],[0.8058,0.8003,0.8003,0.8057]],[[0.6968,0.6912,0.6912,0.6966],[0.6971,0.6915,0.6915,0.6969]]])
    elif N_circ == 3:
        V_y =
np.array([[[0.6972,0.6917,0.6917,0.6971],[0.6973,0.6920,0.6919,0.6972]],[[0.8057,0.8003,0.8003,0.80
55],[0.8054,0.8002,0.8002,0.8052]],[[0.6972,0.6917,0.6917,0.6971],[0.6973,0.6920,0.6920,0.6972]]])
    elif N_circ == 4:
        V_y =
np.array([[[0.6974,0.6922,0.6922,0.6972],[0.6974,0.6925,0.6925,0.6973]],[[0.8050,0.8001,0.8001,0.80
50],[0.8045,0.7999,0.7999,0.8044]],[[0.6974,0.6922,0.6922,0.6972],[0.6974,0.6925,0.6925,0.6973]]])
    elif N_circ == 5:
```

```
        V_y =
np.array([[[0.6974,0.6927,0.6927,0.6973],[0.6973,0.6930,0.6930,0.6974]],[[0.8040,0.7996,0.7996,0.80
39],[0.8034,0.7993,0.7993,0.8033]],[[0.6974,0.6927,0.6927,0.6973],[0.6974,0.6930,0.6930,0.6973]]])
    elif N_circ == 6:
        V_y =
np.array([[[0.6974,0.6933,0.6933,0.6973],[0.6974,0.6936,0.6935,0.6973]],[[0.8027,0.7989,0.7989,0.80
26],[0.8020,0.7985,0.7984,0.8019]],[[0.6973,0.6933,0.6933,0.6973],[0.6974,0.6936,0.6935,0.6973]]])
    elif N_circ == 7:
        V_y =
np.array([[[0.6974,0.6939,0.6938,0.6973],[0.6975,0.6942,0.6941,0.6974]],[[0.8012,0.7979,0.7979,0.80
11],[0.8004,0.7973,0.7973,0.8003]],[[0.6974,0.6939,0.6938,0.6973],[0.6975,0.6942,0.6941,0.6974]]])
    elif N_circ == 8:
        V_y =
np.array([[[0.6975,0.6945,0.6944,0.6975],[0.6976,0.6948,0.6948,0.6976]],[[0.7995,0.7967,0.7967,0.79
94],[0.7986,0.7960,0.7960,0.7985]],[[0.6975,0.6945,0.6944,0.6975],[0.6976,0.6948,0.6948,0.6976]]])
    elif N_circ == 9:
        V_y =
np.array([[[0.6978,0.6951,0.6951,0.6977],[0.6979,0.6955,0.6955,0.6979]],[[0.7976,0.7952,0.7952,0.79
76],[0.7966,0.7944,0.7943,0.7965]],[[0.6978,0.6951,0.6951,0.6977],[0.6979,0.6955,0.6955,0.6979]]])
    elif N_circ == 10:
        V_y =
np.array([[[0.6981,0.6958,0.6958,0.6981],[0.6983,0.6962,0.6962,0.6983]],[[0.7955,0.7934,0.7934,0.79
55],[0.7944,0.7925,0.7925,0.7943]],[[0.6981,0.6958,0.6958,0.6981],[0.6983,0.6962,0.6962,0.6983]]])
    elif N_circ == 11:
        V_y =
np.array([[[0.6986,0.6966,0.6966,0.6985],[0.6988,0.6970,0.6970,0.6988]],[[0.7932,0.7914,0.7914,0.79
32],[0.7920,0.7904,0.7904,0.7920]],[[0.6986,0.6966,0.6966,0.6986],[0.6988,0.6970,0.6970,0.6988]]])
    elif N_circ == 12:
        V_y =
np.array([[[0.6991,0.6974,0.6974,0.6991],[0.6994,0.6978,0.6978,0.6994]],[[0.7907,0.7892,0.7892,0.79
07],[0.7894,0.7880,0.7880,0.7894]],[[0.6991,0.6974,0.6974,0.6991],[0.6994,0.6978,0.6978,0.6994]]])
    elif N_circ == 13:
        V_y =
np.array([[[0.6997,0.6983,0.6982,0.6997],[0.7001,0.6987,0.6987,0.7001]],[[0.7880,0.7867,0.7867,0.78
80],[0.7866,0.7854,0.7854,0.7866]],[[0.6997,0.6983,0.6982,0.6997],[0.7001,0.6987,0.6987,0.7001]]])
    elif N_circ == 14:
        V_y =
np.array([[[0.7004,0.6991,0.6991,0.7004],[0.7008,0.6996,0.6996,0.7008]],[[0.7851,0.7841,0.7841,0.78
51],[0.7836,0.7826,0.7826,0.7836]],[[0.7004,0.6991,0.6991,0.7004],[0.7008,0.6996,0.6996,0.7008]]])
    elif N_circ == 15:
        V_y =
np.array([[[0.7012,0.7001,0.7001,0.7012],[0.7015,0.7005,0.7005,0.7015]],[[0.7820,0.7811,0.7811,0.78
20],[0.7804,0.7796,0.7796,0.7804]],[[0.7012,0.7001,0.7001,0.7012],[0.7015,0.7005,0.7005,0.7015]]])
    elif N_circ == 16:
        V_y =
np.array([[[0.7020,0.7010,0.7010,0.7020],[0.7023,0.7014,0.7014,0.7023]],[[0.7788,0.7780,0.7780,0.77
88],[0.7770,0.7764,0.7764,0.7770]],[[0.7020,0.7010,0.7010,0.7020],[0.7023,0.7014,0.7014,0.7023]]])
    elif N_circ == 17:
        V_y =
np.array([[[0.7027,0.7019,0.7019,0.7027],[0.7031,0.7023,0.7023,0.7031]],[[0.7753,0.7747,0.7747,0.77
53],[0.7735,0.7729,0.7729,0.7735]],[[0.7027,0.7019,0.7019,0.7027],[0.7031,0.7023,0.7023,0.7031]]])
    elif N_circ == 18:
        V_y =
np.array([[[0.7035,0.7028,0.7028,0.7035],[0.7038,0.7032,0.7032,0.7038]],[[0.7716,0.7711,0.7711,0.77
16],[0.7697,0.7692,0.7692,0.7697]],[[0.7035,0.7028,0.7028,0.7035],[0.7038,0.7032,0.7032,0.7038]]])
    elif N_circ == 19:
        V_y =
np.array([[[0.7042,0.7037,0.7037,0.7042],[0.7045,0.7040,0.7040,0.7045]],[[0.7677,0.7673,0.7673,0.76
77],[0.7657,0.7654,0.7654,0.7657]],[[0.7042,0.7037,0.7037,0.7042],[0.7045,0.7040,0.7040,0.7045]]])
    elif N_circ == 20:
        V_y =
np.array([[[0.7049,0.7045,0.7045,0.7049],[0.7051,0.7047,0.7047,0.7051]],[[0.7636,0.7633,0.7633,0.76
36],[0.7614,0.7613,0.7613,0.7614]],[[0.7049,0.7045,0.7045,0.7049],[0.7051,0.7047,0.7047,0.7051]]])
    elif N_circ == 21:
        V_y =
np.array([[[0.7054,0.7052,0.7052,0.7054],[0.7056,0.7054,0.7054,0.7056]],[[0.7592,0.7591,0.7591,0.75
92],[0.7570,0.7570,0.7570,0.7570]],[[0.7054,0.7052,0.7052,0.7054],[0.7056,0.7054,0.7054,0.7056]]])
    elif N_circ == 22:
        V_y =
np.array([[[0.7059,0.7058,0.7058,0.7059],[0.7060,0.7059,0.7059,0.7060]],[[0.7547,0.7547,0.7547,0.75
47],[0.7523,0.7524,0.7524,0.7523]],[[0.7059,0.7058,0.7058,0.7059],[0.7060,0.7059,0.7059,0.7060]]])
    elif N_circ == 23:
        V_y =
np.array([[[0.7063,0.7063,0.7063,0.7063],[0.7063,0.7064,0.7064,0.7063]],[[0.7499,0.7501,0.7501,0.74
```

```
99],[0.7474,0.7477,0.7477,0.7474]],[[0.7063,0.7063,0.7063,0.7063],[0.7063,0.7064,0.7064,0.7063]]])
    elif N_circ == 24:
        V_y =
np.array([[[0.7065,0.7066,0.7066,0.7065],[0.7064,0.7067,0.7067,0.7064]],[[0.7449,0.7452,0.7452,0.74
49],[0.7423,0.7427,0.7427,0.7423]],[[0.7065,0.7066,0.7066,0.7065],[0.7064,0.7067,0.7067,0.7064]]])
    elif N_circ == 25:
        V_y =
np.array([[[0.7065,0.7069,0.7069,0.7065],[0.7063,0.7068,0.7068,0.7063]],[[0.7397,0.7402,0.7402,0.73
97],[0.7370,0.7376,0.7376,0.7370]],[[0.7065,0.7069,0.7069,0.7065],[0.7063,0.7068,0.7068,0.7063]]])
    elif N_circ == 26:
        V_y =
np.array([[[0.7063,0.7069,0.7069,0.7063],[0.7060,0.7067,0.7067,0.7060]],[[0.7343,0.7350,0.7350,0.73
43],[0.7315,0.7323,0.7323,0.7315]],[[0.7063,0.7069,0.7069,0.7063],[0.7060,0.7067,0.7067,0.7060]]])
    elif N_circ == 27:
        V_y =
np.array([[[0.7059,0.7067,0.7067,0.7059],[0.7055,0.7064,0.7064,0.7055]],[[0.7287,0.7296,0.7296,0.72
87],[0.7259,0.7269,0.7269,0.7259]],[[0.7059,0.7067,0.7067,0.7059],[0.7055,0.7064,0.7064,0.7055]]])
    elif N_circ == 28:
        V_y =
np.array([[[0.7053,0.7063,0.7063,0.7053],[0.7047,0.7059,0.7059,0.7047]],[[0.7230,0.7242,0.7242,0.72
30],[0.7201,0.7214,0.7214,0.7201]],[[0.7053,0.7063,0.7063,0.7053],[0.7047,0.7059,0.7059,0.7047]]])
    elif N_circ == 29:
        V_y =
np.array([[[0.7044,0.7057,0.7057,0.7044],[0.7037,0.7052,0.7052,0.7037]],[[0.7173,0.7187,0.7187,0.71
73],[0.7144,0.7160,0.7160,0.7144]],[[0.7044,0.7057,0.7057,0.7044],[0.7037,0.7052,0.7052,0.7037]]])
    elif N_circ == 30:
        V_y =
np.array([[[0.7033,0.7049,0.7049,0.7033],[0.7025,0.7042,0.7042,0.7025]],[[0.7117,0.7133,0.7133,0.71
17],[0.7089,0.7107,0.7107,0.7089]],[[0.7033,0.7049,0.7049,0.7033],[0.7025,0.7042,0.7042,0.7025]]])


    # 2. Heat exchanger geometry

    D_o = 7.0e-3 #tube outer diameter [m]
    D_i = 6.3e-3 #tube inner diameter [m]
    P_t = 25e-3 #transversal tube pitch [m]
    P_l = 22e-3 #longitudinal tube pitch [m]
    t_fin = 0.092e-3 #fin thickness [m]
    P_f = 1.7e-3 #fin pitch[m]
    F_s = P_f-t_fin #fin space[m]
    D_cd = D_o+(2*t_fin)  #fin collar diameter [m]
    e_surf = 0.002e-03 #tube surface roughness
    N_tr = 3 #number of tube rows
    N = 2 #number of tubes per row per circuit
    N_th = 2 #number of tubes high per circuit
    N_tt = N*N_tr #total number of tubes per circuit
    L_tot = 1720e-03 #length of an evaporator tube (heat exchanger width)[m]
    L_tube_tt = L_tot*6 #total length of all the tubes per circuit[m]
    A_i = (math.pi*(D_i)**2)/4 #inner tube free-flow area[m^2]
    HX_height = 2*P_t #heat exchanger height[m]
    HX_width = L_tot #heat exchanger width[m]
    HX_depth = P_l*N_tr #heat exchanger depth[m]
    N_fins_face = L_tot/(P_f+t_fin) #total number of fins on a heat exchanger face
    A_af = HX_height*HX_width #air-side frontal area [m^2]
    V_tot = A_af*HX_depth #Heat exchanger core volume [m^3]
    A_ht_r_tt = math.pi*D_i*L_tot*N_tt #tube-side total heat transfer surface area [m^2]
    A_ht_r_over_V_tot = A_ht_r_tt/V_tot #tube-side heat transfer area per total volume [m^2/m^3]
    N_fin_l = N_fins_face/L_tot #number of fins per unit length [1/m]
    A_ht_pipes = math.pi*D_o*(L_tot-(t_fin*N_fins_face))*N_tt #heat transfer area associated to
exposed pipes/tubes[m^2]
    A_ht_fins = 2*((HX_depth*HX_height)-((math.pi*(D_o)**2)*N_tt/4))*L_tot*N_fin_l #heat transfer
area due to fins [m^2]
    A_ht_a_tt = A_ht_pipes + A_ht_fins #air-side total heat transfer surface area [m^2]
    A_ht_a_over_V_tot = A_ht_a_tt/V_tot #air-side heat transfer area over total volume [m^2/m^3]
    A_ht_ratio = (A_ht_r_over_V_tot/A_ht_a_over_V_tot) #tube side/air side total heat transfer area
ratio

    D_h_i = (4*A_i)/(math.pi*D_i) #inner tube hydraulic diameter[m]
    P_h = math.pi*D_h_i  #wetted perimeter[m]
    alpha = 0.5 #relaxation parameter
    D_h_o = 3.63e-3 #flow passage hydraulic diameter in metres[Kays and London]
    A_ff_over_A_af = 0.534 #free flow area/frontal area ratio
    V_a_max_i = np.array([x*(P_t/(P_t-D_o)) for x in V_y]) #increments' maximum velocities for
staggared tube arrangement
```

```
    V_a_max = V_a_max_i.max() #maximum velocity of air [m/s]

    #air-side efficiency
    A_fin_over_A = A_ht_fins/(A_ht_pipes + A_ht_fins) #fin area over total area
    k_Al = 186 #thermal conductivity of 2024 T6 Al alloy in W/m*K
    L_c = P_l*0.5 #Length in metres of fins between tubes(= tube spacing*0.5)
    A_p = L_c*t_fin
    effec_fin = 0.952 #by calculation (L_c**(3/2))*(h_a_sh/(k_Al*A_p))**0.5 from Frank Incropera
    effec_o = 1-A_fin_over_A*(1-effec_fin) #overall air side efficiency

    #number of increments per tube
    incr_n = 4

    #Geometry calculations per increment
    L = [0]*N_tr*N*incr_n #length of each increment
    A_ht_r = [0]*N_tr*N*incr_n #initialising tube side heat transfer area
    A_ht_a = [0]*N_tr*N*incr_n #initialising air side heat transfer area
    A_ht_ext_tot = [0]*N_tr*N*incr_n #initialising air side total heat transfer area for each
increment
    x_L = [0]*N_tr*N*incr_n #along tube length i.e. x-axis
    z_depth = [0]*N_tr*N*incr_n #for plotting purposes across tube height
    m_air = np.zeros((N_tr,N,incr_n)) #initialising air mass flow rate on each circuit per row
[kg/s]
    m_a = [0]*N_tr*N*incr_n #initiating mass flow per increment[kg/s]
    V_a = [0]*N_tr*N*incr_n #initiating air velocity per increment[m/s]
    L_e = 0.9 #Lewis number

    # 2. Operating Air and Refrigerant Sides Conditions

     #Air side inlet
    T_a_i = 7+273.15  #dry-bulb temperature[K]
    P_a_i = 101325  #air pressure[Pa]
    T_wb_i = 6+273.15 #wet bulb temperature @inlet[K]
    w_a_i = fp.HAPropsSI('W','T',T_a_i,'P', P_a_i,'Twb',T_wb_i) #Dry bulb,wet bulb temperatures &
Pressures given
    h_a_i = fp.HAPropsSI('H','P', P_a_i,'T', T_a_i,'W', w_a_i) #inlet air enthalpy[J/kg]
    v_a_inlet = fp.HAPropsSI('Vha','T', T_a_i,'W', w_a_i,'P',p) #inlet air specific volume
    dens_inlet = 1/v_a_inlet #inlet air density

     #Air side outlet
    T_a_e = 7+273.15  #dry-bulb temperature[K]
    P_a_e = 98613.24 #air pressure[Pa]
    T_wb_e = 6+273.15 #wet bulb temperature @outlet[K]
    w_a_e = fp.HAPropsSI('W','T',T_a_e,'P', P_a_e,'Twb',T_wb_e) #[kg/kg dry air]
    h_a_e = fp.HAPropsSI('H','P', P_a_e,'T', T_a_e,'W', w_a_e) #outlet air enthalpy[J/kg]

     #Refrigerant conditions at heat exchanger inlet
    P_r_i = 876.05e03 #refrigerant pressure[Pa]
    T_r_i = 3+273.15 #refrigerant temperature[K]
    m_ref = 0.02639/30 #[kg/s] #refrigerant mass flow rate per circuit[kg/s]
    m_ref_circuit = m_ref #refrigerant mass flow rate per circuit[kg/s]
    h_r_i = fp.PropsSI('H','P',P_r_i,'T',T_r_i,'R410a') #refrigerant enthalpy[J/kg]

     #Refrigerant conditions at heat exchanger exit
    T_r_e = 5.90+273.15 #refrigerant temperature[K]
    P_r_e = 370e03  #refrigerant pressure[Pa]
    h_r_e = fp.PropsSI('H','P',P_r_e,'T',T_r_e,'R410a') #refrigerant enthalpy[J/kg]

    for i in range(N_tr):
        for j in range(N):
            for k in range(1,N_tr*N*incr_n):
                L[k] = L_tot/incr_n
                x_L[k] = x_L[k-1]+L[k]
                A_ht_r[k] = math.pi*D_i*L[k]
                A_ht_a[k] = A_ht_r[k]/A_ht_ratio
                A_ht_ext_tot[k] = A_ht_a_tt/(N_tr*N*incr_n)
                z_depth[k] = z_depth[k-1] + P_l
                if k%4 == 0:
                    m_air[i,j,0] = dens_inlet*V_y[i,j,0]*A_ht_ext_tot[k] #mass flow rate per
increment[kg/s]
                    m_a[k] =  m_air[i,j,0]
                    V_a[k] = V_y[i,j,0]
                elif k%4 == 1:
                    m_air[i,j,1] = dens_inlet*V_y[i,j,1]*A_ht_ext_tot[k] #mass flow rate per
```

```
increment[kg/s]
                    m_a[k] =  m_air[i,j,1]
                    V_a[k] = V_y[i,j,1]
                elif k%4 == 2:
                    m_air[i,j,2] = dens_inlet*V_y[i,j,2]*A_ht_ext_tot[k] #mass flow rate per
increment[kg/s]
                    m_a[k] =  m_air[i,j,2]
                    V_a[k] = V_y[i,j,2]
                else:
                    m_air[i,j,3] = dens_inlet*V_y[i,j,3]*A_ht_ext_tot[k] #mass flow rate per
increment[kg/s]
                    m_a[k] =  m_air[i,j,3]
                    V_a[k] = V_y[i,j,3]

    # 4. Primary (refrigerant) side properties initialising)
    P_r = [0]*N_tr*N*incr_n  #initialising pressure at each increment
    h_r = [0]*N_tr*N*incr_n  #initialising enthalpy at each increment
    Q_dot_p = [0]*N_tr*N*incr_n #initialising heat transfer at each increment on the tube side
    Psat_r = [0]*N_tr*N*incr_n #initialising refrigeration saturation pressure at each increment on
the tube side

    P_r[0] = P_r_i
    P_r[N_tr*N*incr_n-1] = P_r_e
    deltaP_r = (P_r[N_tr*N*incr_n-1]-P_r[0])/(N_tr*N*incr_n)
    h_r[0] = h_r_i
    h_r[N_tr*N*incr_n-1] = h_r_e
    deltah_r = (h_r[N_tr*N*incr_n-1]-h_r[0])/(N_tr*N*incr_n)
    for i in range(N_tr):
        for j in range(N):
            for k in range(1,N_tr*N*incr_n):
                P_r[k] = P_r[k-1]+deltaP_r
                h_r[k] = h_r[k-1]+deltah_r
                Q_dot_p[k] = m_ref_circuit*(h_r[k]-h_r[k-1])

    T_r = fp.PropsSI('T','P',P_r,'H',h_r,'R410a') #refrigerant temperature[K]
    s_r = fp.PropsSI('S','P',P_r,'H',h_r,'R410a') #refrigerant entropy[J/K]
    Psat_r = fp.PropsSI('P','T',T_r,'Q',1.0,'R410a') #saturated pressure[Pa]
    x_r = fp.PropsSI('Q','P',Psat_r,'H',h_r,'R410a') #quality of the refrigerant
    x_r[x_r==inf] = -1.0

    # 5. Secondary (humid air) side properties initialising
    P_a = [0]*N_tr*N*incr_n  #initialising pressure at each increment
    h_a = [0]*N_tr*N*incr_n #initialising enthalpy at each increment
    w_a = [0]*N_tr*N*incr_n #initialising humidity ratios at each increment
    T_sw = [0]*N_tr*N*incr_n #initialising surface wall temperatures
    w_sw = [0]*N_tr*N*incr_n #initialising surface wall humidity ratios
    Q_dot_s = [0]*N_tr*N*incr_n #initialising total heat transfer on the fin side
    Q_dot_sen = [0]*N_tr*N*incr_n #sensible heat transfer
    Q_dot_lat = [0]*N_tr*N*incr_n #latent heat transfer
    deltaw_a = [0]*N_tr*N*incr_n #change in humidity

    P_a[0] = P_a_e
    P_a[-1] = P_a_i
    deltaP_a = (P_a[-1]-P_a[0])/(N_tr*N*incr_n)

    h_a[0] = h_a_e
    h_a[-1] = h_a_i
    deltah_a = (h_a[-1]-h_a[0])/(N_tr*N*incr_n)

    w_a[0] = w_a_e
    w_a[-1] = w_a_i
    dw_a = (w_a[-1]-w_a[0])/(N_tr*N*incr_n)

    for i in range(N_tr): #tube rows
        for j in range(N): #tube circuits
            for k in range(1,N_tr*N*incr_n): #increments
                P_a[k] = P_a[k-1]+deltaP_a
                h_a[k] = h_a[k-1]+deltah_a
                w_a[k] = w_a[k-1]+dw_a
                Q_dot_s[k] = m_a[k]*(h_a[k-1]-h_a[k])
                Q_dot_sen[k] = Q_dot_s[k]
                Q_dot_lat[k] = 0.0
                deltaw_a[k] = w_a[k-1]-w_a[k]
```

```
T_a = fp.HAPropsSI('T','P',P_a,'H',h_a,'W',w_a) #air temperature[K]
s_a = fp.HAPropsSI('S','P',P_a,'H',h_a,'W',w_a) #air entropy[J/K]

# 6. Guessing wall surface temperature and humidity ratio
for i in range(N_tr):
    for j in range(N):
        for k in range(1,N_tr*N*incr_n):
            T_sw[k] = (T_a[k]+T_r[k])/2 #film temperature guess
            w_sw[k] = fp.HAPropsSI('W','P', P_a[k],'T', T_sw[k],'R', 1.0)
T_sw[0] = T_sw[1]
w_sw[0] = w_sw[1]

# 7. Matrix initialisation
#enthalpies calculations
X_p = np.zeros((N_tr*N*incr_n,N_tr*N*incr_n))
Y_p = [0]*N_tr*N*incr_n
Z_p = [0]*N_tr*N*incr_n
X_s = np.zeros((N_tr*N*incr_n,N_tr*N*incr_n))
Y_s = [0]*N_tr*N*incr_n
Z_s = [0]*N_tr*N*incr_n

#pressure calculations
XX_p= np.zeros((N_tr*N*incr_n,N_tr*N*incr_n))
  YY_p = [0]*N_tr*N*incr_n
ZZ_p = [0]*N_tr*N*incr_n
XX_s = np.zeros((N_tr*N*incr_n,N_tr*N*incr_n))
YY_s = [0]*N_tr*N*incr_n
ZZ_s = [0]*N_tr*N*incr_n

#humidity ratio calculations
XXX_s = np.zeros((N_tr*N*incr_n,N_tr*N*incr_n))
YYY_s = [0]*N_tr*N*incr_n
ZZZ_s = [0]*N_tr*N*incr_n

# 8. Initialisations of heat transfers, surface wall temperature and humidity ratio:
max_err = 1e20
itera = 0.0
Q_dot_p_old = [0]*N_tr*N*incr_n
Q_dot_s_old = [0]*N_tr*N*incr_n
Q_dot_sen_old = [0]*N_tr*N*incr_n
Q_dot_lat_old = [0]*N_tr*N*incr_n
T_sw_old = [0]*N_tr*N*incr_n
deltaw_a_old = [0]*N_tr*N*incr_n

while max_err>1e-04: #Main iteration loop which test convergence of heat transfer rates

    itera = 1+itera

    for i in range(N_tr):
        for j in range(N):
            for k in range(1,N_tr*N*incr_n):
                Q_dot_p_old[k] = Q_dot_p[k]
                Q_dot_s_old[k] = Q_dot_s[k]
                Q_dot_sen_old[k] = Q_dot_sen[k]
                Q_dot_lat_old[k] = Q_dot_lat[k]
                T_sw_old[k] = T_sw[k]
                deltaw_a_old[k] = deltaw_a[k]

    #Primary side average conditions
    P_r_ave = [0]*N_tr*N*incr_n
    h_r_ave = [0]*N_tr*N*incr_n
    T_r_ave = [0]*N_tr*N*incr_n
    Cp_r = [0]*N_tr*N*incr_n
    for i in range(N_tr):
        for j in range(N):
            for k in range(1,N_tr*N*incr_n):
                P_r_ave[k] = (P_r[k]+P_r[k-1])/2
                h_r_ave[k] = (h_r[k]+h_r[k-1])/2
                T_r_ave[k] = (T_r[k]+T_r[k-1])/2
                if x_r[k] > 0:
                    Cp_r[k] = 1e10
                else:
                    Cp_r[k] =(h_r[k]-h_r[k-1])/(T_r[k]-T_r[k-1])
```

```
        #secondary side average conditions
        P_a_ave = [0]*N_tr*N*incr_n
        h_a_ave = [0]*N_tr*N*incr_n
        T_a_ave = [0]*N_tr*N*incr_n
        w_a_ave = [0]*N_tr*N*incr_n
        T_sw_ave = [0]*N_tr*N*incr_n
        w_sw_ave = [0]*N_tr*N*incr_n
        Cp_a = [0]*N_tr*N*incr_n
        P_a_ave[0] = P_a[0]
        h_a_ave[0] = h_a[0]
        T_a_ave[0] = T_a[0]
        w_a_ave[0] = w_a[0]
        T_sw_ave[0] = T_sw[0]
        w_sw_ave[0] = w_sw[0]
        Cp_a[0] = (h_a[1]-h_a[0])/(T_a[1]-T_a[0])
        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    P_a_ave[k] = (P_a[k]+P_a[k-1])/2
                    h_a_ave[k] = (h_a[k]+h_a[k-1])/2
                    T_a_ave[k] = (T_a[k]+T_a[k-1])/2
                    w_a_ave[k] = (w_a[k]+w_a[k-1])/2
                    T_sw_ave[k] = (T_sw[k]+T_sw[k-1])/2
                    w_sw_ave[k] = (w_sw[k]+w_sw[k-1])/2
                    Cp_a[k] = fp.HAPropsSI('Cha','T', T_a_ave[k],'W', w_a_ave[k],'P',p)


    # 9. Heat transfer coefficients

        #Air-side
        #initializing viscosity, thermal conductivity,specific volume, density and reynolds number
of air at each increment;
        visc_a = [0]*N_tr*N*incr_n
        K_a = [0]*N_tr*N*incr_n
        v_a = [0]*N_tr*N*incr_n
        dens_a = [0]*N_tr*N*incr_n
        Re_a = [0]*N_tr*N*incr_n
        Pr_a = [0]*N_tr*N*incr_n #Prandtl number initialization at each increment
        Nu_a = [0]*N_tr*N*incr_n #Nusselt number initialization at each increment
        h_a = [0]*N_tr*N*incr_n #heat transfer at each increment
        htc_a_sp = [0]*N_tr*N*incr_n
        P_w_tp = [0]*N_tr*N*incr_n #initializing vapour pressure at each increment[Pa]
        h_gw_tp = [0]*N_tr*N*incr_n #enthalpy change at vapour pressure [J/kg]
        UA_o = [0]*N_tr*N*incr_n #air UA values at each increment[W/K]
        x_value = [0]*N_tr*N*incr_n #log(Reynolds number)
        y_value = [0]*N_tr*N*incr_n #log(j_H)
        grad = -0.3948 #gradient of j_h vs Reynolds number graph(from Kays and London)
        y_inter = -0.7773 #y_intercept of the graph
        j_h_a = [0]*N_tr*N*incr_n #Colburn j-factor
        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    visc_a[k] = fp.HAPropsSI('M','T',T_a_ave[k],'W',w_a_ave[k],'P',P_a_ave[k])
                    K_a[k] = fp.HAPropsSI('K','T',T_a_ave[k],'W',w_a_ave[k],'P',P_a_ave[k])
                    v_a[k] = fp.HAPropsSI('Vha','T', T_a_ave[k],'W', w_a_ave[k],'P',P_a_ave[k])
                    dens_a[k] =1/v_a[k]
                    Re_a[k] = (dens_a[k]*V_a[k]*D_o)/visc_a[k]


                    #determining Colburn j-factor from j_H vs Reynolds number graph(from Kays and
London)
                    x_value[k] = np.log10(Re_a[k])
                    y_value[k] = grad*x_value[k]+y_inter #equation of straight line
                    j_h_a[k] = 10**(y_value[k])
                    Pr_a[k] = abs((Cp_a[k]*visc_a[k])/K_a[k])
                    Nu_a[k] = j_h_a[k]*Re_a[k]*(Pr_a[k])**(1/3)
                    htc_a_sp[k] = (Nu_a[k]*K_a[k])/D_h_o
                    P_w_tp[k] = fp.HAPropsSI('P_w','T',T_a_ave[k],'W',w_a_ave[k],'P',p)
                    h_gw_tp[k] = fp.HAPropsSI('Hha','T',T_a_ave[k],'W',w_a_ave[k],'P',P_w_tp[k])
                    UA_o[k] = m_ref*(h_r[k]-h_r[k-1])*((T_sw[k]-T_a_ave[k])+((w_sw[k]-
w_a_ave[k])*h_gw_tp[k]/Cp_a[k]))**-1

        #Refrigerant superheat section
```

```
        #C_ref's and C_air's
        #initialisizing C_ref's, C_air's, Q_max, effectiveness, NTU and UA_i;
        C_r_sh = [0]*N_tr*N*incr_n
        C_a_sh = [0]*N_tr*N*incr_n
        C_min_sh = [0]*N_tr*N*incr_n
        C_max_sh = [0]*N_tr*N*incr_n
        Cr_sh = [0]*N_tr*N*incr_n
        Q_max_sh = [0]*N_tr*N*incr_n
        effec = [0]*N_tr*N*incr_n
        NTU = [0]*N_tr*N*incr_n
        UA_i_sh = [0]*N_tr*N*incr_n

        dens_r_sh = [0]*N_tr*N*incr_n #refrigerant density at each increment
        visc_r_sh = [0]*N_tr*N*incr_n #viscosity at each increment
        K_r_sh = [0]*N_tr*N*incr_n #thermal conductivity
        Pr_r_sh = [0]*N_tr*N*incr_n #Prandtl number
        V_r_sh = [0]*N_tr*N*incr_n #refrigerant velocity
        Re_r_sh = [0]*N_tr*N*incr_n #Reynolds number
        Nu_r_sh = [0]*N_tr*N*incr_n #Nusselt number
        ht_r_sh = [0]*N_tr*N*incr_n #heat transfer[W/m2.K]

        #Primary side heat transfer at different regions

        UA_i_tp = [0]*N_tr*N*incr_n #refrigerant UA values at each increment[W/K]
        h_1_tp= [0]*N_tr*N*incr_n #single phase heat transfer coefficient [W/m2.K]
        Bo = [0]*N_tr*N*incr_n
        ht_r_tp = [0]*N_tr*N*incr_n #two-phase refrigerant heat transfer coefficient
        dens_r_tp = [0]*N_tr*N*incr_n #refrigerant density at each increment
        visc_r_tp = [0]*N_tr*N*incr_n #viscosity at each increment
        K_r_tp = [0]*N_tr*N*incr_n #thermal conductivity
        Pr_r_tp = [0]*N_tr*N*incr_n #Prandtl number
        V_r_tp = [0]*N_tr*N*incr_n #refrigerant velocity
        Re_r_tp = [0]*N_tr*N*incr_n #Reynolds number
        Q_sh = [0]*N_tr*N*incr_n #heat transfer in superheat region
        Q_tp = [0]*N_tr*N*incr_n #heat transfer in two-phase


        htc_r = [0]*N_tr*N*incr_n
        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    Q_sh[k] = m_ref*(h_r[k]-h_r[k-1])
                    C_r_sh[k] = m_ref*Cp_r[k]
                    C_a_sh[k] = m_a[k]*Cp_a[k] #C_ref and C_air on each increment in the superheat
section
                    C_min_sh[k] = min(C_r_sh[k],C_a_sh[k])
                    C_max_sh[k] = max(C_r_sh[k],C_a_sh[k])
                    Cr_sh[k] = C_min_sh[k]/C_max_sh[k]
                    Q_max_sh[k] = C_min_sh[k]*(T_a[k]-T_r[k-1]) #max heat trasfer rate in
superheated section; Q_max=C_min*deltaT_max
                    effec[k] = abs(Q_sh[k]/Q_max_sh[k]) #effectiveness


                    if Cr_sh[k] < 1e-06:
                        NTU[k] = -1*np.log(1-effec[k]) #NTU calculation at each increment

                    else:
                        NTU[k] = -(np.log((Cr_sh[k]*np.log(1-effec[k]))+1))/Cr_sh[k] #for cross-
flow

                    UA_i_sh[k] = NTU[k]*C_min_sh[k] #UA value of refrigerant only at each
increment[W/K]

                    dens_r_sh[k] = fp.PropsSI('D','P',P_r[k],'Q',1,'R410a')
                    visc_r_sh[k] = fp.PropsSI('V','P',P_r[k],'Q',1,'R410a')
                    K_r_sh[k] = fp.PropsSI('L','P',P_r[k],'Q',1,'R410a')
                    Pr_r_sh[k] = abs((Cp_r[k]*visc_r_sh[k])/K_r_sh[k])
                    V_r_sh[k] = m_ref/(dens_r_sh[k]*A_i)
                    Re_r_sh[k] = (dens_r_sh[k]*V_r_sh[k]*D_h_i)/visc_r_sh[k]
                    Nu_r_sh[k] = 0.023*((Re_r_sh[k])**0.8)*(Pr_r_sh[k]**0.4) #Dittus Boelter
equation
                    ht_r_sh[k] = (Nu_r_sh[k]*K_r_sh[k])/D_h_i

                    Q_tp[k] = m_ref*(h_r[k]-h_r[k-1])
```

```
                    UA_i_tp[k] = abs(Q_tp[k]/(T_sw[k]-T_r[k]))
                    dens_r_tp[k] = fp.PropsSI('D','P',P_r[k],'Q',abs(x_r[k]),'R410a')
                    visc_r_tp[k] = fp.PropsSI('V','P',P_r[k],'Q',abs(x_r[k]),'R410a')
                    K_r_tp[k] = fp.PropsSI('L','P',P_r[k],'Q',abs(x_r[k]),'R410a')
                    Pr_r_tp[k] = abs((Cp_r[k]*visc_r_tp[k])/K_r_tp[k])
                    V_r_tp[k] = m_ref/(dens_r_tp[k]*A_i)
                    Re_r_tp[k] = (dens_r_tp[k]*V_r_tp[k]*D_h_i)/visc_r_tp[k]
                    h_1_tp[k] = 0.023*((Re_r_tp[k])**0.8)*((Pr_r_tp[k])**0.4)*(K_r_tp[k]/D_h_i)
                    h_f = fp.PropsSI('H','P',P_r[k],'Q',0,'R410a')
                    h_g = fp.PropsSI('H','P',P_r[k],'Q',1,'R410a')
                    h_fg = h_g-h_f
                    Bo[k] = Q_tp[k]/((m_ref/A_i)*h_fg*A_ht_r[k])
                    ht_r_tp[k] = h_1_tp[k]*(4.3+0.4*(Bo[k]*10000)**1.3) #Chato/Wattelet
correlation; D. M. Admiraal and C. W. Bullard,1993, Heat transfer in refrigerator condensers and
evaporators.

                    if x_r[k] == -1 and x_r[k-1] == -1: #single phase region
                        htc_r[k] = ht_r_sh[k]

                    elif x_r[k]>0 and x_r[k-1]>0: #two phase region
                        htc_r[k] = ht_r_tp[k]

                    else: #transitions from one phase to another
                        if (x_r[k-1] == -1.0 and x_r[k] > 0.0) and (h_r[k] < h_r[k-1]):#superheat
to two-phase
                            h_g = fp.PropsSI('H','P',P_r_ave[k],'Q',1.0,'R410a')
                            deltah_sp = h_g-h_r[k-1]
                            deltah_tp = h_r[k]-h_g
                        elif x_r[k] == -1.0 and x_r[k-1]>0 and h_r[k] < h_r[k-1]: #two-phase to
subcooled
                            h_f = fp.PropsSI('H','P',P_r_ave[k],'Q',0.0,'R410a')
                            deltah_tp = h_f-h_r[k-1]
                            deltah_sp = h_r[k]-h_f
                        elif x_r[k]>0 and x_r[k-1] == -1.0 and h_r[k] > h_r[k-1]: #subcooled to
two-phase
                            h_f = fp.PropsSI('H','P',P_r_ave[k],'Q',0.0,'R410a')
                            deltah_sp = h_f-h_r[k-1]
                            deltah_tp = h_r[k]-h_f
                        elif x_r[k] == -1.0 and x_r[k-1]>0 and h_r[k] > h_r[k-1]: #two_phase to
superheat
                            h_g = fp.PropsSI('H','P',P_r_ave[k],'Q',1.0,'R410a')
                            deltah_tp = h_g-h_r[k-1]
                            deltah_sp = h_r[k]-h_g
                        C_tp = (deltah_tp*ht_r_sh[k])/(deltah_sp*ht_r_tp[k])
                        f_tp = C_tp/(1+C_tp)
                        A_tp = f_tp*A_ht_r[k]
                        A_sp = (1-f_tp)*A_ht_r[k]
                        htc_r[k] = ((A_tp*ht_r_tp[k])+(A_sp*ht_r_sh[k]))/A_ht_r[k]

        #Secondary side heat transfer(assuming no phase change)
        htc_a = [0]*N_tr*N*incr_n
        for i in range(N_tr):
            for j in range(N):
                for k in range(N_tr*N*incr_n):
                    htc_a[k] = htc_a_sp[k]

        #Overall UA-value
        UA_overall = [0]*N_tr*N*incr_n

        R_f_i = 0.0004 #C/W fouling factor of refrigerant
        R_f_a = 0.0004 #C/W fouling factor of air
        R_wall = (np.log(D_o/D_i))/2*math.pi*k_Al*L_tot  #wall resistance
        R_fouling_r = [0]*N_tr*N*incr_n
        R_fouling_a = [0]*N_tr*N*incr_n
        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    R_fouling_r[k] = R_f_i/A_ht_r[k]
                    R_fouling_a[k] = R_f_a/A_ht_a[k]
                    UA_overall = 1/(1/(R_fouling_a+R_fouling_r+R_wall+(htc_r[k]*A_ht_r[k])) +
1/(effec_o*((htc_a[k]*A_ht_a[k]))))


        #Heat transfers at each increment on both sides
```

```
        P_a_part = [0]*N_tr*N*incr_n #partial pressure of vapour in air
        h_gw = [0]*N_tr*N*incr_n #enthalpy of saturated vapor at partial pressure of vapour in air
        h_fw = [0]*N_tr*N*incr_n #enthalpy of saturated liquid at partial pressure of liquid in
water film
        h_fgw = [0]*N_tr*N*incr_n #h_gw - h_fw
        P_v_sw = [0]*N_tr*N*incr_n #vapor partial pressure @wall surface
        P_v_a = [0]*N_tr*N*incr_n #vapor partial pressure @free stream
        dens_v_sw = [0]*N_tr*N*incr_n #vapor density @wall surface
        dens_v_fs = [0]*N_tr*N*incr_n #vapor density @free stream
        v_a = [0]*N_tr*N*incr_n #specific volume of air (ideal)
        dens_a = [0]*N_tr*N*incr_n #density of air (ideal)
        M_v = [0]*N_tr*N*incr_n #vapor molar mass
        M_molar = [0]*N_tr*N*incr_n #air molar mass(ideal)

        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    P_a_part[k] = fp.HAPropsSI('P_w','P',P_a_ave[k],'H',h_a_ave[k],'W',w_a_ave[k])
                    P_v_a[k] = fp.PropsSI('P','P',P_a_ave[k],'Q',1.0,'IF97::Water')
                    h_gw[k] = fp.PropsSI('H','T',T_sw_ave[k],'Q',1.0,'IF97::Water')
                    h_fw[k] = fp.PropsSI('H','T',T_sw_ave[k],'Q',0.0,'IF97::Water')
                    h_fgw[k] = h_gw[k]-h_fw[k] #Latent heat/enthalpy of vaporization
                    dens_v_sw[k] = fp.PropsSI('D','T',T_sw_ave[k],'Q',1,'IF97::Water')
                    dens_v_fs[k] = fp.PropsSI('D','P',P_a_ave[k],'Q',1,'IF97::Water')
                    v_a[k] = fp.HAPropsSI('Vha','T', T_a[k],'W', w_a[k],'P',P_a[k]) #air specific
density
                    dens_a[k] = 1/v_a[k] # density of air
                    M_v[k] = fp.PropsSI('M','T',T_sw_ave[k],'P',P_v_sw[k],'IF97::Water') #Vapor
molar mass [kg/mol]
                    M_molar[k] = (dens_a[k]*T_a_ave[k]*8.31447)/P_a_ave[k] #Molar mass =
(dens*R_u*T_a)/P_a
                    if w_sw_ave[k] < w_a_ave[k]: # i.e. if the moisture content in free flow air is
higher than that on the tube surface
                        Q_dot_lat[k] = alpha*(((htc_a[k]*A_ht_a[k])/(L_e*Cp_a[k]))*(w_sw_ave[k]-
w_a_ave[k])*h_gw[k])+((1-alpha)*Q_dot_lat_old[k]) #relaxation is applied
                    else:
                        Q_dot_lat[k] = 0.0
                    Q_dot_sen[k] = alpha*(htc_a[k]*A_ht_a[k]*(T_sw_ave[k]-T_a_ave[k]))+((1-
alpha)*Q_dot_sen_old[k]) #relaxation is applied
                    Q_dot_s[k] = -(Q_dot_sen[k]-Q_dot_lat[k])
                    Q_dot_p[k] = Q_dot_sen[k]+(Q_dot_lat[k]*(h_fgw[k]/h_gw[k]))
                    T_sw[k] = (alpha*(T_a_ave[k] - (Q_dot_lat[k]/(htc_a[k]*A_ht_a[k])) +
((1/(Cp_a[k]*L_e))*(w_sw_ave[k]-w_a_ave[k])*h_gw[k])))+((1-alpha)*T_sw_old[k])
                    deltaw_a[k] = Q_dot_lat[k]/(m_a[k]*h_fgw[k])
        T_sw[0] = T_sw[1]
        w_sw[0] = w_sw[1]
        Q_dot_s_tot = np.sum(Q_dot_s)
        Q_dot_p_tot = np.sum(Q_dot_p)
        Q_dot_sen_tot = np.sum(Q_dot_sen)
        Q_dot_lat_tot = np.sum(Q_dot_lat)

        #Pressure losses

        #Position of each node along the length of the tube
        x_l = [0]*N_tr*N*incr_n
        x_l[0] = 0.0

        for i in range(N_tr):
            for j in range(N):
                for k in range(1,incr_n):
                    x_l[k] = L[k]+x_l[k-1]

        #enthalpy changes of a refrigerant
        h_r_fg_tp = [0]*N_tr*N*incr_n

        #non-dimensional Darcy Weisbach friction factors
        k_f = [0]*N_tr*N*incr_n #Pierre boiling number
        fric_f_i_sh = [0]*N_tr*N*incr_n
        fric_f_i_tp = [0]*N_tr*N*incr_n
        fric_f_o = [0]*N_tr*N*incr_n


        #surface roughness/hydraulic diameter
```

```
        e_dia_ratio_i = e_surf/D_h_i
        e_dia_ratio_o = e_surf/D_h_o

        #Pressure losses per increment
        deltaP_r = [0]*N_tr*N*incr_n
        deltaP_a = [0]*N_tr*N*incr_n

        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n): #air-side pressure drop
                    fric_f_o[k] = 0.25*(np.log((0.27*e_dia_ratio_o)+(5.74/(Re_a[k])**0.9)))**-2
                    deltaP_a[k] =
((fric_f_o[k]*L[k]*A_ht_r_over_V_tot)/(A_ff_over_A_af)**3)*((m_a[k])**2/(2*dens_a[k]*(A_af/N_tr*N*i
ncr_n)**2)) #air-side pressure

                    #tube-side pressure drop
                    if x_r[k] == -1 and x_r[k-1] == -1: #single phase region
                        fric_f_i_sh[k] =
0.25*(np.log((0.27*e_dia_ratio_i)+(5.74/(Re_r_sh[k])**0.9)))**-2
                        deltaP_r[k] = fric_f_i_sh[k]*(L[k]/D_h_i)*0.5*dens_r_sh[k]*(V_r_sh[k])**2

                    else: #two_phase region
                        fric_f_i_tp[k] =
0.25*(np.log((0.27*e_dia_ratio_i)+(5.74/(Re_r_tp[k])**0.9)))**-2
                        deltaP_r[k] = fric_f_i_tp[k]*(L[k]/D_h_i)*0.5*dens_r_tp[k]*(V_r_tp[k])**2
        deltaP_r_tot = np.sum(deltaP_r)
        deltaP_a_tot = np.sum(deltaP_a)

        #Primary side enthalpy matrix equations
        for j in range(N_tr*N*incr_n): #mass matrix rows
            for i in range(N_tr*N*incr_n): #mass matrix columns
                if j == i: #diagonally
                    X_p[j,i] = m_ref_circuit
                if i == j-1:
                    X_p[j,i] = -1*m_ref_circuit
        Z_p[0] = m_ref_circuit*h_r_i #boundary condition
        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    Z_p[k] = Q_dot_p[k]


        Y_p = np.linalg.solve(X_p, Z_p)

        #calculated enthalpies
        h_r = Y_p

        #Primary side pressure matrix equations
        XX_p[0,0] = 1.0
        for j in range(1,N_tr*N*incr_n):
            for i in range(N_tr*N*incr_n):
                if j == i:
                    XX_p[j,i] = -1
                elif i == j-1:
                    XX_p[j,i] = 1
        ZZ_p[0] = P_r_i
        for i in range(N_tr):
            for j in range(N):
                for k in range(1,N_tr*N*incr_n):
                    ZZ_p[k] = deltaP_r[k]


        YY_p = np.linalg.solve(XX_p, ZZ_p)

        #Calculated pressures
        P_r = YY_p

        T_r = fp.PropsSI('T','P',P_r,'H',h_r,'R410a')
        s_r = fp.PropsSI('S','P',P_r,'H',h_r,'R410a')
        Psat_r = fp.PropsSI('P','T',T_r,'Q',1.0,'R410a')
        x_r = fp.PropsSI('Q','P',Psat_r,'H',h_r,'R410a')
        x_r[x_r==inf] = -1.0

        #Secondary side enthalpy matrix equations
```

```
for k in range(N_tr): #tube rows
    for l in range(N): #tube circuit tubes
        for j in range(N_tr*N*incr_n): #mass matrix rows
            for i in range(N_tr*N*incr_n): #mass matrix columns
                if j == i:  #diagonally
                    if j%2 == 0 or i%2 == 0:
                        X_s[j,i] = m_air[k,l,0]
                    else:
                        X_s[j,i] = m_air[k,l,1]
                elif i == j+1:
                    if j%2 == 0 or i%2 == 0:
                        X_s[j,i] = -1*m_air[k,l,0]
                    else:
                        X_s[j,i] = -1*m_air[k,l,1]

Z_s[N_tr*N*incr_n-1] = m_air[0,0,0]*h_a_i #boundary condition
for i in range(N_tr):
    for j in range(N):
        for k in range(N_tr*N*incr_n-1):
            Z_s[k] = Q_dot_s[k+1]


Y_s = np.linalg.solve(X_s, Z_s)

#calculated enthalpies
h_a = Y_s

#Secondar side pressure matrix equations
XX_s[N_tr*N*incr_n-1,N_tr*N*incr_n-1] = 1.0
for j in range(N_tr*N*incr_n-1):
    for i in range(N_tr*N*incr_n):
        if j == i:
            XX_s[j,i] = -1.0
        elif i == j+1:
            XX_s[j,i] = 1.0
ZZ_s[N_tr*N*incr_n-1] = P_a_i
for i in range(N_tr):
    for j in range(N):
        for k in range(N_tr*N*incr_n-1):
            ZZ_s[k] = deltaP_a[k+1]


YY_s = np.linalg.solve(XX_s, ZZ_s)

#calculated pressures
P_a = YY_s

#Secondary side humidity ratio matrix equations
XXX_s[N_tr*N*incr_n-1,N_tr*N*incr_n-1] = 1.0
for j in range(N_tr*N*incr_n-1): #tube rows
    for i in range(N_tr*N*incr_n):#tube circuits
        if j == i:
            XXX_s[j,i] = 1
        elif i == j+1:
            XXX_s[j,i] = -1
ZZZ_s[N_tr*N*incr_n-1] = w_a_i
for i in range(N_tr):
    for j in range(N):
        for k in range(N_tr*N*incr_n-1):
            ZZZ_s[k] = deltaw_a[k+1]

YYY_s = np.linalg.solve(XXX_s, ZZZ_s)

#calculated humidity ratios
w_a = YYY_s

T_a = fp.HAPropsSI('T','P',P_a,'H',h_a,'W',w_a)
s_a = fp.HAPropsSI('S','P',P_a,'H',h_a,'W',w_a)


#Convergence test
err = [0]*N_tr*N*incr_n
for i in range(N_tr):
    for j in range(N):
```

```
                for k in range(1,N_tr*N*incr_n):
                    err[k] = max(abs((Q_dot_p[k]-Q_dot_p_old[k])/Q_dot_p[k]),(abs((Q_dot_s[k]-
Q_dot_s_old[k])/Q_dot_s[k]))) #maximum error between heat transfer on primary and secondary sides
        max_err = np.amax(err)
        print('iteration# = ', itera, '    error = {:0.3e}'.format(max_err))

    print()
    print('Circuit Number: ', N_circ)
    print('Results')
    print()
    np.set_printoptions(formatter={'float_kind':'{:0.1f}'.format})
    print('p_p = ', P_r/1e3, 'kPa')
    print('h_p = ', h_r/1e3, 'kJ/kg')
    print('T_p = ', T_r-273.15, 'C')
    np.set_printoptions(formatter={'float_kind':'{:0.3f}'.format})
    print('s_p = ', s_r/1e3, 'kJ/kgK')
    print('x_p = ',x_r, '')
    print('deltaP_p_tot = {: 0.1f}'.format(deltaP_r_tot), 'Pa')
    print()
    print('p_s = ', P_a/1e3, 'kPa')
    print('h_s = ', h_a/1e3, 'kJ/kg')
    np.set_printoptions(formatter={'float_kind':'{:0.1f}'.format})
    print('T_s = ', T_a-273.15, 'C')
    np.set_printoptions(formatter={'float_kind':'{:0.3f}'.format})
    print('s_s = ', s_a/1e3, 'kJ/kgK')
    print('deltaP_s_tot = {: 0.3f}'.format(deltaP_a_tot/1e3) , 'kPa')
    print()
    print('Q_dot_p_tot = {: 0.3f}'.format(Q_dot_p_tot/1e3), 'kW')
    print('Q_dot_s_tot = {: 0.3f}'.format(Q_dot_s_tot/1e3), 'kW')
    print('Q_dot_sen_tot = {: 0.3f}'.format(Q_dot_sen_tot/1e3), 'kW')
    print('Q_dot_lat_tot = {: 0.3f}'.format(Q_dot_lat_tot/1e3), 'kW')
    print()
    print('UA_overall = {:0.3f}'.format(np.sum(UA_overall)/len(UA_overall)),'W/K')
    print('m_dot_p = {:0.3f}'.format(m_ref),'kg/s')
    print('P_r_i = {:0.1f}'.format(P_r[0]/1e3),'kPa')
    print('P_r_e = {:0.1f}'.format(P_r[incr_n*N*N_tr-1]/1e3),'kPa')
    print('h_r_i = {:0.1f}'.format(h_r[0]/1e3),'kJ/kg')
    print('h_r_e = {:0.1f}'.format(h_r[incr_n*N*N_tr-1]/1e3),'kJ/kg')
    print('T_r_i = {:0.1f}'.format(T_r[0]-273.15),'C')
    print('T_r_e = {:0.1f}'.format(T_r[incr_n*N*N_tr-1]-273.15),'C')
    print()
    print('m_dot_s = ', m_air, 'kg/s')
    print('P_a_i = {:0.1f}'.format(P_a[incr_n*N*N_tr-1]/1e3),'kPa')
    print('P_a_e = {:0.1f}'.format(P_a[0]/1e3),'kPa')
    print('h_a_i = {:0.1f}'.format(h_a[incr_n*N*N_tr-1]/1e3),'kJ/kg dry air')
    print('h_a_e = {:0.1f}'.format(h_a[0]/1e3),'kJ/kg dry air')
    print('w_a_i = {:0.5f}'.format(w_a[incr_n*N*N_tr-1]),'kg/kg dry air')
    print('w_a_e = {:0.5f}'.format(w_a[0]),'kg/kg dry air')
    print('T_a_i = {:0.1f}'.format(T_a[incr_n*N*N_tr-1]-273.15),'C')
    print('T_a_e = {:0.1f}'.format(T_a[0]-273.15),'C')
    print()

    #plots
    fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(12, 10))
    fig.suptitle('')
    ax1.set_title('')
    ax1.set_xlabel('x_L [m]')
    ax1.set_ylabel('T [C]')
    ax1.plot(x_L,T_r-273,marker='o',color='b',alpha=0.5)
    ax1.plot(x_L,T_a-273,marker='s',color='r',alpha=0.5)
    ax2.set_title('')
    ax2.set_xlabel('x_L [m]')
    ax2.set_ylabel('P [kPa]')
    ax2.plot(x_L,P_r/1e3,marker='o',color='b',alpha=0.5)
    ax2.plot(x_L,P_a/1e3,marker='s',color='r',alpha=0.5)
    ax3.set_title('')
    ax3.set_xlabel('x_L [m]')
    ax3.set_ylabel('Q_dot [W]')
    ax3.plot(x_L,Q_dot_p,marker='o',color='b',alpha=0.5)
    ax3.plot(x_L,Q_dot_s,marker='s',color='r',alpha=0.5)
    ax4.set_title('')
    ax4.set_xlabel('T [C]')
    ax4.set_ylabel('w [kg/kg_a]')
    ax4.set_xlim(5*math.floor((T_a[0]-273)/5),5*math.ceil((T_a[incr_n*N*N_tr-1]-273)/5))
```

```
    ax4.plot(T_a-273,w_a,marker='o',color='r',alpha=0.5)

    Q_dot_s_hx = Q_dot_s_tot + Q_dot_s_hx #Total heat transfer on the air side [W]
    Q_dot_p_hx = Q_dot_p_tot + Q_dot_p_hx #Total heat transfer on the refrigerant side [W]

    deltaPress_r_tot = deltaPress_r_tot + deltaP_r_tot #Total pressure losses on the tube side [Pa]
    Q_dot_ref.append(Q_dot_p_tot)
    N_Circuits.append(N_circ)

print()
print('Total heat transfer on the air side = {: 0.3f}'.format(Q_dot_s_hx/1e3), 'kW')
print('Total heat transfer on the refrigerant side = {: 0.3f}'.format(Q_dot_p_hx/1e3), 'kW')
print('Total pressure losses on the refrigerant side = {: 0.3f}'.format(deltaPress_r_tot/1e3),
'kPa')

plt.plot(N_Circuits,Q_dot_ref,marker='o',color='b',alpha=0.5)
plt.xlabel('Circuit Number')
plt.ylabel('Heat transfer [W]')
plt.title('')
plt.tight_layout() #make room for axis labels
plt.xlim([1,30]) #Set limit to the number of circuits in x-axis
plt.show()
```

RESULTS:

```
iteration# =  1.0     error = 2.933e+00

iteration# =  2.0     error = 7.410e+01

iteration# =  3.0     error = 9.705e+00
iteration# =  4.0     error = 1.680e+02
iteration# =  5.0     error = 2.283e+01
iteration# =  6.0     error = 1.705e+00
iteration# =  7.0     error = 4.842e-01
iteration# =  8.0     error = 1.381e-01
iteration# =  9.0     error = 3.646e-02
iteration# =  10.0     error = 1.200e-01
iteration# =  11.0     error = 1.490e-01
iteration# =  12.0     error = 1.350e-01
iteration# =  13.0     error = 9.704e-02
iteration# =  14.0     error = 5.528e-02
iteration# =  15.0     error = 2.282e-02
iteration# =  16.0     error = 3.491e-03
iteration# =  17.0     error = 5.814e-03
iteration# =  18.0     error = 8.267e-03
iteration# =  19.0     error = 7.450e-03
iteration# =  20.0     error = 5.490e-03
iteration# =  21.0     error = 3.506e-03
iteration# =  22.0     error = 1.950e-03
iteration# =  23.0     error = 9.067e-04
iteration# =  24.0     error = 2.966e-04
iteration# =  25.0     error = 6.298e-05

Circuit Number:  1.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.9 220.8 227.3 234.2 241.6 249.5 257.9 266.8 276.1 286.0
 296.3 307.2 318.5 330.3 342.7 355.5 368.8 382.7 397.1 411.9 427.2 443.1] kJ/kg
T_p =  [3.0 6.8 10.9 15.2 19.9 24.7 29.9 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 44.1 59.4] C
s_p =  [1.014 1.032 1.052 1.073 1.095 1.118 1.143 1.169 1.196 1.225 1.255 1.287
 1.321 1.356 1.393 1.431 1.472 1.513 1.556 1.601 1.648 1.696 1.746 1.795] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.007 0.057 0.109 0.165 0.223
 0.285 0.349 0.416 0.486 0.560 0.635 0.714 0.797 0.882 0.970 1.000 1.000]
```

```
deltaP_p_tot =  35.9 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.314
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [21.637 21.657 21.511 21.534 21.393 21.420 21.283 21.315 21.183 21.217
 21.090 21.129 21.006 21.049 20.930 20.977 20.863 20.914 20.804 20.859
 20.754 20.813 20.712 20.775] kJ/kg
T_s =  [8.0 8.0 7.9 7.9 7.7 7.8 7.6 7.7 7.5 7.6 7.4 7.5 7.4 7.4 7.3 7.3 7.2 7.3
 7.2 7.2 7.1 7.2 7.1 7.1] C
s_s =  [0.080 0.081 0.080 0.080 0.080 0.080 0.079 0.079 0.079 0.079 0.078 0.079
 0.078 0.078 0.078 0.078 0.078 0.078 0.077 0.078 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.210 kW
Q_dot_s_tot = -0.210 kW
Q_dot_sen_tot =  0.210 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.407 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 443.1 kJ/kg
T_r_i = 3.0 C
T_r_e = 59.4 C

m_dot_s =  [[[0.222 0.221 0.221 0.222]
  [0.223 0.221 0.221 0.223]]

 [[0.257 0.256 0.256 0.257]
  [0.258 0.256 0.256 0.258]]

 [[0.222 0.221 0.221 0.222]
  [0.223 0.221 0.221 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.8 kJ/kg dry air
h_a_e = 21.6 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 8.0 C

iteration# =  1.0     error = 2.937e+00
iteration# =  2.0     error = 1.560e+01
iteration# =  3.0     error = 1.648e+01
iteration# =  4.0     error = 2.297e+01
iteration# =  5.0     error = 4.245e+01
iteration# =  6.0     error = 1.808e+00
iteration# =  7.0     error = 5.004e-01
iteration# =  8.0     error = 1.441e-01
iteration# =  9.0     error = 3.430e-02
iteration# =  10.0     error = 1.198e-01
iteration# =  11.0     error = 1.502e-01
iteration# =  12.0     error = 1.369e-01
iteration# =  13.0     error = 9.896e-02
iteration# =  14.0     error = 5.673e-02
iteration# =  15.0     error = 2.367e-02
iteration# =  16.0     error = 3.661e-03
iteration# =  17.0     error = 5.725e-03
iteration# =  18.0     error = 8.315e-03
iteration# =  19.0     error = 7.541e-03
iteration# =  20.0     error = 5.579e-03
iteration# =  21.0     error = 3.574e-03
iteration# =  22.0     error = 1.995e-03
iteration# =  23.0     error = 9.331e-04
iteration# =  24.0     error = 3.099e-04
iteration# =  25.0     error = 6.509e-05

Circuit Number:  2.0
Results
```

```
p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.9 220.9 227.4 234.3 241.7 249.6 258.0 266.9 276.3 286.2
 296.6 307.4 318.7 330.6 343.0 355.8 369.2 383.1 397.5 412.3 427.6 443.5] kJ/kg
T_p =  [3.0 6.8 10.9 15.3 19.9 24.8 30.0 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 44.5 59.9] C
s_p =  [1.014 1.032 1.052 1.073 1.095 1.118 1.143 1.169 1.196 1.225 1.256 1.288
 1.322 1.357 1.394 1.432 1.473 1.514 1.558 1.603 1.649 1.698 1.747 1.796] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.008 0.057 0.110 0.166 0.224
 0.286 0.350 0.417 0.488 0.561 0.637 0.716 0.799 0.884 0.972 1.000 1.000]
deltaP_p_tot =  36.0 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [21.723 21.744 21.591 21.615 21.466 21.494 21.350 21.382 21.243 21.278
 21.144 21.183 21.054 21.097 20.972 21.019 20.899 20.950 20.834 20.889
 20.778 20.836 20.729 20.792] kJ/kg
T_s =  [8.1 8.1 7.9 8.0 7.8 7.8 7.7 7.7 7.6 7.6 7.5 7.5 7.4 7.5 7.3 7.4 7.3 7.3
 7.2 7.3 7.1 7.2 7.1 7.2] C
s_s =  [0.081 0.081 0.080 0.080 0.080 0.080 0.079 0.080 0.079 0.079 0.079 0.079
 0.078 0.079 0.078 0.078 0.078 0.078 0.078 0.078 0.077 0.078 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.211 kW
Q_dot_s_tot = -0.211 kW
Q_dot_sen_tot =  0.211 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.410 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 443.5 kJ/kg
T_r_i = 3.0 C
T_r_e = 59.9 C

m_dot_s =  [[[0.223 0.221 0.221 0.223]
  [0.223 0.221 0.221 0.223]]

 [[0.258 0.256 0.256 0.258]
  [0.258 0.256 0.256 0.258]]

 [[0.223 0.221 0.221 0.223]
  [0.223 0.221 0.221 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.8 kJ/kg dry air
h_a_e = 21.7 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.2 C
T_a_e = 8.1 C

iteration# =  1.0     error = 2.939e+00
iteration# =  2.0     error = 2.961e+01
iteration# =  3.0     error = 1.074e+01
iteration# =  4.0     error = 1.583e+02
iteration# =  5.0     error = 2.476e+01
iteration# =  6.0     error = 1.688e+00
iteration# =  7.0     error = 4.817e-01
iteration# =  8.0     error = 1.372e-01
iteration# =  9.0     error = 3.670e-02
iteration# =  10.0     error = 1.201e-01
iteration# =  11.0     error = 1.489e-01
iteration# =  12.0     error = 1.349e-01
iteration# =  13.0     error = 9.699e-02
iteration# =  14.0     error = 5.526e-02
iteration# =  15.0     error = 2.281e-02
iteration# =  16.0     error = 3.505e-03
iteration# =  17.0     error = 5.816e-03
iteration# =  18.0     error = 8.271e-03
iteration# =  19.0     error = 7.455e-03
```

```
iteration# =  20.0     error = 5.495e-03
iteration# =  21.0     error = 3.510e-03
iteration# =  22.0     error = 1.953e-03
iteration# =  23.0     error = 9.088e-04
iteration# =  24.0     error = 2.978e-04
iteration# =  25.0     error = 6.345e-05

Circuit Number:  3.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.9 220.8 227.3 234.2 241.6 249.5 257.9 266.7 276.1 286.0
 296.3 307.2 318.5 330.3 342.7 355.5 368.8 382.7 397.1 411.9 427.2 443.1] kJ/kg
T_p =  [3.0 6.8 10.9 15.2 19.9 24.7 29.9 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 44.1 59.4] C
s_p =  [1.014 1.032 1.052 1.073 1.095 1.118 1.143 1.169 1.196 1.225 1.255 1.287
 1.321 1.356 1.393 1.431 1.472 1.513 1.556 1.601 1.648 1.696 1.746 1.795] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.007 0.057 0.109 0.164 0.223
 0.285 0.349 0.416 0.486 0.560 0.635 0.714 0.797 0.882 0.970 1.000 1.000]
deltaP_p_tot =  35.9 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [21.616 21.637 21.494 21.518 21.379 21.407 21.273 21.304 21.176 21.211
 21.086 21.125 21.005 21.048 20.933 20.980 20.869 20.920 20.813 20.868
 20.766 20.824 20.726 20.789] kJ/kg
T_s =  [8.0 8.0 7.8 7.9 7.7 7.8 7.6 7.7 7.5 7.6 7.4 7.5 7.4 7.4 7.3 7.3 7.2 7.3
 7.2 7.2 7.1 7.2 7.1 7.2] C
s_s =  [0.080 0.080 0.080 0.080 0.080 0.080 0.079 0.079 0.079 0.079 0.078 0.079
 0.078 0.078 0.078 0.078 0.078 0.078 0.077 0.078 0.077 0.078 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.210 kW
Q_dot_s_tot = -0.210 kW
Q_dot_sen_tot =  0.210 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.412 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 443.1 kJ/kg
T_r_i = 3.0 C
T_r_e = 59.4 C

m_dot_s =  [[[0.223 0.221 0.221 0.223]
  [0.223 0.221 0.221 0.223]]

 [[0.258 0.256 0.256 0.258]
  [0.258 0.256 0.256 0.257]]

 [[0.223 0.221 0.221 0.223]
  [0.223 0.221 0.221 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.8 kJ/kg dry air
h_a_e = 21.6 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.2 C
T_a_e = 8.0 C


iteration# =  1.0     error = 2.940e+00
iteration# =  2.0     error = 4.743e+01
iteration# =  3.0     error = 1.030e+01
iteration# =  4.0     error = 2.046e+01
iteration# =  5.0     error = 1.508e+02
iteration# =  6.0     error = 1.536e+00
iteration# =  7.0     error = 4.559e-01
iteration# =  8.0     error = 1.276e-01
```

```
iteration# =  9.0     error = 4.013e-02
iteration# =  10.0     error = 1.205e-01
iteration# =  11.0     error = 1.472e-01
iteration# =  12.0     error = 1.321e-01
iteration# =  13.0     error = 9.417e-02
iteration# =  14.0     error = 5.316e-02
iteration# =  15.0     error = 2.159e-02
iteration# =  16.0     error = 3.284e-03
iteration# =  17.0     error = 5.943e-03
iteration# =  18.0     error = 8.205e-03
iteration# =  19.0     error = 7.328e-03
iteration# =  20.0     error = 5.373e-03
iteration# =  21.0     error = 3.417e-03
iteration# =  22.0     error = 1.892e-03
iteration# =  23.0     error = 8.735e-04
iteration# =  24.0     error = 2.803e-04
iteration# =  25.0     error = 6.096e-05

Circuit Number:  4.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.8 220.8 227.2 234.1 241.4 249.3 257.7 266.5 275.8 285.7
 296.0 306.8 318.1 329.9 342.3 355.0 368.3 382.2 396.5 411.3 426.6 442.5] kJ/kg
T_p =  [3.0 6.8 10.9 15.2 19.8 24.7 29.8 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 43.5 58.8] C
s_p =  [1.014 1.032 1.052 1.072 1.094 1.118 1.142 1.168 1.195 1.224 1.254 1.286
 1.320 1.355 1.392 1.430 1.470 1.512 1.555 1.600 1.646 1.694 1.744 1.793] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.006 0.055 0.108 0.163 0.221
 0.283 0.347 0.414 0.484 0.557 0.633 0.711 0.794 0.879 0.966 1.000 1.000]
deltaP_p_tot =  35.8 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [21.470 21.490 21.361 21.384 21.259 21.286 21.166 21.197 21.081 21.116
 21.004 21.043 20.936 20.979 20.876 20.922 20.824 20.874 20.780 20.835
 20.745 20.803 20.718 20.780] kJ/kg
T_s =  [7.8 7.8 7.7 7.7 7.6 7.6 7.5 7.6 7.4 7.5 7.4 7.4 7.3 7.3 7.2 7.3 7.2 7.2
 7.1 7.2 7.1 7.2 7.1 7.1] C
s_s =  [0.080 0.080 0.079 0.080 0.079 0.079 0.079 0.079 0.078 0.079 0.078 0.078
 0.078 0.078 0.078 0.078 0.078 0.077 0.078 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.210 kW
Q_dot_s_tot = -0.210 kW
Q_dot_sen_tot =  0.210 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.413 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 442.5 kJ/kg
T_r_i = 3.0 C
T_r_e = 58.8 C

m_dot_s =  [[[0.223 0.221 0.221 0.223]
  [0.223 0.221 0.221 0.223]]

 [[0.257 0.256 0.256 0.257]
  [0.257 0.256 0.256 0.257]]

 [[0.223 0.221 0.221 0.223]
  [0.223 0.221 0.221 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.8 kJ/kg dry air
h_a_e = 21.5 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
```

```
T_a_e = 7.8 C

iteration# =  1.0     error = 2.940e+00
iteration# =  2.0     error = 1.077e+01
iteration# =  3.0     error = 5.596e+01
iteration# =  4.0     error = 3.760e+01
iteration# =  5.0     error = 4.323e+01
iteration# =  6.0     error = 1.365e+00
iteration# =  7.0     error = 4.236e-01
iteration# =  8.0     error = 1.151e-01
iteration# =  9.0     error = 4.464e-02
iteration# =  10.0     error = 1.211e-01
iteration# =  11.0     error = 1.449e-01
iteration# =  12.0     error = 1.284e-01
iteration# =  13.0     error = 9.050e-02
iteration# =  14.0     error = 5.044e-02
iteration# =  15.0     error = 2.001e-02
iteration# =  16.0     error = 3.027e-03
iteration# =  17.0     error = 6.106e-03
iteration# =  18.0     error = 8.119e-03
iteration# =  19.0     error = 7.163e-03
iteration# =  20.0     error = 5.214e-03
iteration# =  21.0     error = 3.296e-03
iteration# =  22.0     error = 1.812e-03
iteration# =  23.0     error = 8.275e-04
iteration# =  24.0     error = 2.575e-04
iteration# =  25.0     error = 5.760e-05

Circuit Number:  5.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.8 220.7 227.1 233.9 241.2 249.1 257.4 266.2 275.5 285.3
 295.6 306.3 317.6 329.4 341.7 354.4 367.7 381.5 395.8 410.6 425.8 441.7] kJ/kg
T_p =  [3.0 6.8 10.8 15.1 19.7 24.5 29.7 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 42.8 58.0] C
s_p =  [1.014 1.032 1.052 1.072 1.094 1.117 1.142 1.167 1.194 1.223 1.253 1.285
 1.318 1.353 1.390 1.428 1.468 1.510 1.553 1.598 1.644 1.692 1.741 1.790] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.004 0.054 0.106 0.161 0.219
 0.280 0.344 0.411 0.481 0.554 0.629 0.708 0.790 0.874 0.962 1.000 1.000]
deltaP_p_tot =  35.7 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [21.284 21.304 21.192 21.215 21.106 21.133 21.029 21.060 20.960 20.994
 20.898 20.937 20.846 20.888 20.801 20.847 20.764 20.815 20.736 20.790
 20.715 20.774 20.703 20.765] kJ/kg
T_s =  [7.6 7.7 7.5 7.6 7.5 7.5 7.4 7.4 7.3 7.4 7.3 7.3 7.2 7.3 7.2 7.2 7.1 7.2
 7.1 7.2 7.1 7.1 7.1 7.1] C
s_s =  [0.079 0.079 0.079 0.079 0.079 0.079 0.078 0.078 0.078 0.078 0.078 0.078
 0.078 0.078 0.077 0.078 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.209 kW
Q_dot_s_tot = -0.209 kW
Q_dot_sen_tot =  0.209 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.413 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 441.7 kJ/kg
T_r_i = 3.0 C
T_r_e = 58.0 C

m_dot_s =  [[[0.223 0.221 0.221 0.223]
  [0.223 0.222 0.222 0.223]]

 [[0.257 0.256 0.256 0.257]
  [0.257 0.256 0.256 0.257]]
```

```
 [[0.223 0.221 0.221 0.223]
  [0.223 0.222 0.222 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.8 kJ/kg dry air
h_a_e = 21.3 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 7.6 C

iteration# =  1.0     error = 2.940e+00
iteration# =  2.0     error = 6.021e+00
iteration# =  3.0     error = 1.313e+01
iteration# =  4.0     error = 1.448e+01
iteration# =  5.0     error = 4.828e+01
iteration# =  6.0     error = 1.188e+00
iteration# =  7.0     error = 3.856e-01
iteration# =  8.0     error = 1.001e-01
iteration# =  9.0     error = 5.019e-02
iteration# =  10.0     error = 1.217e-01
iteration# =  11.0     error = 1.421e-01
iteration# =  12.0     error = 1.239e-01
iteration# =  13.0     error = 8.612e-02
iteration# =  14.0     error = 4.721e-02
iteration# =  15.0     error = 1.814e-02
iteration# =  16.0     error = 2.748e-03
iteration# =  17.0     error = 6.298e-03
iteration# =  18.0     error = 8.016e-03
iteration# =  19.0     error = 6.968e-03
iteration# =  20.0     error = 5.026e-03
iteration# =  21.0     error = 3.153e-03
iteration# =  22.0     error = 1.718e-03
iteration# =  23.0     error = 7.731e-04
iteration# =  24.0     error = 2.305e-04
iteration# =  25.0     error = 5.368e-05

Circuit Number:  6.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.8 220.6 226.9 233.7 241.0 248.8 257.1 265.8 275.1 284.8
 295.1 305.8 317.0 328.8 341.0 353.7 366.9 380.7 395.0 409.7 424.9 440.7] kJ/kg
T_p =  [3.0 6.8 10.8 15.1 19.6 24.4 29.5 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6 41.9 57.1] C
s_p =  [1.014 1.032 1.051 1.072 1.094 1.117 1.141 1.166 1.193 1.222 1.252 1.283
 1.317 1.352 1.388 1.426 1.466 1.507 1.550 1.595 1.641 1.689 1.738 1.787] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.003 0.052 0.104 0.158 0.216
 0.277 0.341 0.407 0.477 0.550 0.625 0.703 0.785 0.869 0.957 1.000 1.000]
deltaP_p_tot =  35.5 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [21.064 21.084 20.991 21.014 20.924 20.951 20.866 20.896 20.815 20.850
 20.772 20.811 20.738 20.780 20.711 20.758 20.693 20.743 20.682 20.736
 20.680 20.738 20.685 20.747] kJ/kg
T_s =  [7.4 7.4 7.4 7.4 7.3 7.3 7.2 7.3 7.2 7.2 7.1 7.2 7.1 7.1 7.1 7.1 7.1 7.1
 7.0 7.1 7.0 7.1 7.0 7.1] C
s_s =  [0.078 0.078 0.078 0.078 0.078 0.078 0.078 0.078 0.078 0.078 0.077 0.077
 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.208 kW
Q_dot_s_tot = -0.208 kW
Q_dot_sen_tot =  0.208 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.413 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
```

```
h_r_i = 204.2 kJ/kg
h_r_e = 440.7 kJ/kg
T_r_i = 3.0 C
T_r_e = 57.1 C

m_dot_s =  [[[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]

 [[0.257 0.255 0.255 0.257]
  [0.256 0.255 0.255 0.256]]

 [[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 21.1 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 7.4 C


iteration# =  1.0     error = 2.941e+00
iteration# =  2.0     error = 4.441e+00
iteration# =  3.0     error = 1.930e+01
iteration# =  4.0     error = 7.569e+01
iteration# =  5.0     error = 2.603e+01
iteration# =  6.0     error = 1.056e+00
iteration# =  7.0     error = 3.541e-01
iteration# =  8.0     error = 8.718e-02
iteration# =  9.0     error = 5.505e-02
iteration# =  10.0     error = 1.223e-01
iteration# =  11.0     error = 1.397e-01
iteration# =  12.0     error = 1.201e-01
iteration# =  13.0     error = 8.234e-02
iteration# =  14.0     error = 4.444e-02
iteration# =  15.0     error = 1.655e-02
iteration# =  16.0     error = 2.520e-03
iteration# =  17.0     error = 6.457e-03
iteration# =  18.0     error = 7.925e-03
iteration# =  19.0     error = 6.799e-03
iteration# =  20.0     error = 4.864e-03
iteration# =  21.0     error = 3.029e-03
iteration# =  22.0     error = 1.637e-03
iteration# =  23.0     error = 7.264e-04
iteration# =  24.0     error = 2.074e-04
iteration# =  25.0     error = 5.076e-05

Circuit Number:  7.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.7 220.5 226.8 233.6 240.8 248.6 256.8 265.5 274.7 284.5
 294.7 305.4 316.6 328.3 340.5 353.2 366.4 380.1 394.3 409.0 424.2 439.9] kJ/kg
T_p =  [3.0 6.8 10.8 15.0 19.5 24.3 29.4 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 41.2 56.4] C
s_p =  [1.014 1.032 1.051 1.072 1.093 1.116 1.140 1.166 1.192 1.221 1.251 1.282
 1.315 1.350 1.387 1.425 1.464 1.506 1.548 1.593 1.639 1.687 1.736 1.785] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.050 0.102 0.156 0.214
 0.275 0.338 0.404 0.474 0.546 0.622 0.700 0.781 0.866 0.953 1.000 1.000]
deltaP_p_tot =  35.4 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.879 20.899 20.822 20.844 20.771 20.797 20.728 20.758 20.692 20.727
 20.665 20.703 20.646 20.688 20.634 20.680 20.631 20.681 20.635 20.689
 20.647 20.705 20.667 20.729] kJ/kg
T_s =  [7.2 7.3 7.2 7.2 7.1 7.2 7.1 7.1 7.1 7.1 7.0 7.1 7.0 7.1 7.0 7.0 7.0 7.0
 7.0 7.1 7.0 7.1 7.0 7.1] C
s_s =  [0.078 0.078 0.078 0.078 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077
 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077] kJ/kgK
```

```
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.207 kW
Q_dot_s_tot = -0.207 kW
Q_dot_sen_tot =  0.207 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.414 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 439.9 kJ/kg
T_r_i = 3.0 C
T_r_e = 56.4 C

m_dot_s =  [[[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]

 [[0.256 0.255 0.255 0.256]
  [0.256 0.255 0.255 0.256]]

 [[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.9 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 7.2 C

iteration# =  1.0     error = 2.942e+00
iteration# =  2.0     error = 7.557e+00
iteration# =  3.0     error = 2.634e+01
iteration# =  4.0     error = 2.974e+01
iteration# =  5.0     error = 2.267e+01
iteration# =  6.0     error = 9.417e-01
iteration# =  7.0     error = 3.239e-01
iteration# =  8.0     error = 7.449e-02
iteration# =  9.0     error = 5.992e-02
iteration# =  10.0     error = 1.229e-01
iteration# =  11.0     error = 1.373e-01
iteration# =  12.0     error = 1.163e-01
iteration# =  13.0     error = 7.872e-02
iteration# =  14.0     error = 4.180e-02
iteration# =  15.0     error = 1.503e-02
iteration# =  16.0     error = 2.335e-03
iteration# =  17.0     error = 6.615e-03
iteration# =  18.0     error = 7.844e-03
iteration# =  19.0     error = 6.641e-03
iteration# =  20.0     error = 4.713e-03
iteration# =  21.0     error = 2.913e-03
iteration# =  22.0     error = 1.560e-03
iteration# =  23.0     error = 6.822e-04
iteration# =  24.0     error = 1.854e-04
iteration# =  25.0     error = 5.157e-05

Circuit Number:  8.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.7 220.5 226.7 233.4 240.6 248.4 256.6 265.3 274.4 284.1
 294.3 305.0 316.1 327.8 340.0 352.6 365.8 379.5 393.6 408.3 423.5 439.2] kJ/kg
T_p =  [3.0 6.8 10.8 15.0 19.4 24.2 29.3 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 40.6 55.6] C
s_p =  [1.014 1.032 1.051 1.071 1.093 1.116 1.140 1.165 1.192 1.220 1.250 1.281
 1.314 1.349 1.385 1.423 1.463 1.504 1.546 1.591 1.637 1.685 1.734 1.783] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.049 0.100 0.155 0.212
 0.272 0.336 0.402 0.471 0.543 0.618 0.696 0.777 0.862 0.949 1.000 1.000]
deltaP_p_tot =  35.3 Pa
```

```
p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.699 20.719 20.657 20.680 20.621 20.648 20.593 20.624 20.574 20.608
 20.561 20.599 20.557 20.599 20.560 20.606 20.572 20.621 20.591 20.645
 20.618 20.676 20.652 20.714] kJ/kg
T_s =  [7.1 7.1 7.0 7.0 7.0 7.0 7.0 7.0 6.9 7.0 6.9 7.0 6.9 7.0 6.9 7.0 6.9 7.0
 7.0 7.0 7.0 7.0 7.0 7.1] C
s_s =  [0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077
 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.207 kW
Q_dot_s_tot = -0.207 kW
Q_dot_sen_tot =  0.207 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.415 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 439.2 kJ/kg
T_r_i = 3.0 C
T_r_e = 55.6 C

m_dot_s =  [[[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]

 [[0.256 0.255 0.255 0.256]
  [0.255 0.255 0.255 0.255]]

 [[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.7 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 7.1 C

iteration# =  1.0     error = 2.944e+00
iteration# =  2.0     error = 2.068e+01
iteration# =  3.0     error = 1.160e+01
iteration# =  4.0     error = 2.356e+01
iteration# =  5.0     error = 8.630e+00
iteration# =  6.0     error = 8.588e-01
iteration# =  7.0     error = 3.003e-01
iteration# =  8.0     error = 6.430e-02
iteration# =  9.0     error = 6.388e-02
iteration# =  10.0     error = 1.234e-01
iteration# =  11.0     error = 1.354e-01
iteration# =  12.0     error = 1.133e-01
iteration# =  13.0     error = 7.581e-02

iteration# =  14.0     error = 3.969e-02
iteration# =  15.0     error = 1.382e-02
iteration# =  16.0     error = 2.188e-03
iteration# =  17.0     error = 6.734e-03
iteration# =  18.0     error = 7.775e-03
iteration# =  19.0     error = 6.514e-03
iteration# =  20.0     error = 4.591e-03
iteration# =  21.0     error = 2.821e-03
iteration# =  22.0     error = 1.500e-03
iteration# =  23.0     error = 6.473e-04
iteration# =  24.0     error = 1.682e-04
iteration# =  25.0     error = 5.815e-05

Circuit Number:  9.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
```

```
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.7 220.4 226.6 233.3 240.5 248.2 256.4 265.0 274.2 283.8
 294.0 304.6 315.8 327.4 339.6 352.2 365.3 379.0 393.2 407.8 423.0 438.6] kJ/kg
T_p =  [3.0 6.8 10.7 14.9 19.4 24.1 29.2 34.5 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 40.1 55.1] C
s_p =  [1.014 1.032 1.051 1.071 1.093 1.115 1.139 1.164 1.191 1.219 1.249 1.280
 1.313 1.348 1.384 1.422 1.461 1.502 1.545 1.589 1.636 1.683 1.732 1.781] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.048 0.099 0.153 0.210
 0.271 0.334 0.400 0.469 0.541 0.616 0.694 0.775 0.859 0.946 1.000 1.000]
deltaP_p_tot =  38.3 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.556 20.576 20.526 20.549 20.503 20.529 20.487 20.517 20.479 20.513
 20.479 20.517 20.486 20.528 20.501 20.547 20.524 20.574 20.555 20.609
 20.594 20.652 20.641 20.702] kJ/kg
T_s =  [6.9 6.9 6.9 6.9 6.9 6.9 6.9 6.9 6.8 6.9 6.8 6.9 6.9 6.9 6.9 6.9 6.9 6.9
 6.9 7.0 7.0 7.0 7.0 7.1] C
s_s =  [0.077 0.077 0.076 0.077 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076
 0.076 0.076 0.076 0.077 0.076 0.077 0.077 0.077 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.206 kW
Q_dot_s_tot = -0.206 kW
Q_dot_sen_tot =  0.206 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.417 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 438.6 kJ/kg
T_r_i = 3.0 C
T_r_e = 55.1 C

m_dot_s =  [[[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]

 [[0.255 0.254 0.254 0.255]
  [0.255 0.254 0.254 0.255]]

 [[0.223 0.222 0.222 0.223]
  [0.223 0.222 0.222 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.6 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 6.9 C

iteration# =  1.0     error = 2.947e+00
iteration# =  2.0     error = 6.450e+01
iteration# =  3.0     error = 4.823e+01
iteration# =  4.0     error = 3.852e+01
iteration# =  5.0     error = 5.696e+00
iteration# =  6.0     error = 7.990e-01
iteration# =  7.0     error = 2.821e-01
iteration# =  8.0     error = 5.631e-02
iteration# =  9.0     error = 6.700e-02
iteration# =  10.0     error = 1.237e-01
iteration# =  11.0     error = 1.338e-01
iteration# =  12.0     error = 1.109e-01
iteration# =  13.0     error = 7.349e-02
iteration# =  14.0     error = 3.802e-02
iteration# =  15.0     error = 1.288e-02
iteration# =  16.0     error = 2.087e-03
iteration# =  17.0     error = 6.821e-03
iteration# =  18.0     error = 7.715e-03
iteration# =  19.0     error = 6.409e-03
iteration# =  20.0     error = 4.492e-03
```

```
iteration# =  21.0     error = 2.746e-03
iteration# =  22.0     error = 1.451e-03
iteration# =  23.0     error = 6.195e-04
iteration# =  24.0     error = 1.546e-04
iteration# =  25.0     error = 6.323e-05

Circuit Number:  10.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.7 220.4 226.6 233.3 240.4 248.1 256.2 264.9 274.0 283.7
 293.8 304.4 315.5 327.2 339.3 351.9 365.0 378.7 392.8 407.5 422.6 438.3] kJ/kg
T_p =  [3.0 6.8 10.7 14.9 19.3 24.1 29.1 34.4 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 39.7 54.7] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.115 1.139 1.164 1.191 1.219 1.248 1.280
 1.313 1.347 1.383 1.421 1.460 1.501 1.544 1.588 1.634 1.682 1.731 1.780] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.047 0.098 0.152 0.209
 0.269 0.332 0.398 0.467 0.539 0.614 0.692 0.773 0.857 0.943 1.000 1.000]
deltaP_p_tot =  38.2 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.446 20.466 20.426 20.448 20.411 20.438 20.404 20.435 20.405 20.439
 20.414 20.452 20.430 20.472 20.454 20.500 20.486 20.536 20.526 20.580
 20.573 20.631 20.629 20.690] kJ/kg
T_s =  [6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.8 6.9 6.9 6.9
 6.9 6.9 6.9 7.0 7.0 7.1] C
s_s =  [0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076
 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.206 kW
Q_dot_s_tot = -0.206 kW
Q_dot_sen_tot =  0.206 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.419 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 438.3 kJ/kg
T_r_i = 3.0 C
T_r_e = 54.7 C

m_dot_s =  [[[0.223 0.222 0.222 0.223]
  [0.223 0.223 0.223 0.223]]

 [[0.254 0.254 0.254 0.254]
  [0.254 0.253 0.253 0.254]]

 [[0.223 0.222 0.222 0.223]
  [0.223 0.223 0.223 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.4 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.1 C
T_a_e = 6.8 C

iteration# =  1.0     error = 2.950e+00
iteration# =  2.0     error = 1.551e+01
iteration# =  3.0     error = 2.521e+01
iteration# =  4.0     error = 2.769e+01
iteration# =  5.0     error = 4.225e+00
iteration# =  6.0     error = 7.448e-01
iteration# =  7.0     error = 2.649e-01
iteration# =  8.0     error = 4.863e-02
iteration# =  9.0     error = 7.004e-02
iteration# =  10.0     error = 1.240e-01
```

```
iteration# =  11.0     error = 1.323e-01
iteration# =  12.0     error = 1.086e-01
iteration# =  13.0     error = 7.131e-02
iteration# =  14.0     error = 3.646e-02
iteration# =  15.0     error = 1.199e-02
iteration# =  16.0     error = 1.996e-03
iteration# =  17.0     error = 6.904e-03
iteration# =  18.0     error = 7.660e-03
iteration# =  19.0     error = 6.313e-03
iteration# =  20.0     error = 4.400e-03
iteration# =  21.0     error = 2.677e-03
iteration# =  22.0     error = 1.406e-03
iteration# =  23.0     error = 5.938e-04
iteration# =  24.0     error = 1.440e-04
iteration# =  25.0     error = 6.790e-05

Circuit Number:  11.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.6 220.3 226.5 233.2 240.3 248.0 256.1 264.7 273.9 283.5
 293.6 304.2 315.3 326.9 339.0 351.6 364.7 378.4 392.5 407.1 422.3 437.9] kJ/kg
T_p =  [3.0 6.8 10.7 14.9 19.3 24.0 29.0 34.3 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 39.4 54.4] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.115 1.139 1.164 1.190 1.218 1.248 1.279
 1.312 1.346 1.382 1.420 1.460 1.501 1.543 1.587 1.633 1.681 1.730 1.779] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046 0.097 0.151 0.208
 0.268 0.331 0.397 0.466 0.538 0.613 0.690 0.771 0.855 0.942 1.000 1.000]
deltaP_p_tot =  38.2 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.340 20.360 20.329 20.351 20.323 20.349 20.325 20.355 20.335 20.369
 20.352 20.390 20.378 20.419 20.410 20.456 20.451 20.501 20.500 20.553
 20.556 20.613 20.620 20.682] kJ/kg
T_s =  [6.7 6.7 6.7 6.7 6.7 6.7 6.7 6.7 6.7 6.7 6.8 6.7 6.8 6.8 6.8 6.8 6.9
 6.9 6.9 6.9 7.0 7.0 7.0] C
s_s =  [0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076
 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.206 kW
Q_dot_s_tot = -0.206 kW
Q_dot_sen_tot =  0.206 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.422 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.9 kJ/kg
T_r_i = 3.0 C
T_r_e = 54.4 C

m_dot_s =  [[[0.223 0.223 0.223 0.223]
  [0.223 0.223 0.223 0.223]]

 [[0.254 0.253 0.253 0.254]
  [0.253 0.253 0.253 0.253]]

 [[0.223 0.223 0.223 0.223]
  [0.223 0.223 0.223 0.223]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.3 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.7 C
```

```
iteration# =  1.0     error = 2.954e+00
iteration# =  2.0     error = 9.831e+00
iteration# =  3.0     error = 1.191e+01
iteration# =  4.0     error = 2.276e+01
iteration# =  5.0     error = 3.547e+00
iteration# =  6.0     error = 7.086e-01
iteration# =  7.0     error = 2.529e-01
iteration# =  8.0     error = 4.318e-02
iteration# =  9.0     error = 7.219e-02
iteration# =  10.0     error = 1.242e-01
iteration# =  11.0     error = 1.312e-01
iteration# =  12.0     error = 1.069e-01
iteration# =  13.0     error = 6.971e-02
iteration# =  14.0     error = 3.532e-02
iteration# =  15.0     error = 1.136e-02
iteration# =  16.0     error = 1.927e-03
iteration# =  17.0     error = 6.955e-03
iteration# =  18.0     error = 7.613e-03
iteration# =  19.0     error = 6.238e-03
iteration# =  20.0     error = 4.331e-03
iteration# =  21.0     error = 2.625e-03
iteration# =  22.0     error = 1.372e-03
iteration# =  23.0     error = 5.749e-04
iteration# =  24.0     error = 1.387e-04
iteration# =  25.0     error = 7.111e-05

Circuit Number:  12.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.6 220.3 226.5 233.1 240.3 247.9 256.0 264.7 273.8 283.4
 293.5 304.1 315.2 326.8 338.9 351.5 364.6 378.2 392.3 407.0 422.1 437.7] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.3 24.0 29.0 34.3 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 39.2 54.2] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.115 1.138 1.163 1.190 1.218 1.248 1.279
 1.312 1.346 1.382 1.420 1.459 1.500 1.543 1.587 1.633 1.680 1.729 1.778] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.046 0.097 0.151 0.208
 0.268 0.331 0.396 0.465 0.537 0.612 0.689 0.770 0.854 0.941 1.000 1.000]
deltaP_p_tot =  38.1 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.317 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.267 20.287 20.262 20.284 20.262 20.288 20.270 20.300 20.285 20.319
 20.308 20.346 20.340 20.381 20.378 20.424 20.425 20.474 20.479 20.533
 20.541 20.599 20.611 20.673] kJ/kg
T_s =  [6.6 6.7 6.6 6.7 6.6 6.7 6.6 6.7 6.7 6.7 6.7 6.7 6.7 6.8 6.7 6.8 6.8 6.8
 6.8 6.9 6.9 7.0 7.0 7.0] C
s_s =  [0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076
 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.425 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.7 kJ/kg
T_r_i = 3.0 C
T_r_e = 54.2 C

m_dot_s =  [[[0.224 0.223 0.223 0.224]
  [0.224 0.223 0.223 0.224]]

 [[0.253 0.252 0.252 0.253]
  [0.252 0.252 0.252 0.252]]

 [[0.224 0.223 0.223 0.224]
```

```
  [0.224 0.223 0.223 0.224]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.3 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.6 C

iteration# =  1.0     error = 2.959e+00
iteration# =  2.0     error = 7.345e+00
iteration# =  3.0     error = 2.108e+01
iteration# =  4.0     error = 3.000e+02
iteration# =  5.0     error = 3.038e+00
iteration# =  6.0     error = 6.738e-01
iteration# =  7.0     error = 2.409e-01
iteration# =  8.0     error = 3.773e-02
iteration# =  9.0     error = 7.435e-02
iteration# =  10.0     error = 1.244e-01
iteration# =  11.0     error = 1.300e-01
iteration# =  12.0     error = 1.051e-01
iteration# =  13.0     error = 6.811e-02
iteration# =  14.0     error = 3.419e-02
iteration# =  15.0     error = 1.072e-02
iteration# =  16.0     error = 2.101e-03
iteration# =  17.0     error = 7.004e-03
iteration# =  18.0     error = 7.566e-03
iteration# =  19.0     error = 6.162e-03
iteration# =  20.0     error = 4.261e-03
iteration# =  21.0     error = 2.573e-03
iteration# =  22.0     error = 1.339e-03
iteration# =  23.0     error = 5.562e-04
iteration# =  24.0     error = 1.336e-04
iteration# =  25.0     error = 7.424e-05

Circuit Number:  13.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.6 220.3 226.5 233.1 240.2 247.9 256.0 264.6 273.7 283.3
 293.4 304.0 315.1 326.7 338.8 351.4 364.5 378.1 392.2 406.8 421.9 437.6] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.3 24.0 29.0 34.2 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 39.1 54.0] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.190 1.218 1.247 1.278
 1.311 1.346 1.382 1.419 1.459 1.500 1.542 1.586 1.632 1.680 1.729 1.778] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.045 0.096 0.150 0.207
 0.267 0.330 0.396 0.464 0.536 0.611 0.689 0.769 0.853 0.940 1.000 1.000]
deltaP_p_tot =  38.1 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.194 20.214 20.195 20.217 20.201 20.227 20.214 20.244 20.236 20.270
 20.265 20.303 20.302 20.343 20.346 20.392 20.399 20.448 20.459 20.512
 20.527 20.584 20.602 20.664] kJ/kg
T_s =  [6.6 6.6 6.6 6.6 6.6 6.6 6.6 6.6 6.6 6.6 6.6 6.7 6.7 6.7 6.7 6.8 6.8 6.8
 6.8 6.9 6.9 7.0 7.0 7.0] C
s_s =  [0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.076 0.076 0.076
 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.429 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.6 kJ/kg
```

```
T_r_i = 3.0 C
T_r_e = 54.0 C

m_dot_s =  [[[0.224 0.223 0.223 0.224]
  [0.224 0.223 0.223 0.224]]

 [[0.252 0.252 0.252 0.252]
  [0.252 0.251 0.251 0.252]]

 [[0.224 0.223 0.223 0.224]
  [0.224 0.223 0.223 0.224]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.2 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.6 C

iteration# =  1.0     error = 2.963e+00
iteration# =  2.0     error = 6.240e+00
iteration# =  3.0     error = 3.867e+02
iteration# =  4.0     error = 2.061e+01
iteration# =  5.0     error = 2.660e+00
iteration# =  6.0     error = 6.422e-01
iteration# =  7.0     error = 2.298e-01
iteration# =  8.0     error = 3.507e-02
iteration# =  9.0     error = 7.640e-02
iteration# =  10.0     error = 1.246e-01
iteration# =  11.0     error = 1.289e-01
iteration# =  12.0     error = 1.036e-01
iteration# =  13.0     error = 6.665e-02
iteration# =  14.0     error = 3.316e-02
iteration# =  15.0     error = 1.015e-02
iteration# =  16.0     error = 2.319e-03
iteration# =  17.0     error = 7.052e-03
iteration# =  18.0     error = 7.525e-03
iteration# =  19.0     error = 6.096e-03
iteration# =  20.0     error = 4.200e-03
iteration# =  21.0     error = 2.527e-03
iteration# =  22.0     error = 1.310e-03
iteration# =  23.0     error = 5.396e-04
iteration# =  24.0     error = 1.298e-04
iteration# =  25.0     error = 7.706e-05

Circuit Number:  14.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.6 220.3 226.4 233.1 240.2 247.8 255.9 264.5 273.6 283.2
 293.3 303.9 315.0 326.6 338.6 351.2 364.3 377.9 392.0 406.7 421.8 437.4] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.2 24.0 28.9 34.2 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 39.0 53.9] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.247 1.278
 1.311 1.345 1.381 1.419 1.458 1.499 1.542 1.586 1.632 1.679 1.728 1.777] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.045 0.096 0.150 0.207
 0.267 0.329 0.395 0.464 0.535 0.610 0.688 0.768 0.852 0.939 1.000 1.000]
deltaP_p_tot =  38.1 Pa

p_s =  [101.307 101.308 101.309 101.310 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.124 20.145 20.131 20.153 20.143 20.169 20.162 20.192 20.190 20.223
 20.224 20.262 20.267 20.308 20.317 20.363 20.375 20.425 20.441 20.495
 20.515 20.572 20.596 20.658] kJ/kg
T_s =  [6.5 6.5 6.5 6.5 6.5 6.5 6.5 6.6 6.6 6.6 6.6 6.6 6.6 6.7 6.7 6.7 6.7 6.8
 6.8 6.9 6.9 6.9 7.0 7.0] C
s_s =  [0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.076
 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
```

```
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.433 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.4 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.9 C

m_dot_s =  [[[0.224 0.224 0.224 0.224]
  [0.224 0.224 0.224 0.224]]

 [[0.251 0.251 0.251 0.251]
  [0.251 0.250 0.250 0.251]]

 [[0.224 0.224 0.224 0.224]
  [0.224 0.224 0.224 0.224]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.1 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.5 C


iteration# =  1.0     error = 2.968e+00
iteration# =  2.0     error = 8.253e+00
iteration# =  3.0     error = 2.670e+01
iteration# =  4.0     error = 3.399e+01
iteration# =  5.0     error = 2.370e+00
iteration# =  6.0     error = 6.134e-01
iteration# =  7.0     error = 2.195e-01
iteration# =  8.0     error = 3.498e-02
iteration# =  9.0     error = 7.835e-02
iteration# =  10.0     error = 1.248e-01
iteration# =  11.0     error = 1.280e-01
iteration# =  12.0     error = 1.022e-01
iteration# =  13.0     error = 6.533e-02
iteration# =  14.0     error = 3.223e-02
iteration# =  15.0     error = 9.628e-03
iteration# =  16.0     error = 2.519e-03
iteration# =  17.0     error = 7.097e-03
iteration# =  18.0     error = 7.491e-03
iteration# =  19.0     error = 6.038e-03
iteration# =  20.0     error = 4.146e-03
iteration# =  21.0     error = 2.487e-03
iteration# =  22.0     error = 1.284e-03
iteration# =  23.0     error = 5.249e-04
iteration# =  24.0     error = 1.265e-04
iteration# =  25.0     error = 7.957e-05

Circuit Number:  15.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.6 220.3 226.4 233.0 240.1 247.7 255.8 264.4 273.5 283.1
 293.2 303.8 314.8 326.4 338.5 351.1 364.2 377.8 391.9 406.5 421.6 437.3] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.2 23.9 28.9 34.1 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.8 53.7] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.247 1.278
 1.311 1.345 1.381 1.419 1.458 1.499 1.541 1.586 1.631 1.679 1.728 1.777] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.044 0.095 0.149 0.206
 0.266 0.329 0.394 0.463 0.535 0.609 0.687 0.768 0.851 0.938 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
```

```
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.058 20.079 20.071 20.093 20.088 20.114 20.114 20.143 20.146 20.180
 20.187 20.225 20.235 20.277 20.291 20.337 20.355 20.405 20.427 20.480
 20.506 20.563 20.593 20.655] kJ/kg
T_s =  [6.4 6.5 6.4 6.5 6.5 6.5 6.5 6.5 6.5 6.6 6.6 6.6 6.6 6.6 6.7 6.7 6.7 6.8
 6.8 6.8 6.9 6.9 7.0 7.0] C
s_s =  [0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075
 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.437 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.3 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.7 C

m_dot_s =  [[[0.224 0.224 0.224 0.224]
  [0.224 0.224 0.224 0.224]]

 [[0.250 0.250 0.250 0.250]
  [0.250 0.249 0.249 0.250]]

 [[0.224 0.224 0.224 0.224]
  [0.224 0.224 0.224 0.224]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.1 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.4 C

iteration# =  1.0     error = 2.973e+00
iteration# =  2.0     error = 1.022e+01
iteration# =  3.0     error = 1.676e+01
iteration# =  4.0     error = 1.805e+02
iteration# =  5.0     error = 2.237e+00
iteration# =  6.0     error = 5.986e-01
iteration# =  7.0     error = 2.140e-01
iteration# =  8.0     error = 3.489e-02
iteration# =  9.0     error = 7.934e-02
iteration# =  10.0     error = 1.248e-01
iteration# =  11.0     error = 1.274e-01
iteration# =  12.0     error = 1.013e-01
iteration# =  13.0     error = 6.458e-02
iteration# =  14.0     error = 3.171e-02
iteration# =  15.0     error = 9.348e-03
iteration# =  16.0     error = 2.619e-03
iteration# =  17.0     error = 7.112e-03
iteration# =  18.0     error = 7.464e-03
iteration# =  19.0     error = 6.001e-03
iteration# =  20.0     error = 4.113e-03
iteration# =  21.0     error = 2.463e-03
iteration# =  22.0     error = 1.269e-03
iteration# =  23.0     error = 5.168e-04
iteration# =  24.0     error = 1.247e-04
iteration# =  25.0     error = 8.066e-05

Circuit Number:  16.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.3 214.6 220.3 226.4 233.0 240.1 247.7 255.8 264.4 273.5 283.1
 293.2 303.8 314.8 326.4 338.5 351.1 364.2 377.8 391.9 406.5 421.6 437.3] kJ/kg
```

```
T_p =  [3.0 6.8 10.7 14.8 19.2 23.9 28.9 34.1 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.8 53.7] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.247 1.278
 1.311 1.345 1.381 1.419 1.458 1.499 1.541 1.586 1.631 1.679 1.728 1.777] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.044 0.095 0.149 0.206
 0.266 0.329 0.394 0.463 0.535 0.609 0.687 0.768 0.851 0.938 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [20.024 20.044 20.039 20.061 20.060 20.086 20.088 20.118 20.124 20.157
 20.167 20.204 20.218 20.260 20.277 20.322 20.344 20.393 20.418 20.471
 20.500 20.558 20.591 20.652] kJ/kg
T_s =  [6.4 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.5 6.5 6.5 6.6 6.6 6.6 6.6 6.7 6.7 6.8
 6.8 6.8 6.9 6.9 7.0 7.0] C
s_s =  [0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075
 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.441 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.3 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.7 C

m_dot_s =  [[[0.224 0.224 0.224 0.224]
  [0.225 0.224 0.224 0.225]]

 [[0.249 0.249 0.249 0.249]
  [0.248 0.248 0.248 0.248]]

 [[0.224 0.224 0.224 0.224]
  [0.225 0.224 0.224 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.7 kJ/kg dry air
h_a_e = 20.0 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.4 C

iteration# =  1.0     error = 2.979e+00
iteration# =  2.0     error = 1.491e+01
iteration# =  3.0     error = 1.260e+01
iteration# =  4.0     error = 4.599e+01
iteration# =  5.0     error = 2.103e+00
iteration# =  6.0     error = 5.825e-01
iteration# =  7.0     error = 2.079e-01
iteration# =  8.0     error = 3.473e-02
iteration# =  9.0     error = 8.047e-02
iteration# =  10.0     error = 1.248e-01
iteration# =  11.0     error = 1.267e-01
iteration# =  12.0     error = 1.004e-01
iteration# =  13.0     error = 6.370e-02
iteration# =  14.0     error = 3.110e-02
iteration# =  15.0     error = 9.019e-03
iteration# =  16.0     error = 2.738e-03
iteration# =  17.0     error = 7.128e-03
iteration# =  18.0     error = 7.431e-03
iteration# =  19.0     error = 5.956e-03
iteration# =  20.0     error = 4.073e-03
iteration# =  21.0     error = 2.434e-03
iteration# =  22.0     error = 1.250e-03
iteration# =  23.0     error = 5.069e-04
```

```
iteration# =  24.0     error = 1.223e-04
iteration# =  25.0     error = 8.204e-05

Circuit Number:  17.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.3 226.4 233.0 240.1 247.7 255.8 264.4 273.5 283.1
 293.2 303.7 314.8 326.4 338.5 351.1 364.2 377.8 391.9 406.5 421.6 437.3] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.2 23.9 28.9 34.1 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.8 53.8] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.247 1.278
 1.311 1.345 1.381 1.419 1.458 1.499 1.541 1.586 1.631 1.679 1.728 1.777] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.044 0.095 0.149 0.206
 0.266 0.328 0.394 0.463 0.535 0.609 0.687 0.768 0.851 0.938 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.987 20.007 20.005 20.027 20.028 20.054 20.059 20.089 20.098 20.131
 20.144 20.182 20.198 20.239 20.260 20.305 20.329 20.379 20.407 20.460
 20.492 20.549 20.585 20.646] kJ/kg
T_s =  [6.4 6.4 6.4 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.5 6.6 6.6 6.6 6.6 6.7 6.7 6.7
 6.8 6.8 6.9 6.9 7.0 7.0] C
s_s =  [0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075
 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.446 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.3 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.8 C

m_dot_s =  [[[0.225 0.224 0.224 0.225]
  [0.225 0.225 0.225 0.225]]

 [[0.248 0.248 0.248 0.248]
  [0.247 0.247 0.247 0.247]]

 [[0.225 0.224 0.224 0.225]
  [0.225 0.225 0.225 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 20.0 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.4 C


iteration# =  1.0     error = 2.983e+00
iteration# =  2.0     error = 2.933e+01
iteration# =  3.0     error = 2.881e+01
iteration# =  4.0     error = 2.018e+01
iteration# =  5.0     error = 1.903e+00
iteration# =  6.0     error = 5.559e-01
iteration# =  7.0     error = 1.978e-01
iteration# =  8.0     error = 3.464e-02
iteration# =  9.0     error = 8.244e-02
iteration# =  10.0     error = 1.250e-01
iteration# =  11.0     error = 1.258e-01
iteration# =  12.0     error = 9.896e-02
```

```
iteration# =  13.0     error = 6.239e-02
iteration# =  14.0     error = 3.019e-02
iteration# =  15.0     error = 8.508e-03
iteration# =  16.0     error = 2.937e-03
iteration# =  17.0     error = 7.173e-03
iteration# =  18.0     error = 7.398e-03
iteration# =  19.0     error = 5.900e-03
iteration# =  20.0     error = 4.021e-03
iteration# =  21.0     error = 2.395e-03
iteration# =  22.0     error = 1.225e-03
iteration# =  23.0     error = 4.925e-04
iteration# =  24.0     error = 1.191e-04
iteration# =  25.0     error = 8.452e-05

Circuit Number:  18.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.4 233.0 240.1 247.7 255.8 264.3 273.4 283.0
 293.1 303.6 314.7 326.3 338.4 351.0 364.0 377.6 391.8 406.4 421.5 437.1] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.2 23.9 28.9 34.1 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.7 53.6] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.246 1.277
 1.310 1.345 1.381 1.418 1.457 1.498 1.541 1.585 1.631 1.678 1.727 1.776] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.044 0.095 0.149 0.205
 0.265 0.328 0.394 0.462 0.534 0.608 0.686 0.767 0.850 0.937 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.922 19.942 19.945 19.967 19.974 20.000 20.011 20.041 20.055 20.089
 20.107 20.144 20.167 20.208 20.234 20.279 20.309 20.358 20.392 20.445
 20.483 20.540 20.582 20.643] kJ/kg
T_s =  [6.3 6.3 6.3 6.3 6.4 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.5 6.6 6.6 6.7 6.7 6.7
 6.8 6.8 6.9 6.9 6.9 7.0] C
s_s =  [0.074 0.074 0.074 0.074 0.075 0.075 0.075 0.075 0.075 0.075 0.075 0.075
 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.450 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.1 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.6 C

m_dot_s =  [[[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]

 [[0.247 0.247 0.247 0.247]
  [0.246 0.246 0.246 0.246]]

 [[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.9 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.3 C

iteration# =  1.0     error = 2.988e+00
iteration# =  2.0     error = 6.939e+01
```

```
iteration# =  3.0     error = 9.758e+01
iteration# =  4.0     error = 4.106e+01
iteration# =  5.0     error = 1.808e+00
iteration# =  6.0     error = 5.421e-01
iteration# =  7.0     error = 1.924e-01
iteration# =  8.0     error = 3.456e-02
iteration# =  9.0     error = 8.346e-02
iteration# =  10.0     error = 1.251e-01
iteration# =  11.0     error = 1.252e-01
iteration# =  12.0     error = 9.815e-02
iteration# =  13.0     error = 6.166e-02
iteration# =  14.0     error = 2.968e-02
iteration# =  15.0     error = 8.232e-03
iteration# =  16.0     error = 3.040e-03
iteration# =  17.0     error = 7.190e-03
iteration# =  18.0     error = 7.373e-03
iteration# =  19.0     error = 5.864e-03
iteration# =  20.0     error = 3.989e-03
iteration# =  21.0     error = 2.372e-03
iteration# =  22.0     error = 1.211e-03
iteration# =  23.0     error = 4.846e-04
iteration# =  24.0     error = 1.172e-04
iteration# =  25.0     error = 8.567e-05

Circuit Number:  19.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.4 233.0 240.1 247.7 255.7 264.3 273.4 283.0
 293.1 303.6 314.7 326.3 338.4 350.9 364.0 377.6 391.7 406.4 421.5 437.1] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.2 23.9 28.9 34.1 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.7 53.6] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.246 1.277
 1.310 1.345 1.381 1.418 1.457 1.498 1.541 1.585 1.631 1.678 1.727 1.776] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.044 0.095 0.149 0.205
 0.265 0.328 0.393 0.462 0.534 0.608 0.686 0.767 0.850 0.937 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.888 19.908 19.914 19.936 19.946 19.972 19.985 20.015 20.032 20.066
 20.087 20.125 20.150 20.191 20.220 20.265 20.298 20.347 20.384 20.437
 20.477 20.534 20.579 20.640] kJ/kg
T_s =  [6.3 6.3 6.3 6.3 6.3 6.3 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.6 6.6 6.6 6.7 6.7
 6.8 6.8 6.8 6.9 6.9 7.0] C
s_s =  [0.074 0.074 0.074 0.074 0.074 0.075 0.075 0.075 0.075 0.075 0.075 0.075
 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.453 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.1 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.6 C

m_dot_s =  [[[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]

 [[0.245 0.245 0.245 0.245]
  [0.245 0.245 0.245 0.245]]

 [[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]] kg/s
P_a_i = 101.3 kPa
```

```
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.9 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.3 C

iteration# =  1.0     error = 2.992e+00
iteration# =  2.0     error = 3.470e+02
iteration# =  3.0     error = 1.041e+02
iteration# =  4.0     error = 4.711e+02
iteration# =  5.0     error = 1.731e+00
iteration# =  6.0     error = 5.302e-01
iteration# =  7.0     error = 1.878e-01
iteration# =  8.0     error = 3.455e-02
iteration# =  9.0     error = 8.437e-02
iteration# =  10.0     error = 1.251e-01
iteration# =  11.0     error = 1.248e-01
iteration# =  12.0     error = 9.750e-02
iteration# =  13.0     error = 6.107e-02
iteration# =  14.0     error = 2.927e-02
iteration# =  15.0     error = 8.007e-03
iteration# =  16.0     error = 3.132e-03
iteration# =  17.0     error = 7.207e-03
iteration# =  18.0     error = 7.356e-03
iteration# =  19.0     error = 5.838e-03
iteration# =  20.0     error = 3.965e-03
iteration# =  21.0     error = 2.354e-03
iteration# =  22.0     error = 1.199e-03
iteration# =  23.0     error = 4.785e-04
iteration# =  24.0     error = 1.159e-04
iteration# =  25.0     error = 8.659e-05

Circuit Number:  20.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.3 233.0 240.0 247.6 255.7 264.3 273.4 282.9
 293.0 303.6 314.7 326.2 338.3 350.9 364.0 377.6 391.7 406.3 421.4 437.1] kJ/kg
T_p =  [3.0 6.8 10.7 14.8 19.2 23.9 28.8 34.1 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.7 53.6] C
s_p =  [1.014 1.032 1.051 1.071 1.092 1.114 1.138 1.163 1.189 1.217 1.246 1.277
 1.310 1.344 1.380 1.418 1.457 1.498 1.541 1.585 1.631 1.678 1.727 1.776] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.044 0.095 0.148 0.205
 0.265 0.328 0.393 0.462 0.534 0.608 0.686 0.766 0.850 0.937 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.857 19.877 19.886 19.908 19.921 19.947 19.963 19.993 20.013 20.046
 20.070 20.108 20.136 20.177 20.209 20.254 20.289 20.339 20.378 20.431
 20.475 20.532 20.579 20.640] kJ/kg
T_s =  [6.2 6.3 6.3 6.3 6.3 6.3 6.3 6.4 6.4 6.4 6.4 6.5 6.5 6.5 6.6 6.6 6.7 6.7
 6.7 6.8 6.8 6.9 6.9 7.0] C
s_s =  [0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.075 0.075 0.075 0.075 0.075
 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.457 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 437.1 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.6 C
```

```
m_dot_s =  [[[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]

 [[0.244 0.244 0.244 0.244]
  [0.243 0.243 0.243 0.243]]

 [[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.9 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.2 C


iteration# =  1.0     error = 2.995e+00
iteration# =  2.0     error = 3.029e+01
iteration# =  3.0     error = 1.737e+01
iteration# =  4.0     error = 2.019e+01
iteration# =  5.0     error = 1.574e+00
iteration# =  6.0     error = 5.040e-01
iteration# =  7.0     error = 1.772e-01
iteration# =  8.0     error = 3.439e-02
iteration# =  9.0     error = 8.651e-02
iteration# =  10.0     error = 1.254e-01
iteration# =  11.0     error = 1.237e-01
iteration# =  12.0     error = 9.598e-02
iteration# =  13.0     error = 5.966e-02
iteration# =  14.0     error = 2.828e-02
iteration# =  15.0     error = 7.452e-03
iteration# =  16.0     error = 3.368e-03
iteration# =  17.0     error = 7.258e-03
iteration# =  18.0     error = 7.322e-03
iteration# =  19.0     error = 5.777e-03
iteration# =  20.0     error = 3.908e-03
iteration# =  21.0     error = 2.311e-03
iteration# =  22.0     error = 1.171e-03
iteration# =  23.0     error = 4.625e-04
iteration# =  24.0     error = 1.121e-04
iteration# =  25.0     error = 8.946e-05

Circuit Number:  21.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.3 232.9 240.0 247.6 255.6 264.2 273.3 282.8
 292.9 303.5 314.5 326.1 338.2 350.8 363.8 377.4 391.5 406.1 421.3 436.9] kJ/kg
T_p =  [3.0 6.9 10.7 14.8 19.2 23.8 28.8 34.0 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.5 53.4] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.114 1.137 1.162 1.189 1.216 1.246 1.277
 1.310 1.344 1.380 1.418 1.457 1.498 1.540 1.584 1.630 1.678 1.727 1.776] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.043 0.094 0.148 0.205
 0.264 0.327 0.393 0.461 0.533 0.607 0.685 0.765 0.849 0.936 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.789 19.810 19.825 19.846 19.865 19.890 19.912 19.942 19.968 20.001
 20.031 20.068 20.102 20.143 20.180 20.225 20.267 20.316 20.361 20.414
 20.463 20.520 20.573 20.634] kJ/kg
T_s =  [6.2 6.2 6.2 6.2 6.2 6.3 6.3 6.3 6.3 6.4 6.4 6.4 6.5 6.5 6.6 6.6 6.6 6.7
 6.7 6.8 6.8 6.9 6.9 7.0] C
s_s =  [0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.075 0.075 0.075
 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
```

```
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.460 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 436.9 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.4 C

m_dot_s =  [[[0.226 0.225 0.225 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.243 0.243 0.243 0.243]
  [0.242 0.242 0.242 0.242]]

 [[0.226 0.225 0.225 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.8 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.2 C
iteration# =  1.0     error = 2.998e+00
iteration# =  2.0     error = 2.305e+01
iteration# =  3.0     error = 1.328e+01
iteration# =  4.0     error = 2.481e+01
iteration# =  5.0     error = 1.511e+00
iteration# =  6.0     error = 4.926e-01
iteration# =  7.0     error = 1.726e-01
iteration# =  8.0     error = 3.438e-02
iteration# =  9.0     error = 8.744e-02
iteration# =  10.0     error = 1.255e-01
iteration# =  11.0     error = 1.233e-01
iteration# =  12.0     error = 9.535e-02
iteration# =  13.0     error = 5.908e-02
iteration# =  14.0     error = 2.787e-02
iteration# =  15.0     error = 7.228e-03
iteration# =  16.0     error = 3.463e-03
iteration# =  17.0     error = 7.278e-03
iteration# =  18.0     error = 7.308e-03
iteration# =  19.0     error = 5.754e-03
iteration# =  20.0     error = 3.886e-03
iteration# =  21.0     error = 2.295e-03
iteration# =  22.0     error = 1.161e-03
iteration# =  23.0     error = 4.564e-04
iteration# =  24.0     error = 1.108e-04
iteration# =  25.0     error = 9.050e-05

Circuit Number:  22.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.3 232.9 240.0 247.5 255.6 264.2 273.2 282.8
 292.9 303.4 314.5 326.1 338.1 350.7 363.8 377.4 391.5 406.1 421.2 436.8] kJ/kg
T_p =  [3.0 6.9 10.7 14.8 19.2 23.8 28.8 34.0 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.4 53.3] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.114 1.137 1.162 1.189 1.216 1.246 1.277
 1.310 1.344 1.380 1.417 1.457 1.498 1.540 1.584 1.630 1.678 1.727 1.776] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.043 0.094 0.148 0.204
 0.264 0.327 0.392 0.461 0.532 0.607 0.685 0.765 0.849 0.935 1.000 1.000]
deltaP_p_tot =  38.0 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.759 19.779 19.797 19.819 19.839 19.865 19.890 19.919 19.948 19.981
 20.014 20.051 20.088 20.129 20.169 20.214 20.258 20.307 20.355 20.408
```

```
 20.460 20.517 20.573 20.634] kJ/kg
T_s =  [6.1 6.2 6.2 6.2 6.2 6.2 6.3 6.3 6.3 6.4 6.4 6.4 6.5 6.5 6.5 6.6 6.6 6.7
 6.7 6.8 6.8 6.9 6.9 7.0] C
s_s =  [0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.075 0.075 0.075
 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.205 kW
Q_dot_s_tot = -0.205 kW
Q_dot_sen_tot =  0.205 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.462 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 436.8 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.3 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.241 0.241 0.241 0.241]
  [0.241 0.241 0.241 0.241]]

 [[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.8 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.1 C

iteration# =  1.0     error = 3.000e+00
iteration# =  2.0     error = 1.436e+01
iteration# =  3.0     error = 3.714e+01
iteration# =  4.0     error = 9.445e+01
iteration# =  5.0     error = 1.388e+00
iteration# =  6.0     error = 4.691e-01
iteration# =  7.0     error = 1.629e-01
iteration# =  8.0     error = 3.429e-02
iteration# =  9.0     error = 8.950e-02
iteration# =  10.0     error = 1.257e-01
iteration# =  11.0     error = 1.224e-01
iteration# =  12.0     error = 9.400e-02
iteration# =  13.0     error = 5.782e-02
iteration# =  14.0     error = 2.698e-02
iteration# =  15.0     error = 6.725e-03
iteration# =  16.0     error = 3.682e-03
iteration# =  17.0     error = 7.331e-03
iteration# =  18.0     error = 7.283e-03
iteration# =  19.0     error = 5.704e-03
iteration# =  20.0     error = 3.837e-03
iteration# =  21.0     error = 2.258e-03
iteration# =  22.0     error = 1.136e-03
iteration# =  23.0     error = 4.424e-04
iteration# =  24.0     error = 1.075e-04
iteration# =  25.0     error = 9.319e-05

Circuit Number:  23.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.3 232.8 239.9 247.5 255.5 264.1 273.1 282.7
 292.7 303.3 314.4 325.9 338.0 350.5 363.6 377.2 391.3 405.9 421.0 436.6] kJ/kg
T_p =  [3.0 6.9 10.7 14.8 19.1 23.8 28.8 34.0 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.2 53.1] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.114 1.137 1.162 1.188 1.216 1.245 1.277
```

```
 1.309 1.343 1.379 1.417 1.456 1.497 1.539 1.584 1.629 1.677 1.726 1.775] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.043 0.093 0.147 0.204
 0.263 0.326 0.391 0.460 0.531 0.606 0.683 0.764 0.848 0.934 1.000 1.000]
deltaP_p_tot =  37.9 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.695 19.715 19.738 19.760 19.786 19.812 19.842 19.872 19.906 19.939
 19.977 20.015 20.057 20.098 20.144 20.189 20.238 20.287 20.341 20.394
 20.452 20.509 20.570 20.631] kJ/kg
T_s =  [6.1 6.1 6.1 6.1 6.2 6.2 6.2 6.2 6.3 6.3 6.4 6.4 6.4 6.5 6.5 6.6 6.6 6.7
 6.7 6.8 6.8 6.9 6.9 7.0] C
s_s =  [0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.075 0.075
 0.075 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.204 kW
Q_dot_s_tot = -0.204 kW
Q_dot_sen_tot =  0.204 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.463 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 436.6 kJ/kg
T_r_i = 3.0 C
T_r_e = 53.1 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.240 0.240 0.240 0.240]
  [0.239 0.239 0.239 0.239]]

 [[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.7 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.1 C

iteration# =  1.0     error = 3.001e+00
iteration# =  2.0     error = 1.059e+01
iteration# =  3.0     error = 4.777e+01
iteration# =  4.0     error = 2.152e+01
iteration# =  5.0     error = 1.280e+00
iteration# =  6.0     error = 4.463e-01
iteration# =  7.0     error = 1.532e-01
iteration# =  8.0     error = 3.420e-02
iteration# =  9.0     error = 9.160e-02
iteration# =  10.0     error = 1.260e-01
iteration# =  11.0     error = 1.215e-01
iteration# =  12.0     error = 9.266e-02
iteration# =  13.0     error = 5.657e-02
iteration# =  14.0     error = 2.610e-02
iteration# =  15.0     error = 6.222e-03
iteration# =  16.0     error = 3.903e-03
iteration# =  17.0     error = 7.387e-03
iteration# =  18.0     error = 7.261e-03
iteration# =  19.0     error = 5.656e-03
iteration# =  20.0     error = 3.790e-03
iteration# =  21.0     error = 2.221e-03
iteration# =  22.0     error = 1.112e-03
iteration# =  23.0     error = 4.284e-04
iteration# =  24.0     error = 1.049e-04
iteration# =  25.0     error = 9.600e-05
```

```
Circuit Number:  24.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.2 226.2 232.8 239.9 247.4 255.5 264.0 273.0 282.6
 292.6 303.1 314.2 325.7 337.8 350.3 363.4 377.0 391.0 405.6 420.7 436.3] kJ/kg
T_p =  [3.0 6.9 10.7 14.7 19.1 23.8 28.7 33.9 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 38.0 52.8] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.113 1.137 1.162 1.188 1.216 1.245 1.276
 1.309 1.343 1.379 1.416 1.456 1.496 1.539 1.583 1.629 1.676 1.725 1.774] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.042 0.093 0.146 0.203
 0.262 0.325 0.390 0.459 0.530 0.605 0.682 0.763 0.846 0.933 1.000 1.000]
deltaP_p_tot =  37.9 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.631 19.651 19.680 19.701 19.733 19.759 19.795 19.824 19.864 19.897
 19.941 19.978 20.026 20.067 20.118 20.163 20.219 20.267 20.327 20.380
 20.443 20.500 20.568 20.628] kJ/kg
T_s =  [6.0 6.0 6.1 6.1 6.1 6.1 6.2 6.2 6.2 6.3 6.3 6.4 6.4 6.4 6.5 6.5 6.6 6.6
 6.7 6.7 6.8 6.9 6.9 7.0] C
s_s =  [0.073 0.073 0.073 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.074 0.075
 0.075 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.204 kW
Q_dot_s_tot = -0.204 kW
Q_dot_sen_tot =  0.204 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.464 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 436.3 kJ/kg
T_r_i = 3.0 C
T_r_e = 52.8 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.238 0.238 0.238 0.238]
  [0.237 0.237 0.237 0.237]]

 [[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.6 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 6.0 C


iteration# =  1.0     error = 3.000e+00
iteration# =  2.0     error = 9.488e+00
iteration# =  3.0     error = 1.464e+01
iteration# =  4.0     error = 1.265e+02
iteration# =  5.0     error = 1.183e+00
iteration# =  6.0     error = 4.242e-01
iteration# =  7.0     error = 1.436e-01
iteration# =  8.0     error = 3.411e-02
iteration# =  9.0     error = 9.373e-02
iteration# =  10.0     error = 1.263e-01
iteration# =  11.0     error = 1.207e-01
iteration# =  12.0     error = 9.134e-02
iteration# =  13.0     error = 5.534e-02
iteration# =  14.0     error = 2.522e-02
iteration# =  15.0     error = 5.718e-03
```

```
iteration# =  16.0     error = 4.128e-03
iteration# =  17.0     error = 7.447e-03
iteration# =  18.0     error = 7.242e-03
iteration# =  19.0     error = 5.610e-03
iteration# =  20.0     error = 3.744e-03
iteration# =  21.0     error = 2.186e-03
iteration# =  22.0     error = 1.088e-03
iteration# =  23.0     error = 4.144e-04
iteration# =  24.0     error = 1.022e-04
iteration# =  25.0     error = 9.893e-05

Circuit Number:  25.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.1 226.2 232.7 239.8 247.3 255.4 263.9 272.9 282.4
 292.5 303.0 314.0 325.5 337.6 350.1 363.2 376.7 390.8 405.3 420.4 436.0] kJ/kg
T_p =  [3.0 6.9 10.7 14.7 19.1 23.7 28.7 33.9 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 37.7 52.5] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.113 1.137 1.162 1.188 1.215 1.245 1.276
 1.308 1.342 1.378 1.416 1.455 1.496 1.538 1.582 1.628 1.675 1.724 1.773] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.042 0.092 0.146 0.202
 0.262 0.324 0.389 0.458 0.529 0.603 0.681 0.761 0.845 0.931 1.000 1.000]
deltaP_p_tot =  37.8 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.323 101.324 101.325] kPa
h_s =  [19.567 19.587 19.621 19.643 19.680 19.706 19.747 19.777 19.822 19.855
 19.904 19.941 19.994 20.035 20.093 20.138 20.199 20.247 20.313 20.365
 20.435 20.491 20.565 20.625] kJ/kg
T_s =  [5.9 6.0 6.0 6.0 6.1 6.1 6.1 6.2 6.2 6.2 6.3 6.3 6.4 6.4 6.5 6.5 6.6 6.6
 6.7 6.7 6.8 6.9 6.9 7.0] C
s_s =  [0.073 0.073 0.073 0.073 0.073 0.074 0.074 0.074 0.074 0.074 0.074 0.074
 0.075 0.075 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.204 kW
Q_dot_s_tot = -0.204 kW
Q_dot_sen_tot =  0.204 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.464 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 436.0 kJ/kg
T_r_i = 3.0 C
T_r_e = 52.5 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.237 0.237 0.237 0.237]
  [0.236 0.236 0.236 0.236]]

 [[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.6 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 5.9 C

iteration# =  1.0     error = 2.998e+00
iteration# =  2.0     error = 1.472e+01
iteration# =  3.0     error = 4.890e+01
iteration# =  4.0     error = 3.864e+01
iteration# =  5.0     error = 1.096e+00
```

```
iteration# =  6.0     error = 4.027e-01
iteration# =  7.0     error = 1.340e-01
iteration# =  8.0     error = 3.402e-02
iteration# =  9.0     error = 9.591e-02
iteration# =  10.0     error = 1.266e-01
iteration# =  11.0     error = 1.199e-01
iteration# =  12.0     error = 9.003e-02
iteration# =  13.0     error = 5.411e-02
iteration# =  14.0     error = 2.434e-02
iteration# =  15.0     error = 5.213e-03
iteration# =  16.0     error = 4.356e-03
iteration# =  17.0     error = 7.510e-03
iteration# =  18.0     error = 7.226e-03
iteration# =  19.0     error = 5.567e-03
iteration# =  20.0     error = 3.700e-03
iteration# =  21.0     error = 2.151e-03
iteration# =  22.0     error = 1.065e-03
iteration# =  23.0     error = 4.003e-04
iteration# =  24.0     error = 9.945e-05

Circuit Number:  26.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.6 220.1 226.1 232.7 239.7 247.2 255.2 263.8 272.8 282.3
 292.3 302.8 313.8 325.3 337.3 349.9 362.9 376.4 390.5 405.0 420.1 435.7] kJ/kg
T_p =  [3.0 6.9 10.7 14.7 19.0 23.7 28.6 33.8 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6 37.4 52.2] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.113 1.137 1.161 1.187 1.215 1.244 1.275
 1.308 1.342 1.378 1.415 1.454 1.495 1.537 1.581 1.627 1.674 1.723 1.772] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.041 0.091 0.145 0.201
 0.261 0.323 0.388 0.456 0.528 0.602 0.679 0.760 0.843 0.929 1.000 1.000]
deltaP_p_tot =  37.8 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.323 101.324 101.325] kPa
h_s =  [19.503 19.523 19.563 19.584 19.627 19.653 19.700 19.729 19.780 19.813
 19.868 19.905 19.963 20.004 20.067 20.112 20.179 20.228 20.298 20.351
 20.426 20.483 20.562 20.622] kJ/kg
T_s =  [5.9 5.9 5.9 6.0 6.0 6.0 6.1 6.1 6.2 6.2 6.2 6.3 6.3 6.4 6.4 6.5 6.6 6.6
 6.7 6.7 6.8 6.9 6.9 7.0] C
s_s =  [0.073 0.073 0.073 0.073 0.073 0.073 0.074 0.074 0.074 0.074 0.074 0.074
 0.074 0.075 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.204 kW
Q_dot_s_tot = -0.204 kW
Q_dot_sen_tot =  0.204 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.462 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 435.7 kJ/kg
T_r_i = 3.0 C
T_r_e = 52.2 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.235 0.235 0.235 0.235]
  [0.234 0.234 0.234 0.234]]

 [[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.5 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
```

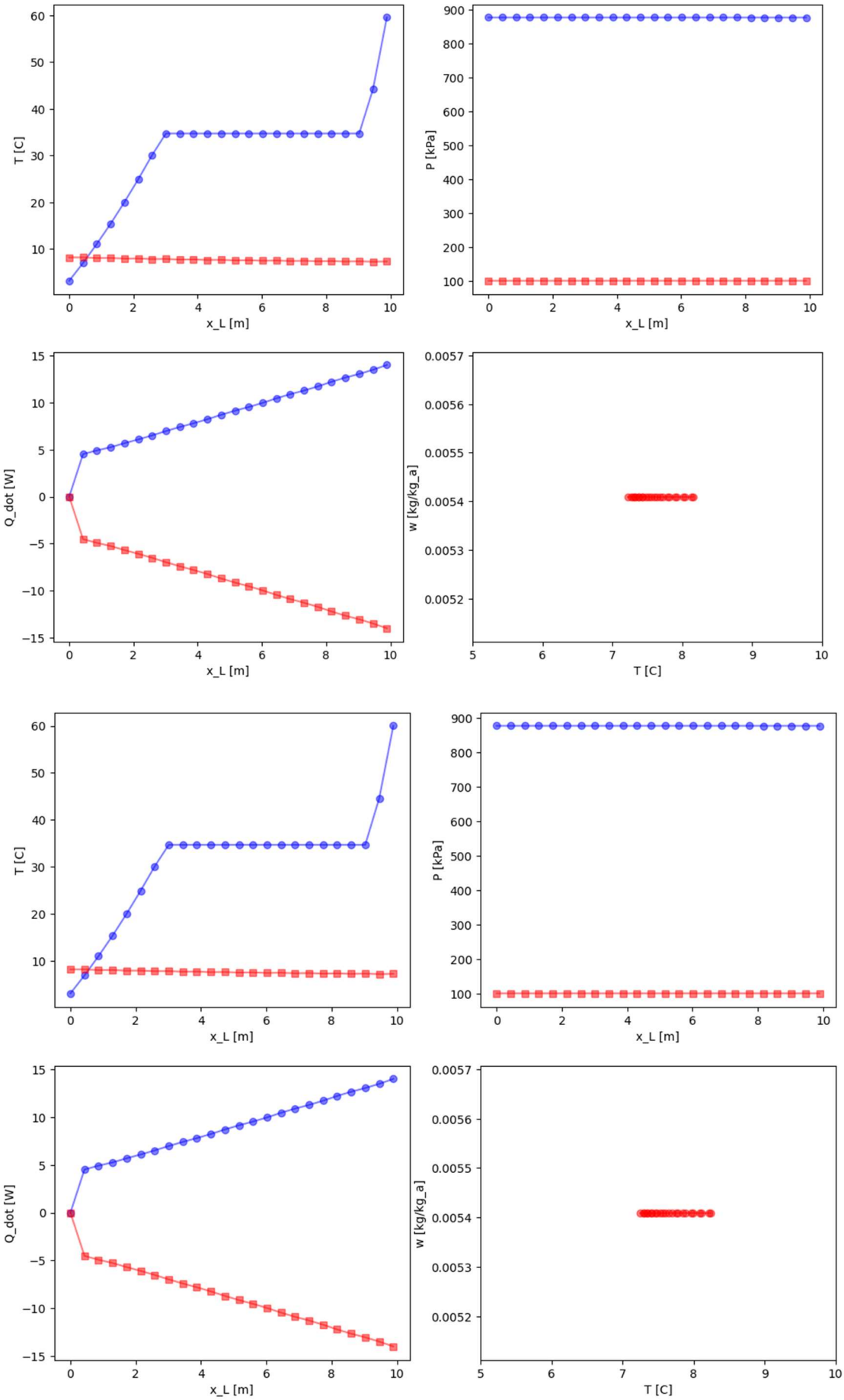
```
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 5.9 C

iteration# =  1.0     error = 2.995e+00
iteration# =  2.0     error = 3.035e+01
iteration# =  3.0     error = 3.180e+01
iteration# =  4.0     error = 7.119e+01
iteration# =  5.0     error = 1.017e+00
iteration# =  6.0     error = 3.819e-01
iteration# =  7.0     error = 1.245e-01
iteration# =  8.0     error = 3.393e-02
iteration# =  9.0     error = 9.813e-02
iteration# =  10.0     error = 1.270e-01
iteration# =  11.0     error = 1.191e-01
iteration# =  12.0     error = 8.873e-02
iteration# =  13.0     error = 5.288e-02
iteration# =  14.0     error = 2.346e-02
iteration# =  15.0     error = 4.873e-03
iteration# =  16.0     error = 4.588e-03
iteration# =  17.0     error = 7.577e-03
iteration# =  18.0     error = 7.213e-03
iteration# =  19.0     error = 5.525e-03
iteration# =  20.0     error = 3.656e-03
iteration# =  21.0     error = 2.116e-03
iteration# =  22.0     error = 1.041e-03
iteration# =  23.0     error = 3.862e-04
iteration# =  24.0     error = 9.670e-05

Circuit Number:  27.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.5 220.1 226.1 232.6 239.6 247.1 255.1 263.6 272.6 282.1
 292.1 302.6 313.6 325.1 337.1 349.6 362.6 376.1 390.1 404.7 419.7 435.3] kJ/kg
T_p =  [3.0 6.9 10.6 14.7 19.0 23.6 28.6 33.7 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6 37.1 51.8] C
s_p =  [1.014 1.032 1.051 1.070 1.091 1.113 1.136 1.161 1.187 1.215 1.244 1.275
 1.307 1.341 1.377 1.414 1.453 1.494 1.536 1.580 1.626 1.673 1.722 1.771] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.040 0.091 0.144 0.200
 0.259 0.322 0.387 0.455 0.526 0.600 0.678 0.758 0.841 0.927 1.000 1.000]
deltaP_p_tot =  37.7 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.439 19.459 19.504 19.526 19.574 19.600 19.652 19.681 19.738 19.771
 19.831 19.868 19.932 19.973 20.042 20.086 20.159 20.208 20.284 20.337
 20.417 20.474 20.559 20.620] kJ/kg
T_s =  [5.8 5.8 5.9 5.9 6.0 6.0 6.0 6.1 6.1 6.2 6.2 6.2 6.3 6.3 6.4 6.5 6.5 6.6
 6.7 6.7 6.8 6.8 6.9 7.0] C
s_s =  [0.073 0.073 0.073 0.073 0.073 0.073 0.073 0.073 0.074 0.074 0.074 0.074
 0.074 0.074 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.203 kW
Q_dot_s_tot = -0.203 kW
Q_dot_sen_tot =  0.203 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.459 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 435.3 kJ/kg
T_r_i = 3.0 C
T_r_e = 51.8 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]

 [[0.233 0.233 0.233 0.233]
```

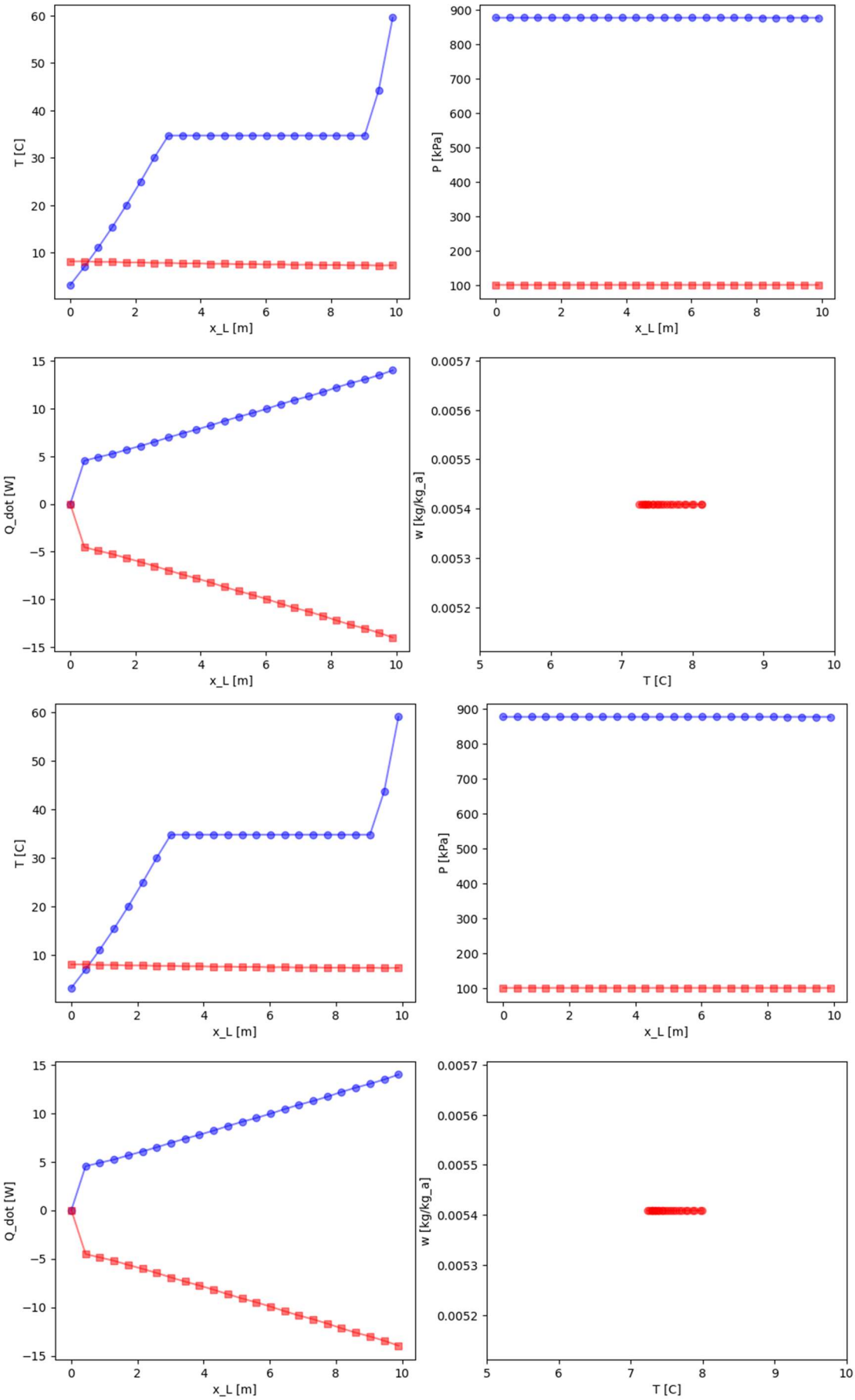
```
  [0.232 0.232 0.232 0.232]]

 [[0.226 0.226 0.226 0.226]
  [0.226 0.226 0.226 0.226]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.4 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 5.8 C


iteration# =  1.0     error = 2.989e+00
iteration# =  2.0     error = 8.112e+01
iteration# =  3.0     error = 4.966e+01
iteration# =  4.0     error = 1.247e+01
iteration# =  5.0     error = 9.131e-01
iteration# =  6.0     error = 3.523e-01
iteration# =  7.0     error = 1.107e-01
iteration# =  8.0     error = 3.384e-02
iteration# =  9.0     error = 1.014e-01
iteration# =  10.0     error = 1.275e-01
iteration# =  11.0     error = 1.179e-01
iteration# =  12.0     error = 8.686e-02
iteration# =  13.0     error = 5.112e-02
iteration# =  14.0     error = 2.219e-02
iteration# =  15.0     error = 4.539e-03
iteration# =  16.0     error = 4.926e-03
iteration# =  17.0     error = 7.678e-03
iteration# =  18.0     error = 7.197e-03
iteration# =  19.0     error = 5.467e-03
iteration# =  20.0     error = 3.595e-03
iteration# =  21.0     error = 2.067e-03
iteration# =  22.0     error = 1.007e-03
iteration# =  23.0     error = 3.660e-04
iteration# =  24.0     error = 9.276e-05

Circuit Number:  28.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.5 220.0 226.0 232.5 239.5 247.0 255.0 263.4 272.4 281.9
 291.8 302.3 313.3 324.7 336.7 349.2 362.2 375.6 389.6 404.2 419.2 434.7] kJ/kg
T_p =  [3.0 6.9 10.6 14.6 19.0 23.6 28.5 33.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6 36.6 51.3] C
s_p =  [1.014 1.032 1.050 1.070 1.090 1.113 1.136 1.160 1.186 1.214 1.243 1.274
 1.306 1.340 1.376 1.413 1.452 1.493 1.535 1.579 1.624 1.671 1.720 1.769] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.039 0.089 0.143 0.199
 0.258 0.320 0.385 0.453 0.524 0.598 0.675 0.755 0.838 0.924 1.000 1.000]
deltaP_p_tot =  37.6 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.312 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.345 19.365 19.418 19.440 19.496 19.522 19.582 19.611 19.676 19.709
 19.777 19.814 19.887 19.928 20.005 20.049 20.130 20.179 20.264 20.317
 20.406 20.463 20.556 20.617] kJ/kg
T_s =  [5.7 5.8 5.8 5.8 5.9 5.9 6.0 6.0 6.1 6.1 6.2 6.2 6.3 6.3 6.4 6.4 6.5 6.6
 6.6 6.7 6.8 6.8 6.9 7.0] C
s_s =  [0.072 0.072 0.073 0.073 0.073 0.073 0.073 0.073 0.073 0.074 0.074 0.074
 0.074 0.074 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.203 kW
Q_dot_s_tot = -0.203 kW
Q_dot_sen_tot =  0.203 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.455 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
```

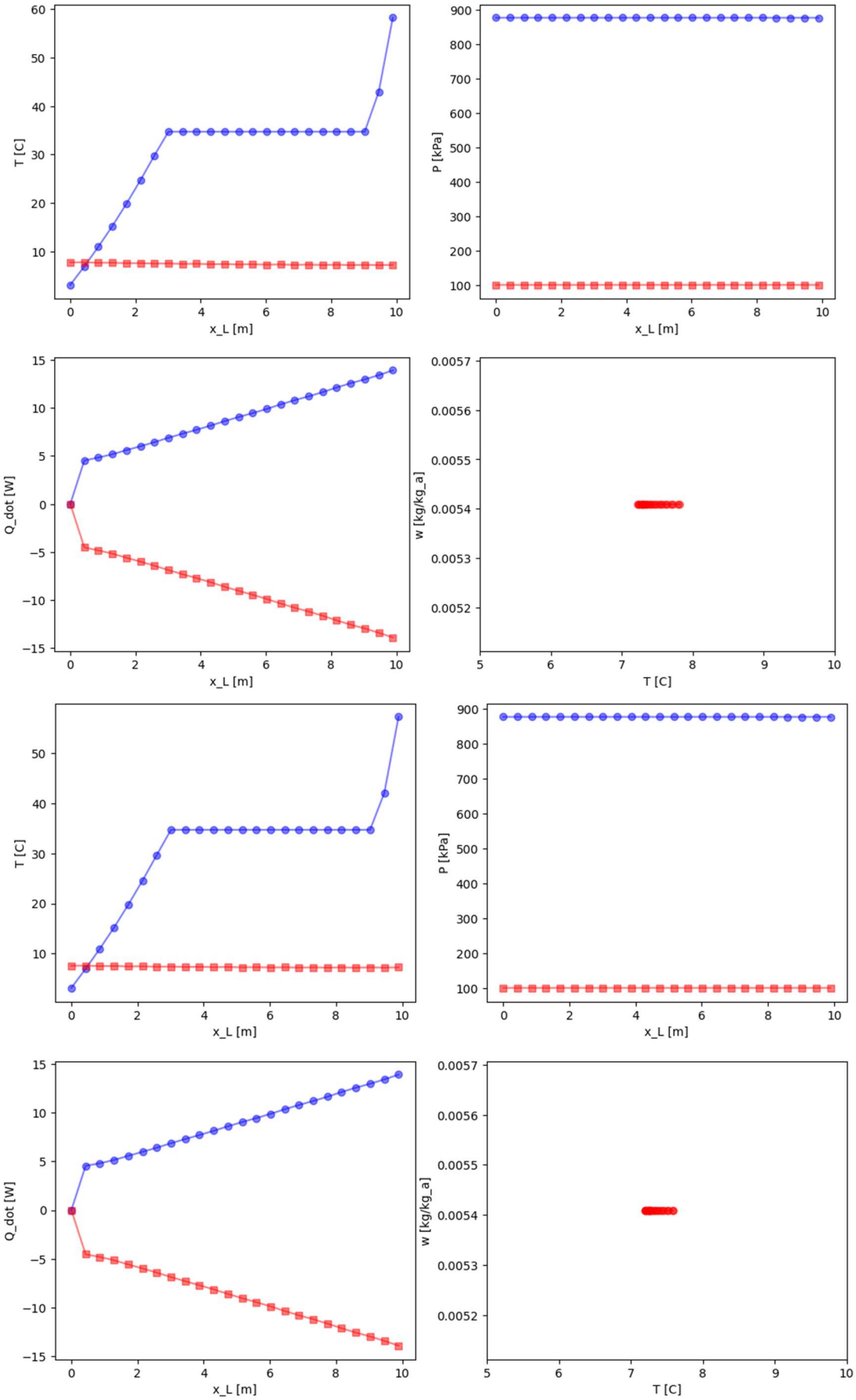
```
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 434.7 kJ/kg
T_r_i = 3.0 C
T_r_e = 51.3 C

m_dot_s =  [[[0.226 0.226 0.226 0.226]
  [0.225 0.226 0.226 0.225]]

 [[0.231 0.232 0.232 0.231]
  [0.230 0.231 0.231 0.230]]

 [[0.226 0.226 0.226 0.226]
  [0.225 0.226 0.226 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.3 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 5.7 C

iteration# =  1.0     error = 2.983e+00
iteration# =  2.0     error = 1.682e+01
iteration# =  3.0     error = 1.794e+01
iteration# =  4.0     error = 5.490e+00
iteration# =  5.0     error = 8.189e-01
iteration# =  6.0     error = 3.227e-01
iteration# =  7.0     error = 9.644e-02
iteration# =  8.0     error = 3.717e-02
iteration# =  9.0     error = 1.050e-01
iteration# =  10.0     error = 1.281e-01
iteration# =  11.0     error = 1.166e-01
iteration# =  12.0     error = 8.485e-02
iteration# =  13.0     error = 4.922e-02
iteration# =  14.0     error = 2.083e-02
iteration# =  15.0     error = 4.218e-03
iteration# =  16.0     error = 5.286e-03
iteration# =  17.0     error = 7.783e-03
iteration# =  18.0     error = 7.177e-03
iteration# =  19.0     error = 5.403e-03
iteration# =  20.0     error = 3.528e-03
iteration# =  21.0     error = 2.013e-03
iteration# =  22.0     error = 9.708e-04
iteration# =  23.0     error = 3.440e-04
iteration# =  24.0     error = 8.837e-05

Circuit Number:  29.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.5 220.0 226.0 232.4 239.4 246.8 254.8 263.2 272.2 281.6
 291.5 302.0 312.9 324.4 336.3 348.7 361.7 375.1 389.1 403.6 418.6 434.1] kJ/kg
T_p =  [3.0 6.9 10.6 14.6 18.9 23.5 28.4 33.5 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 36.0 50.7] C
s_p =  [1.014 1.032 1.050 1.070 1.090 1.112 1.135 1.160 1.186 1.213 1.242 1.273
 1.305 1.339 1.375 1.412 1.451 1.491 1.533 1.577 1.622 1.669 1.718 1.767] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.038 0.088 0.141 0.197
 0.256 0.318 0.383 0.451 0.521 0.595 0.672 0.752 0.835 0.921 1.000 1.000]
deltaP_p_tot =  37.5 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.248 19.268 19.329 19.351 19.415 19.441 19.509 19.538 19.611 19.644
 19.721 19.758 19.839 19.880 19.965 20.009 20.099 20.147 20.241 20.294
 20.391 20.448 20.550 20.611] kJ/kg
T_s =  [5.6 5.7 5.7 5.7 5.8 5.8 5.9 5.9 6.0 6.0 6.1 6.1 6.2 6.3 6.3 6.4 6.5 6.5
 6.6 6.7 6.8 6.8 6.9 7.0] C
s_s =  [0.072 0.072 0.072 0.072 0.073 0.073 0.073 0.073 0.073 0.073 0.074 0.074
 0.074 0.074 0.074 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa
```

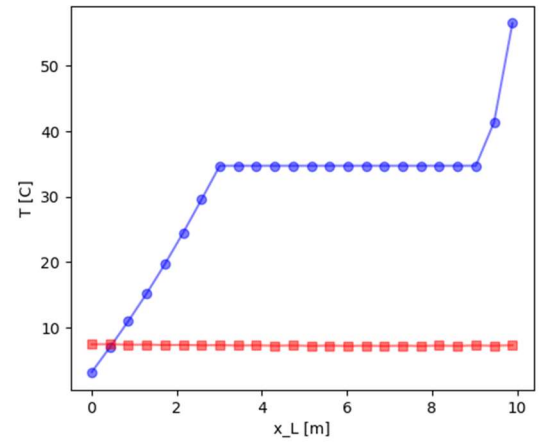
```
Q_dot_p_tot =  0.202 kW
Q_dot_s_tot = -0.202 kW
Q_dot_sen_tot =  0.202 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.449 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 434.1 kJ/kg
T_r_i = 3.0 C
T_r_e = 50.7 C

m_dot_s =  [[[0.225 0.226 0.226 0.225]
  [0.225 0.225 0.225 0.225]]

 [[0.229 0.230 0.230 0.229]
  [0.228 0.229 0.229 0.228]]

 [[0.225 0.226 0.226 0.225]
  [0.225 0.225 0.225 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.2 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 5.6 C

iteration# =  1.0     error = 2.975e+00
iteration# =  2.0     error = 1.314e+01
iteration# =  3.0     error = 1.934e+02
iteration# =  4.0     error = 3.973e+00
iteration# =  5.0     error = 7.632e-01
iteration# =  6.0     error = 3.039e-01
iteration# =  7.0     error = 8.712e-02
iteration# =  8.0     error = 4.205e-02
iteration# =  9.0     error = 1.074e-01
iteration# =  10.0     error = 1.286e-01
iteration# =  11.0     error = 1.159e-01
iteration# =  12.0     error = 8.358e-02
iteration# =  13.0     error = 4.800e-02
iteration# =  14.0     error = 1.995e-02
iteration# =  15.0     error = 4.038e-03
iteration# =  16.0     error = 5.531e-03
iteration# =  17.0     error = 7.863e-03
iteration# =  18.0     error = 7.174e-03
iteration# =  19.0     error = 5.368e-03
iteration# =  20.0     error = 3.487e-03
iteration# =  21.0     error = 1.980e-03
iteration# =  22.0     error = 9.475e-04
iteration# =  23.0     error = 3.296e-04
iteration# =  24.0     error = 8.611e-05

Circuit Number:  30.0
Results

p_p =  [876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0
 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0 876.0] kPa
h_p =  [204.2 209.4 214.5 219.9 225.9 232.3 239.3 246.7 254.6 263.1 272.0 281.4
 291.3 301.7 312.6 324.0 336.0 348.4 361.3 374.7 388.7 403.1 418.1 433.5] kJ/kg
T_p =  [3.0 6.8 10.6 14.6 18.9 23.4 28.3 33.4 34.6 34.6 34.6 34.6 34.6 34.6 34.6
 34.6 34.6 34.6 34.6 34.6 34.6 34.6 34.6 35.6 50.1] C
s_p =  [1.014 1.032 1.050 1.069 1.090 1.112 1.135 1.160 1.185 1.213 1.242 1.272
 1.305 1.338 1.374 1.411 1.450 1.490 1.532 1.576 1.621 1.668 1.717 1.765] kJ/kgK
x_p =  [0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.037 0.087 0.140 0.196
 0.255 0.317 0.381 0.449 0.520 0.593 0.670 0.749 0.832 0.918 1.000 1.000]
deltaP_p_tot =  37.4 Pa

p_s =  [101.307 101.308 101.309 101.309 101.310 101.311 101.312 101.313 101.313
 101.314 101.315 101.316 101.316 101.317 101.318 101.319 101.320 101.320
```

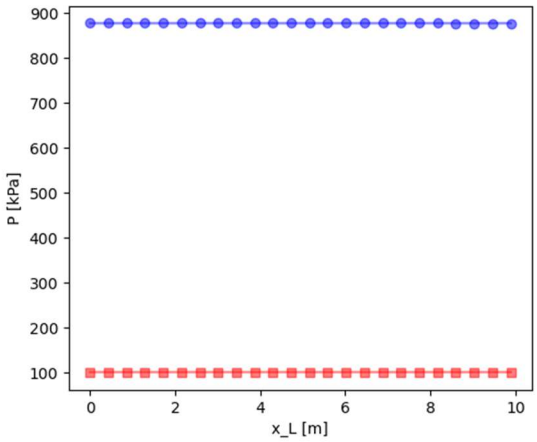
```
 101.321 101.322 101.323 101.323 101.324 101.325] kPa
h_s =  [19.183 19.204 19.270 19.292 19.362 19.387 19.461 19.490 19.568 19.601
 19.684 19.721 19.807 19.848 19.939 19.983 20.078 20.127 20.226 20.279
 20.383 20.439 20.547 20.608] kJ/kg
T_s =  [5.6 5.6 5.7 5.7 5.7 5.8 5.8 5.9 6.0 6.0 6.1 6.1 6.2 6.2 6.3 6.4 6.5 6.5
 6.6 6.7 6.8 6.8 6.9 7.0] C
s_s =  [0.072 0.072 0.072 0.072 0.072 0.072 0.073 0.073 0.073 0.073 0.073 0.074
 0.074 0.074 0.074 0.075 0.075 0.075 0.075 0.076 0.076 0.076 0.077 0.077] kJ/kgK
deltaP_s_tot =  0.018 kPa

Q_dot_p_tot =  0.202 kW
Q_dot_s_tot = -0.202 kW
Q_dot_sen_tot =  0.202 kW
Q_dot_lat_tot =  0.000 kW

UA_overall = 7.443 W/K
m_dot_p = 0.001 kg/s
P_r_i = 876.0 kPa
P_r_e = 876.0 kPa
h_r_i = 204.2 kJ/kg
h_r_e = 433.5 kJ/kg
T_r_i = 3.0 C
T_r_e = 50.1 C

m_dot_s =  [[[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]

 [[0.228 0.228 0.228 0.228]
  [0.227 0.227 0.227 0.227]]

 [[0.225 0.225 0.225 0.225]
  [0.225 0.225 0.225 0.225]]] kg/s
P_a_i = 101.3 kPa
P_a_e = 101.3 kPa
h_a_i = 20.6 kJ/kg dry air
h_a_e = 19.2 kJ/kg dry air
w_a_i = 0.00541 kg/kg dry air
w_a_e = 0.00541 kg/kg dry air
T_a_i = 7.0 C
T_a_e = 5.6 C


Total heat transfer on the air side = -6.171 kW
Total heat transfer on the refrigerant side =  6.171 kW
Total pressure losses on the refrigerant side =  1.120 kPa
```

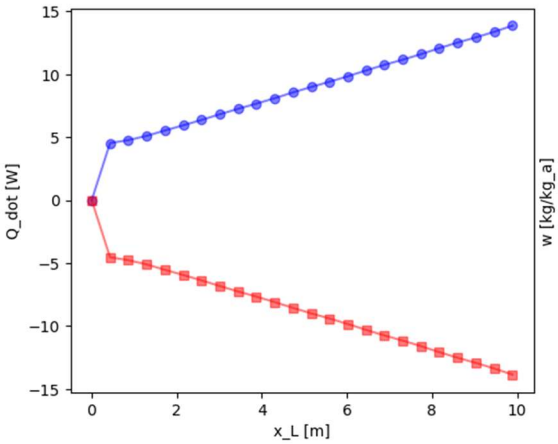

T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]
T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]

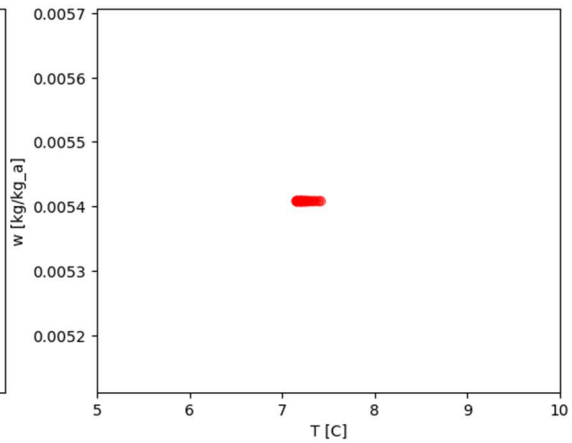

T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]
T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]

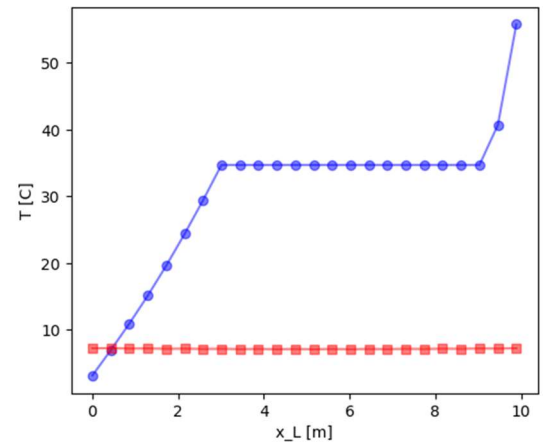

T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]
T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]

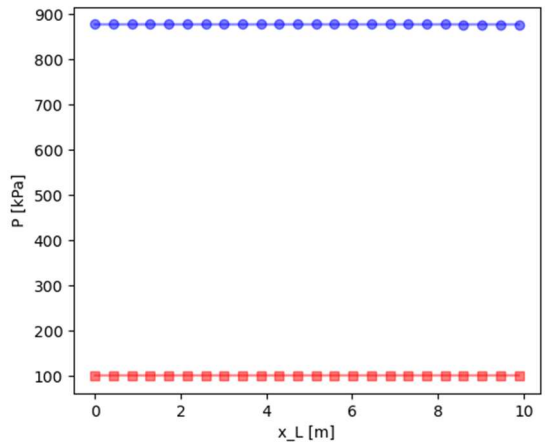

T [C]
x_L [m]

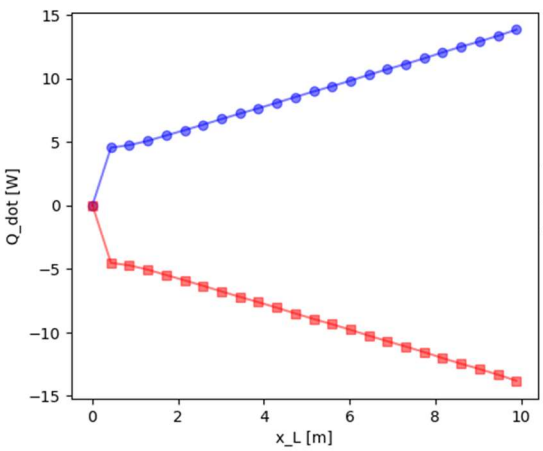

P [kPa]
x_L [m]

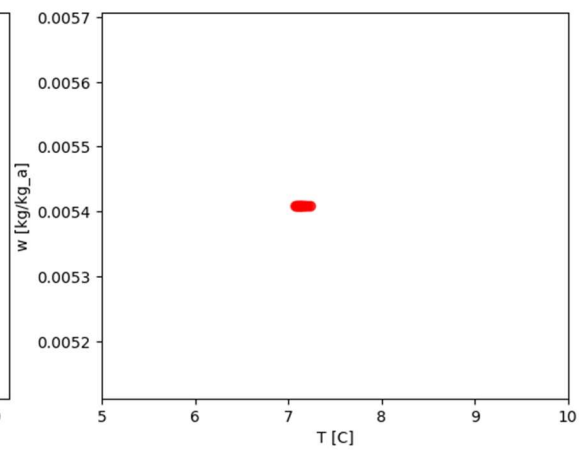

Q_dot [W]
x_L [m]

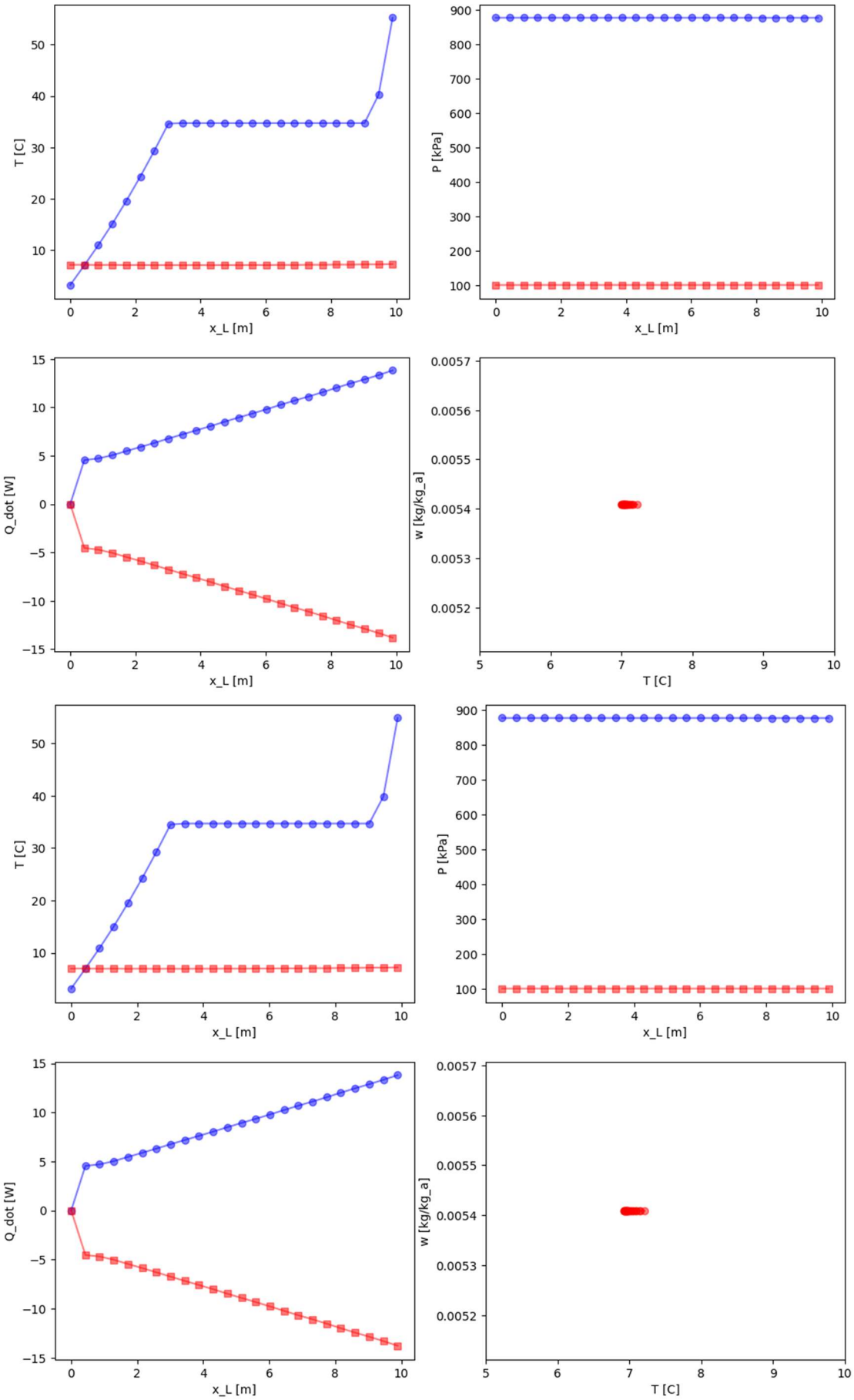

w [kg/kg_a]
T [C]

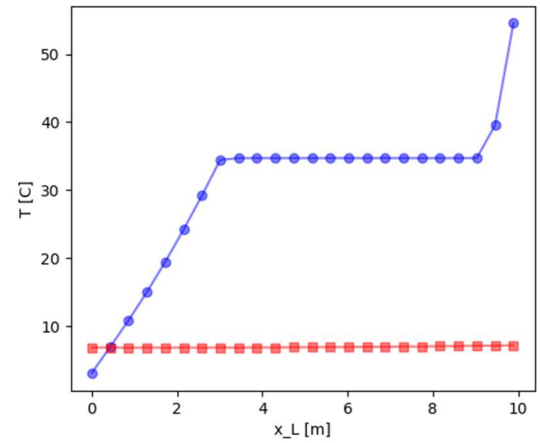

T [C]
x_L [m]

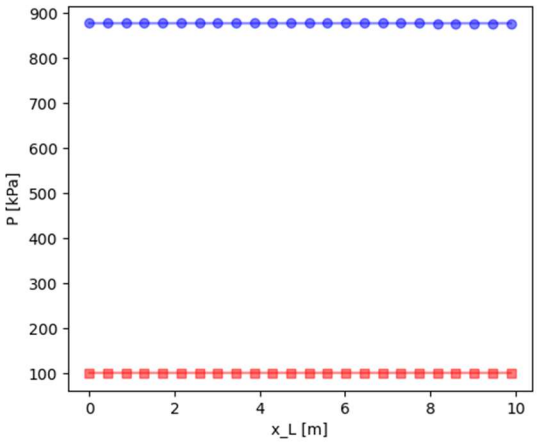

P [kPa]
x_L [m]

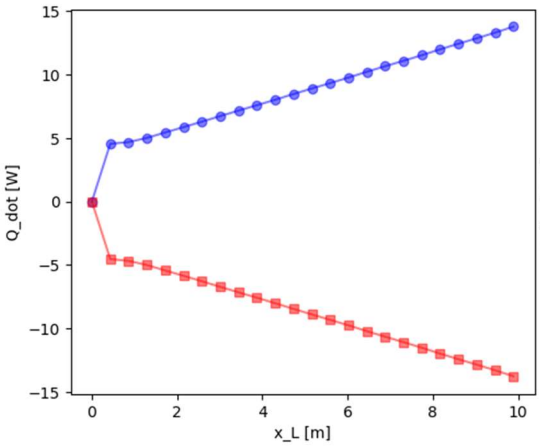

Q_dot [W]
x_L [m]

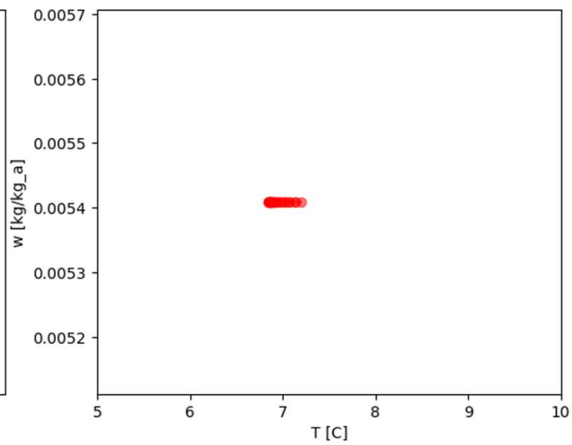

w [kg/kg_a]
T [C]

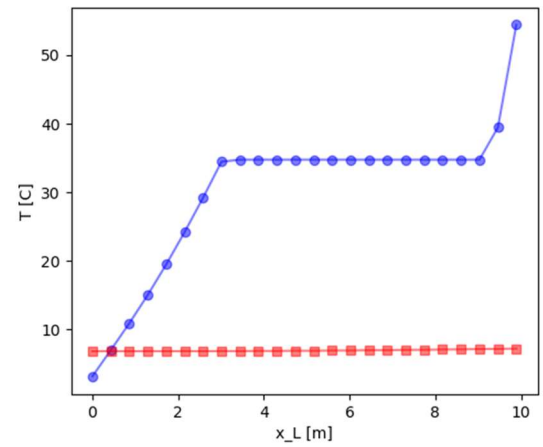

T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]
T [C]
x_L [m]
P [kPa]
x_L [m]
Q_dot [W]
x_L [m]
w [kg/kg_a]
T [C]

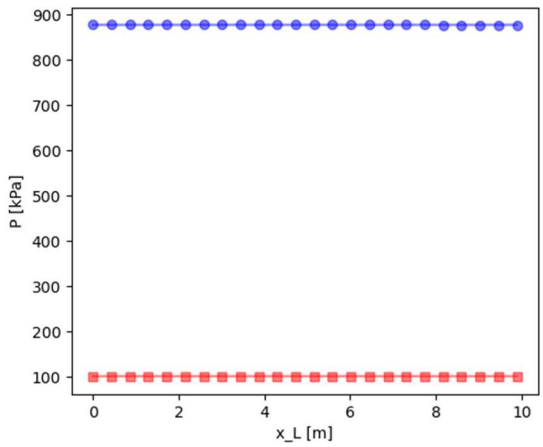

T [C]
x_L [m]

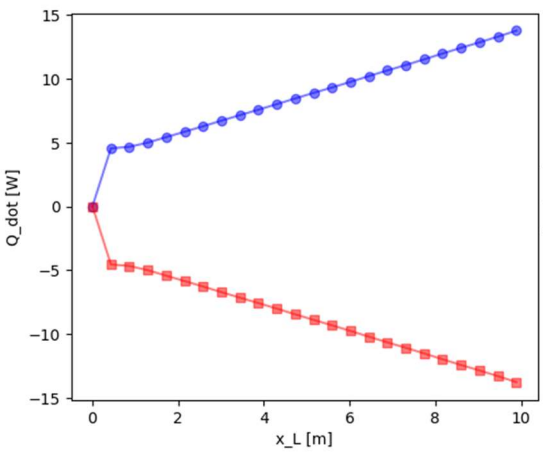

P [kPa]
x_L [m]

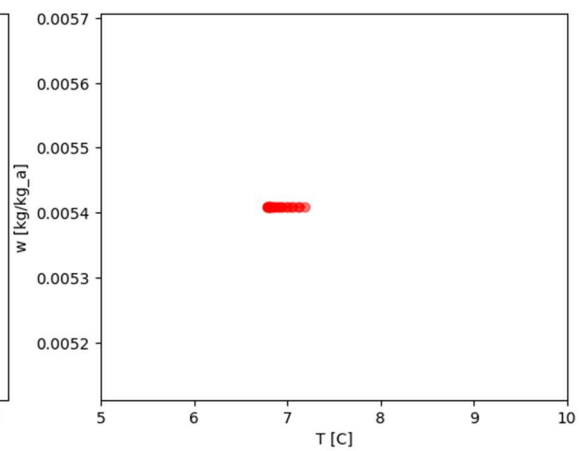

Q_dot [W]
x_L [m]

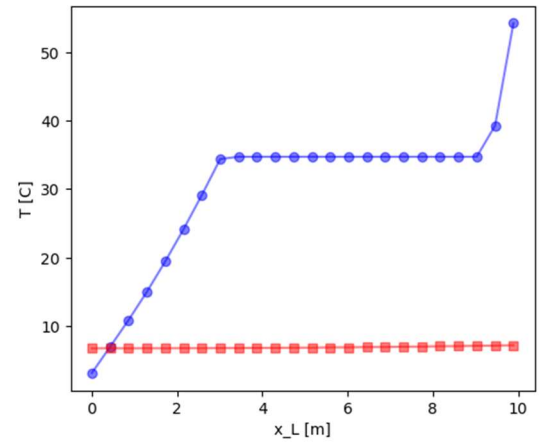

w [kg/kg_a]
T [C]

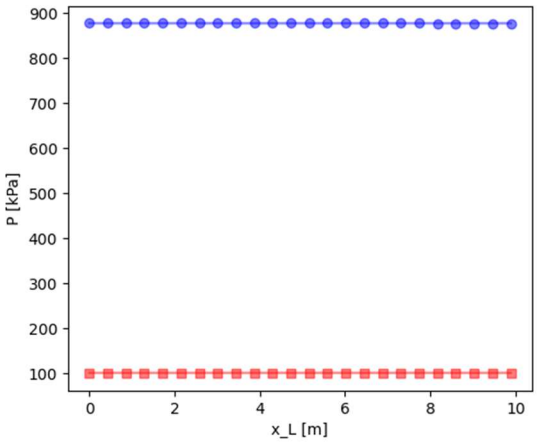

T [C]
x_L [m]

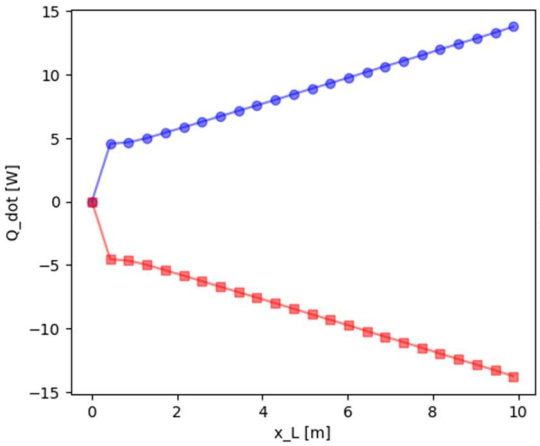

P [kPa]
x_L [m]

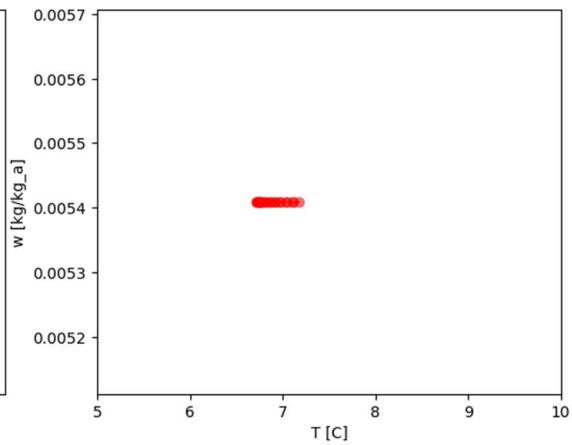

Q_dot [W]
x_L [m]

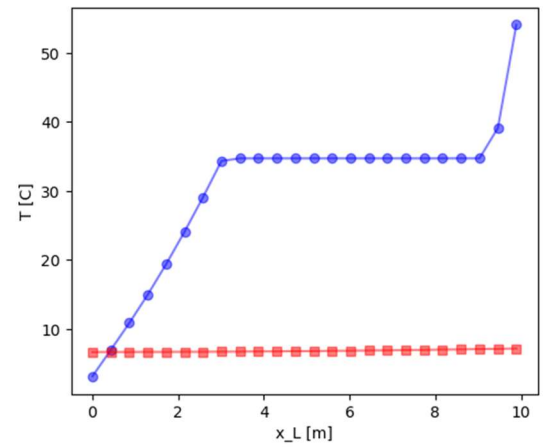

w [kg/kg_a]
T [C]

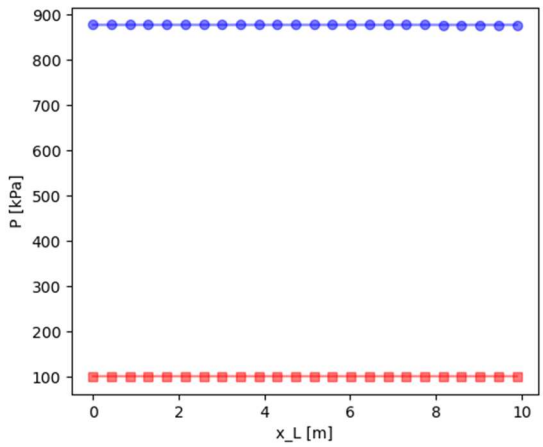

T [C]
x_L [m]

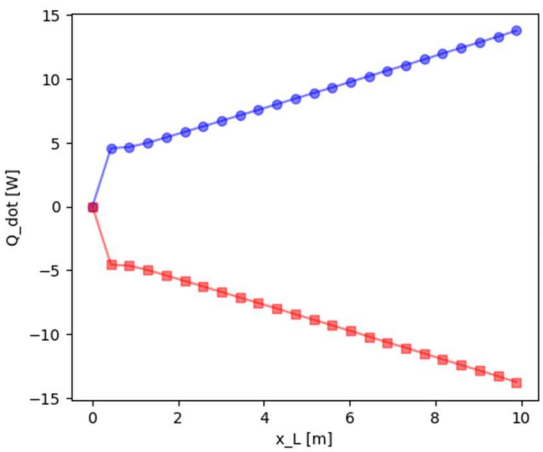

P [kPa]
x_L [m]

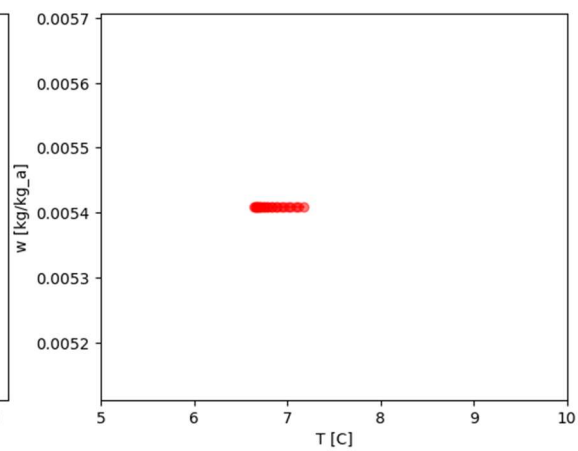

Q_dot [W]
x_L [m]

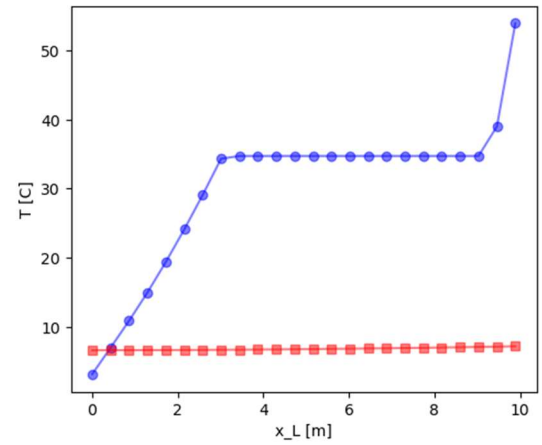

w [kg/kg_a]
T [C]

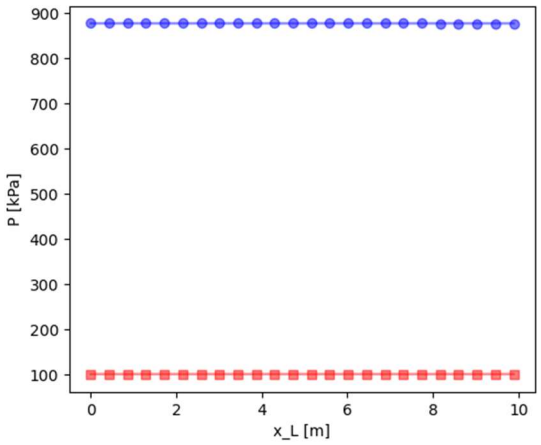

T [C]
x_L [m]

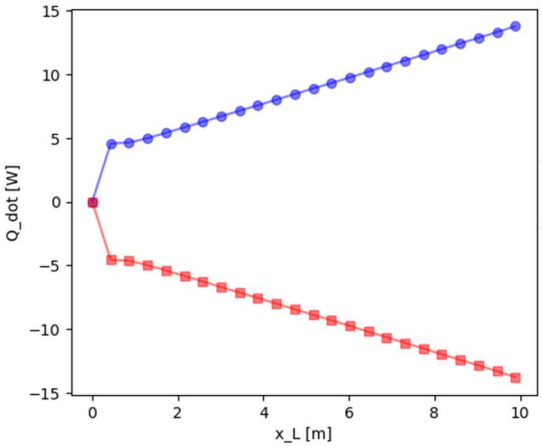

P [kPa]
x_L [m]

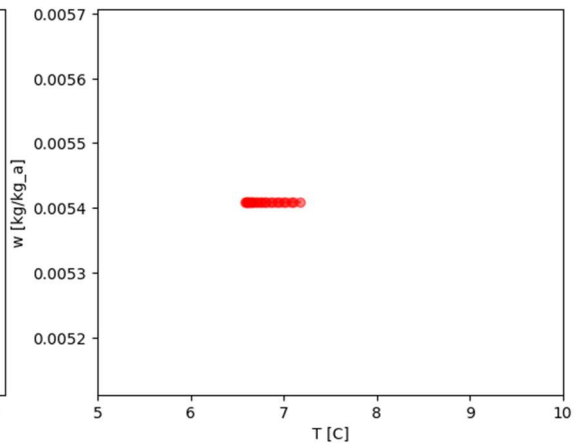

Q_dot [W]
x_L [m]

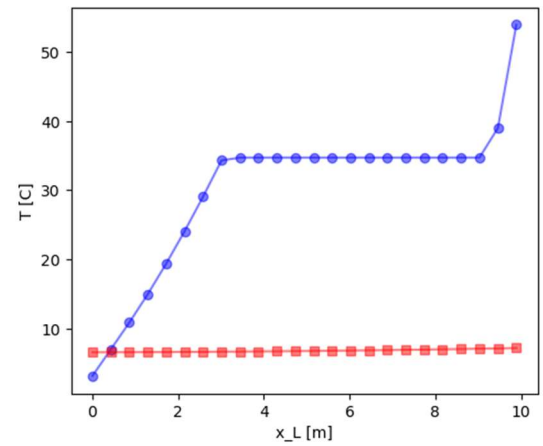

w [kg/kg_a]
T [C]

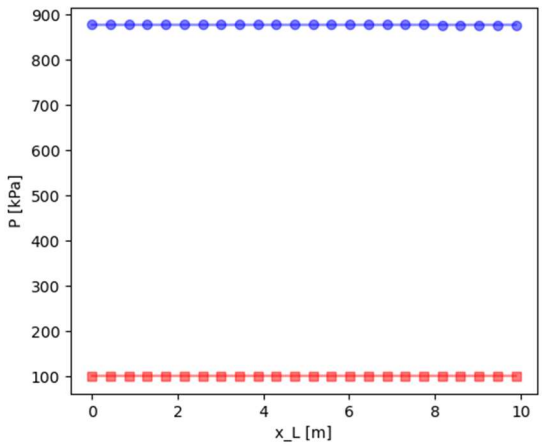

T [C]
x_L [m]

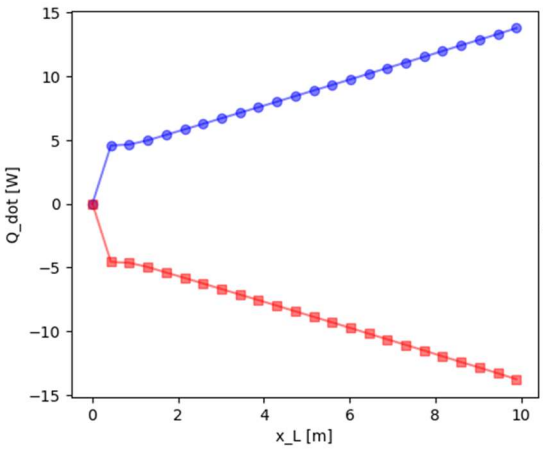

P [kPa]
x_L [m]

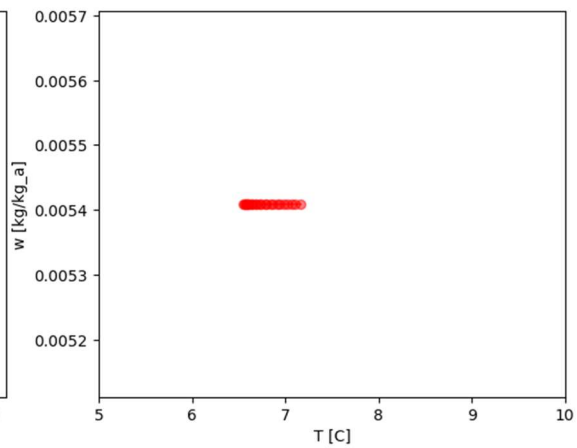

Q_dot [W]
x_L [m]

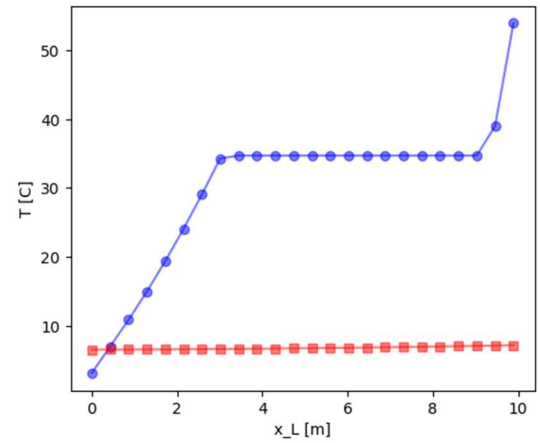

w [kg/kg_a]
T [C]

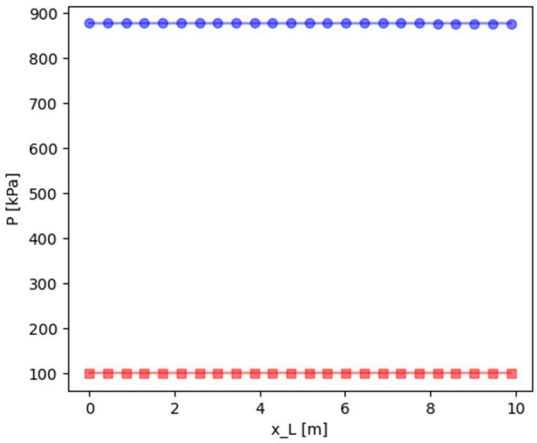

T [C]
x_L [m]

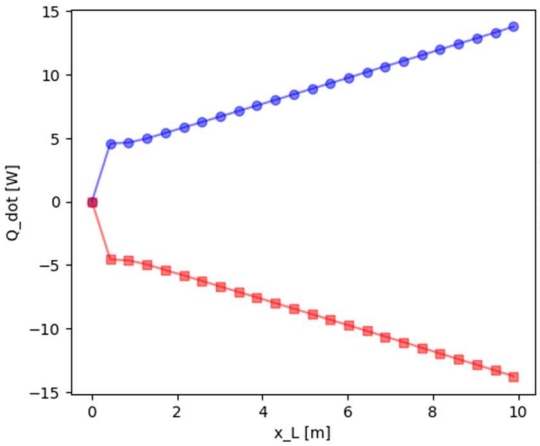

P [kPa]
x_L [m]

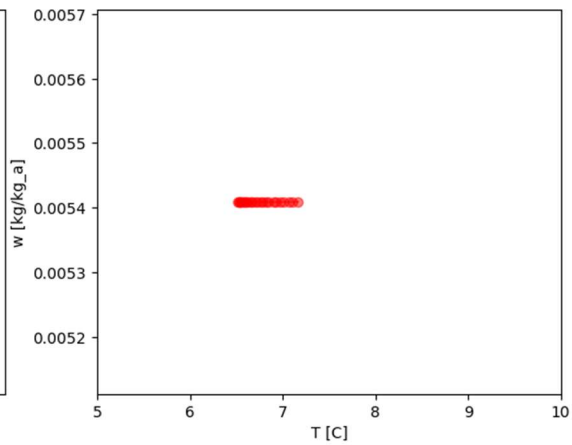

Q_dot [W]
x_L [m]

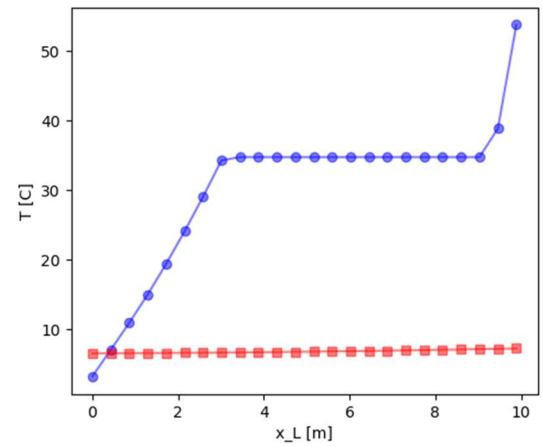

w [kg/kg_a]
T [C]

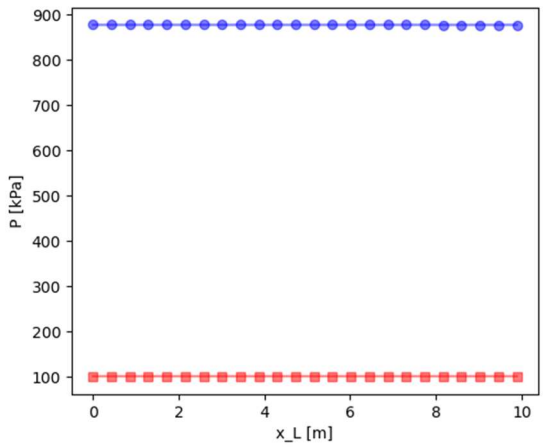

T [C]
x_L [m]

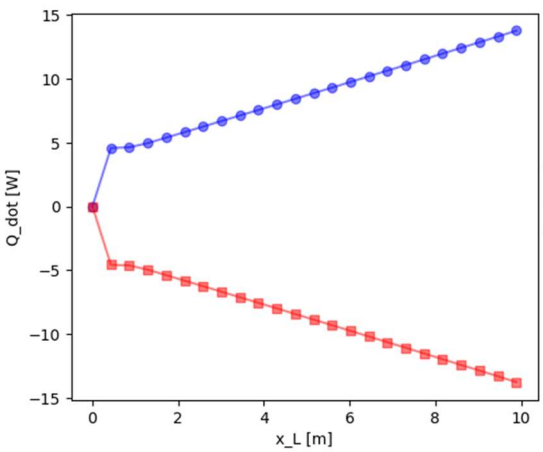

P [kPa]
x_L [m]

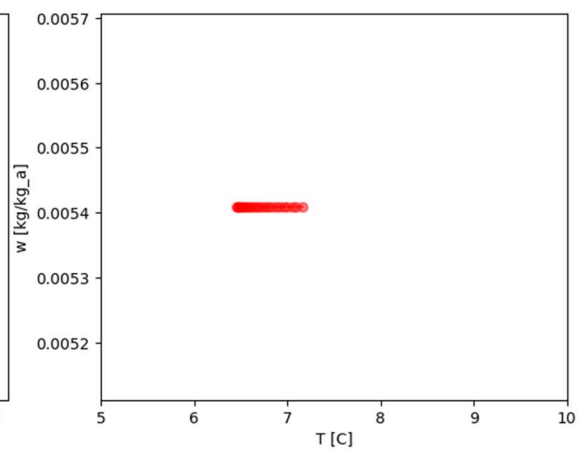

Q_dot [W]
x_L [m]

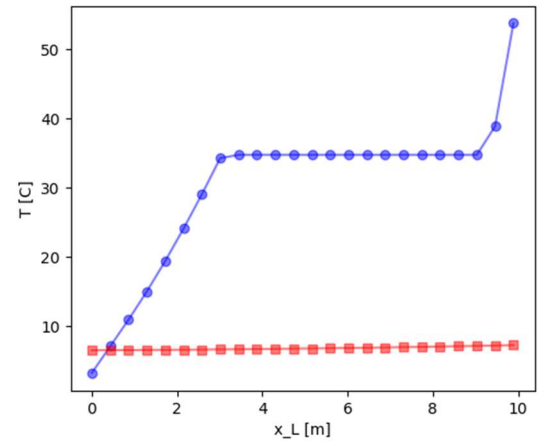

w [kg/kg_a]
T [C]

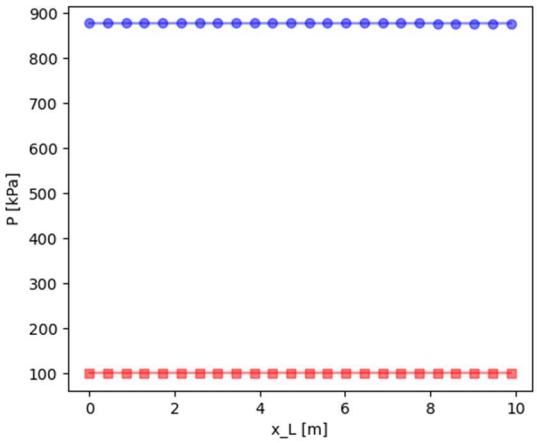

T [C]
x_L [m]

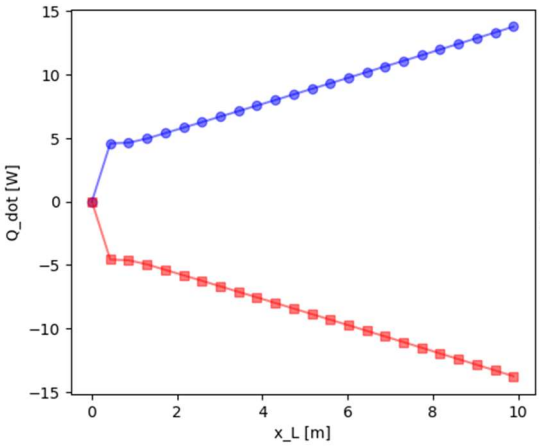

P [kPa]
x_L [m]

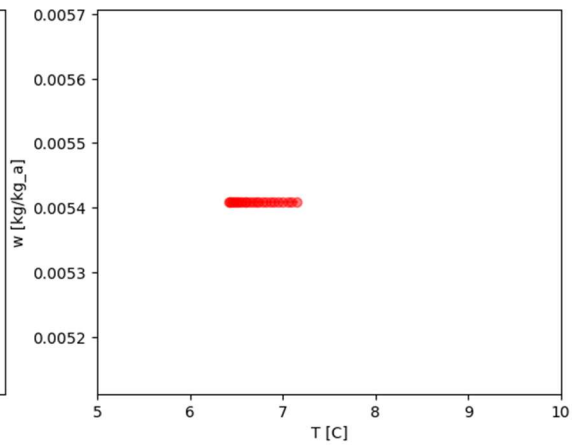

Q_dot [W]
x_L [m]

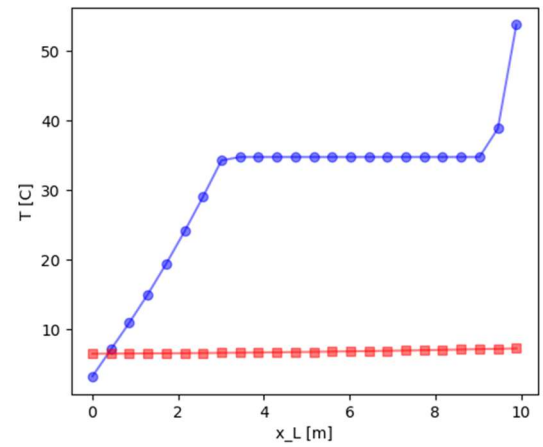

w [kg/kg_a]
T [C]

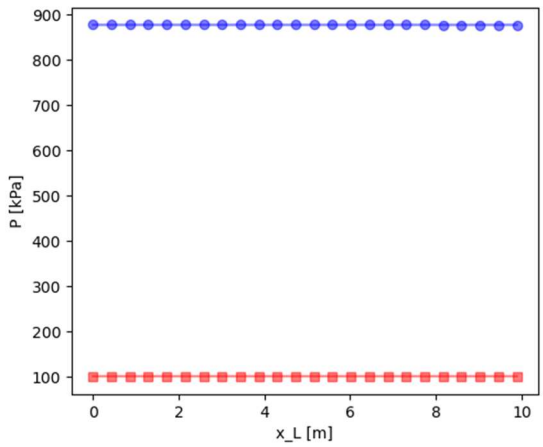

T [C]
x_L [m]

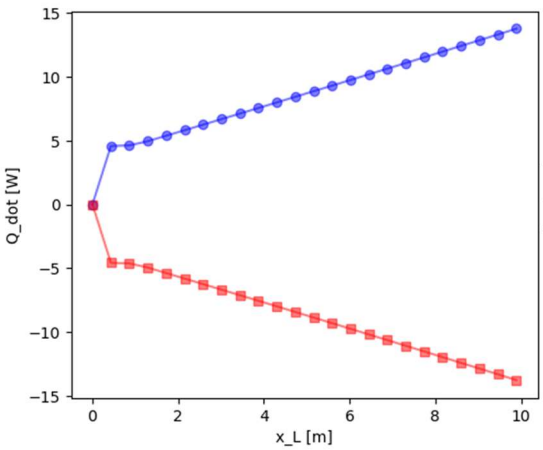

P [kPa]
x_L [m]

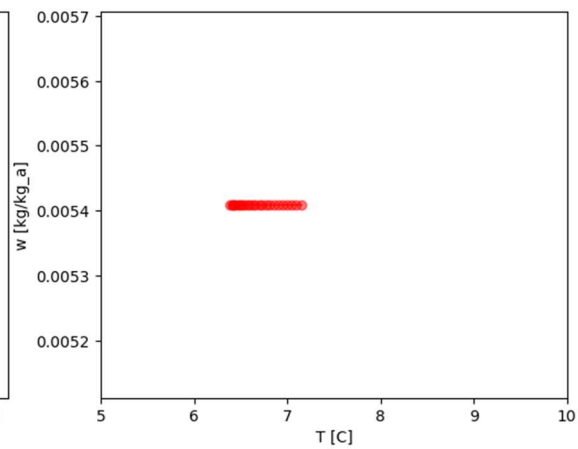

Q_dot [W]
x_L [m]

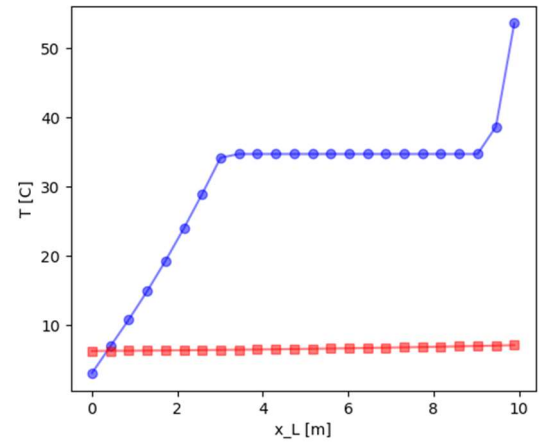

w [kg/kg_a]
T [C]

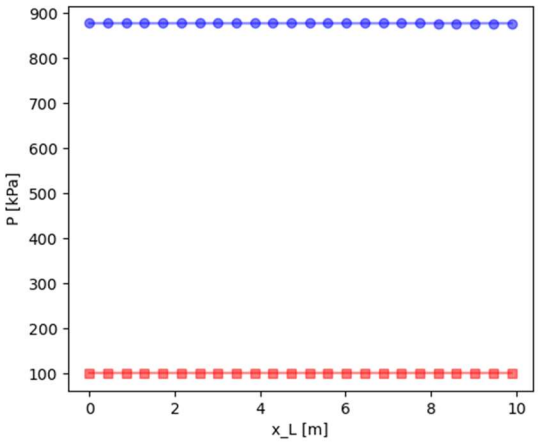

T [C]
x_L [m]

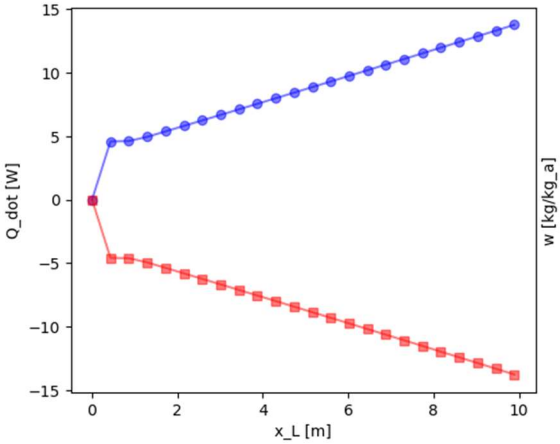

P [kPa]
x_L [m]

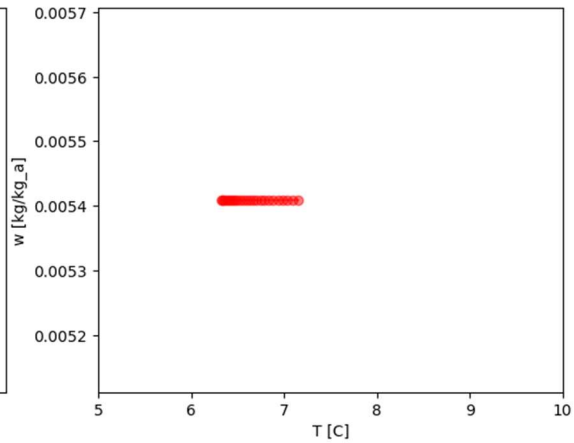

Q_dot [W]
x_L [m]

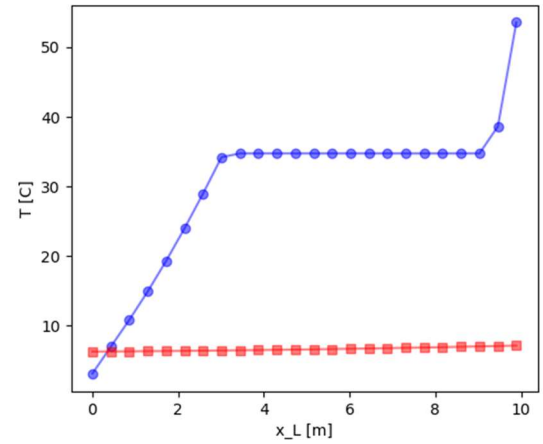

w [kg/kg_a]
T [C]

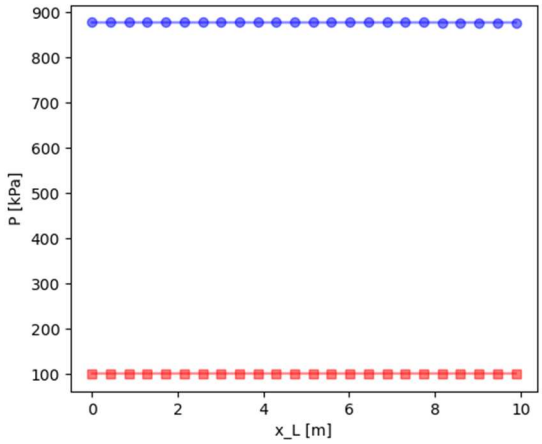

T [C]
x_L [m]

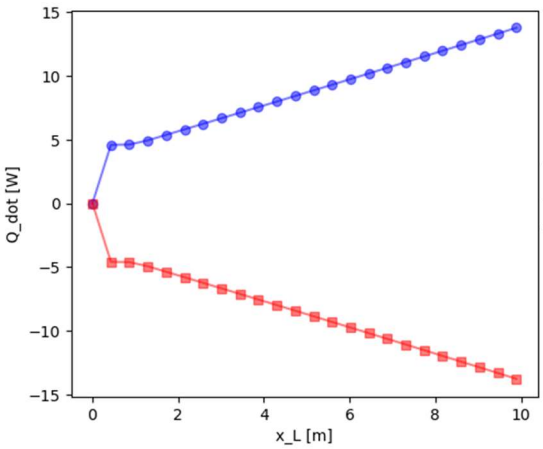

P [kPa]
x_L [m]

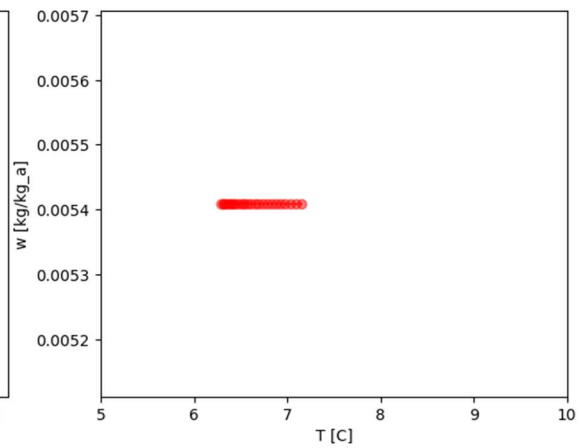

Q_dot [W]
x_L [m]

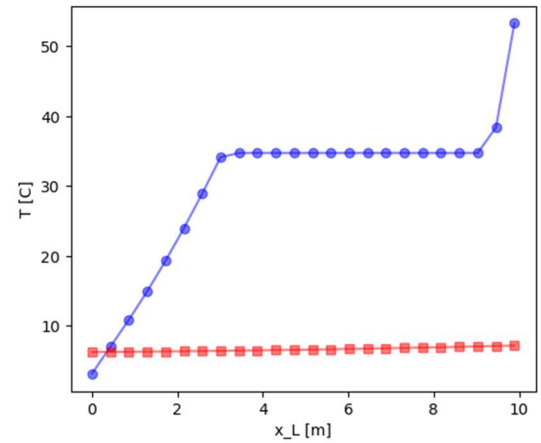

w [kg/kg_a]
T [C]

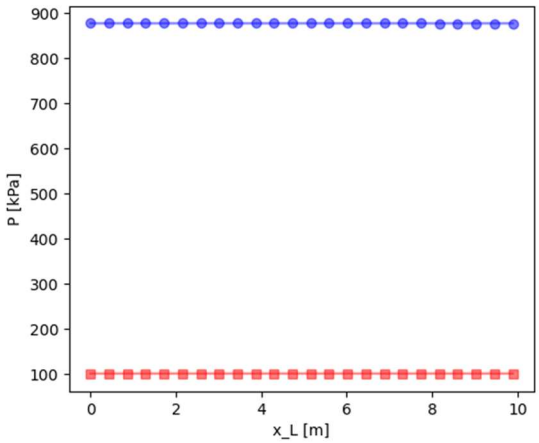

T [C]
x_L [m]

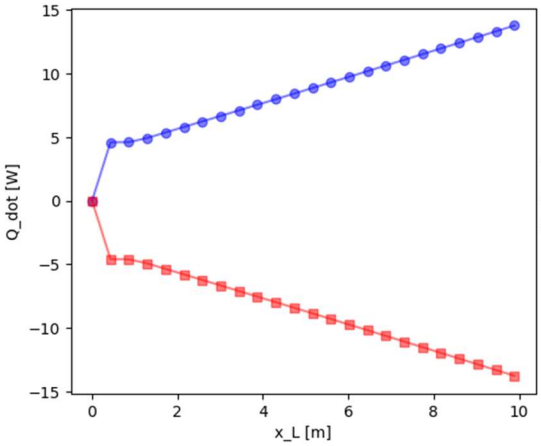

P [kPa]
x_L [m]

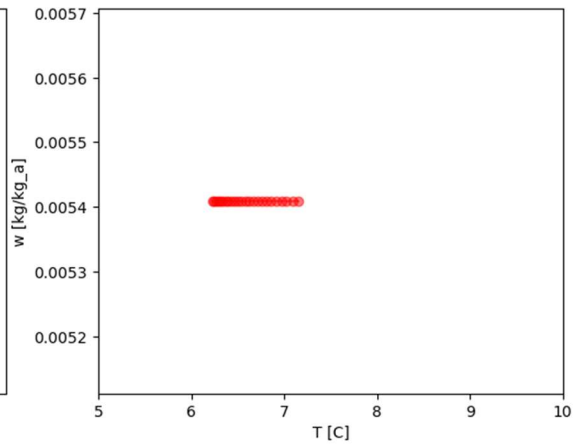

Q_dot [W]
x_L [m]

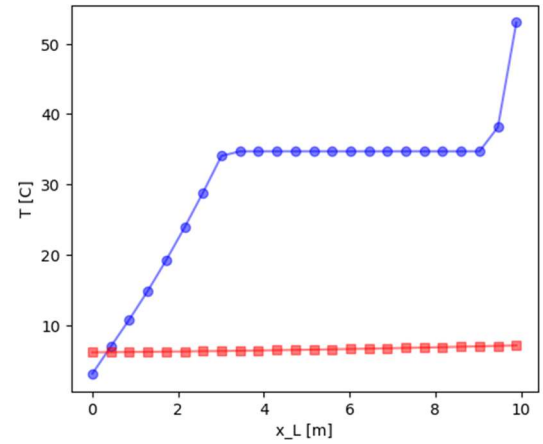

w [kg/kg_a]
T [C]

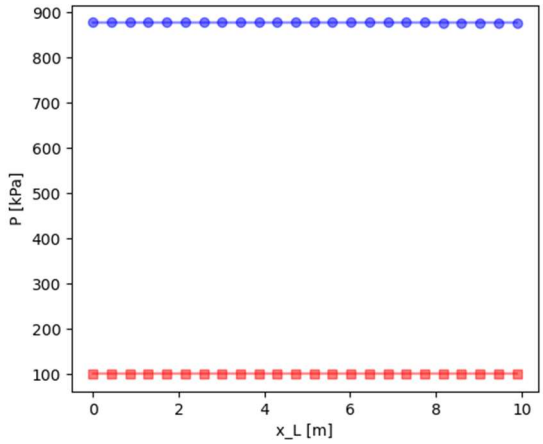

T [C]
x_L [m]

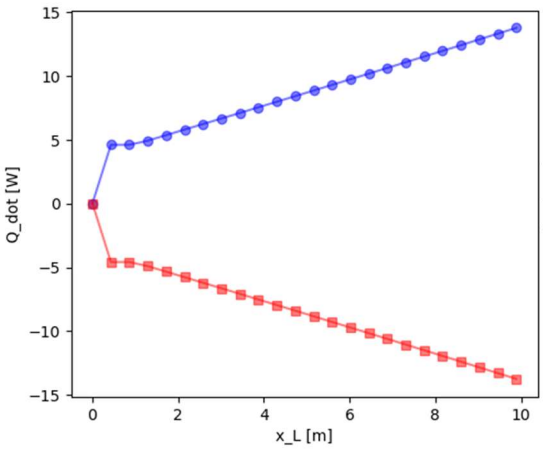

P [kPa]
x_L [m]

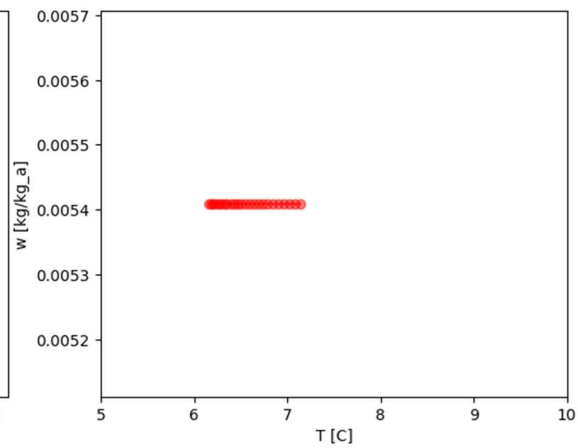

Q_dot [W]
x_L [m]

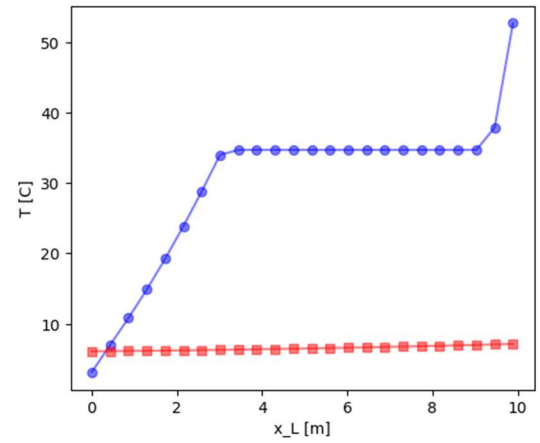

w [kg/kg_a]
T [C]

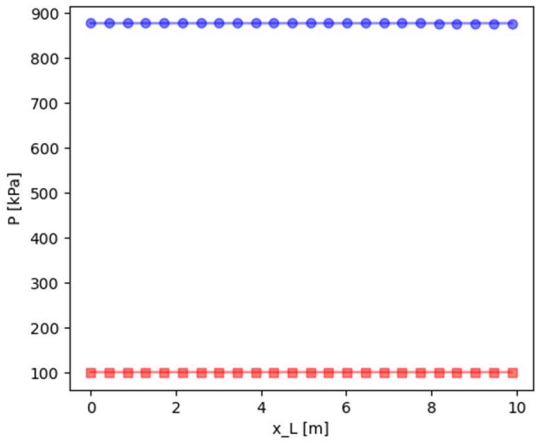

T [C]
x_L [m]

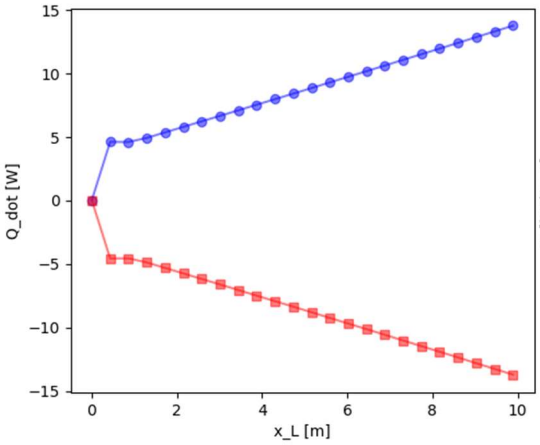

P [kPa]
x_L [m]

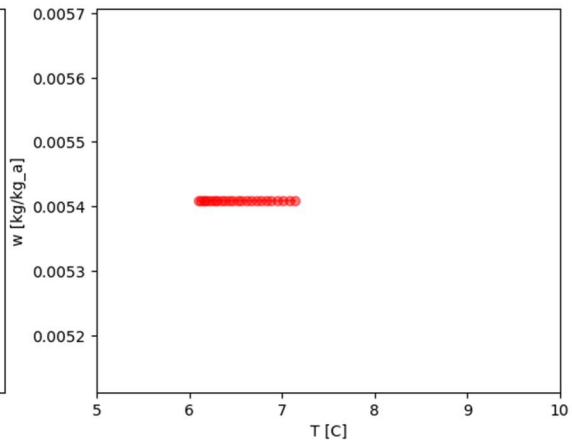

Q_dot [W]
x_L [m]

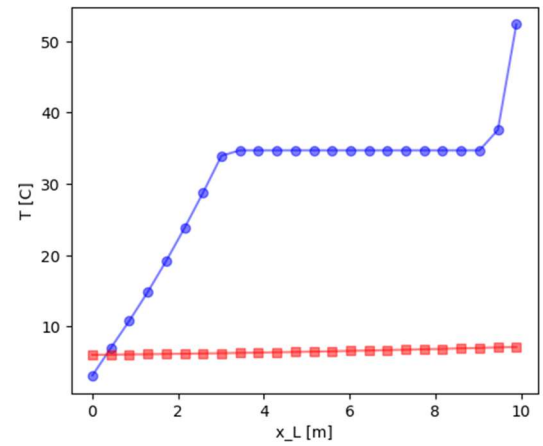

w [kg/kg_a]
T [C]

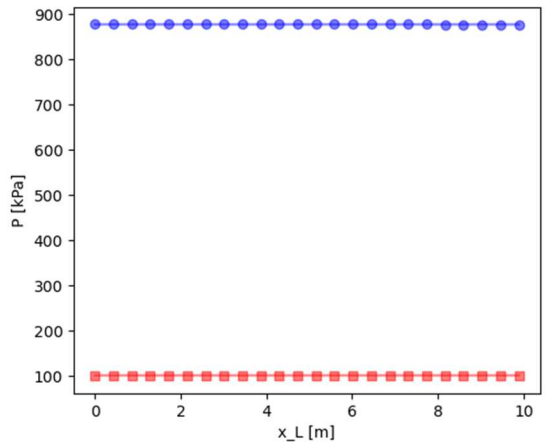

T [C]
x_L [m]

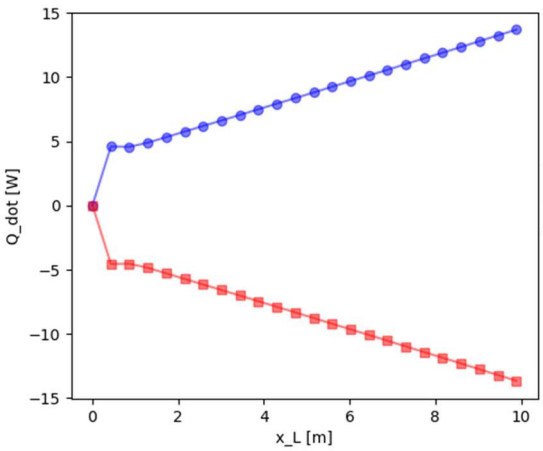

P [kPa]
x_L [m]

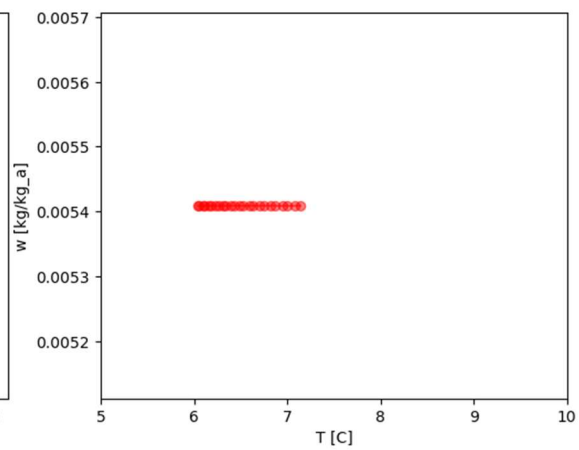

Q_dot [W]
x_L [m]

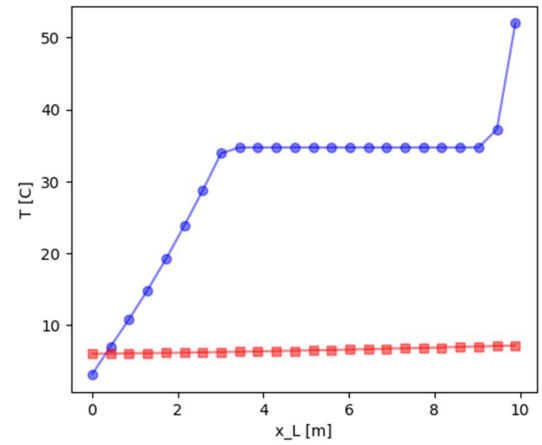

w [kg/kg_a]
T [C]

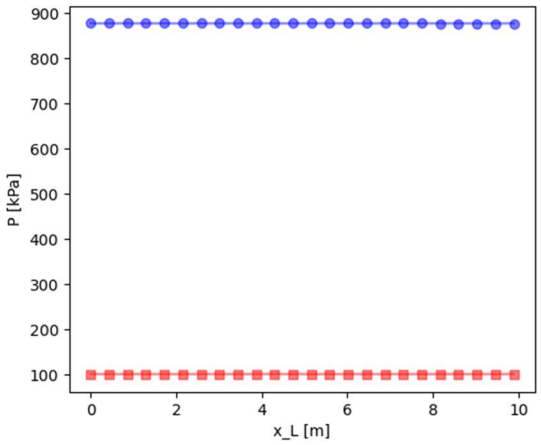

T [C]
x_L [m]

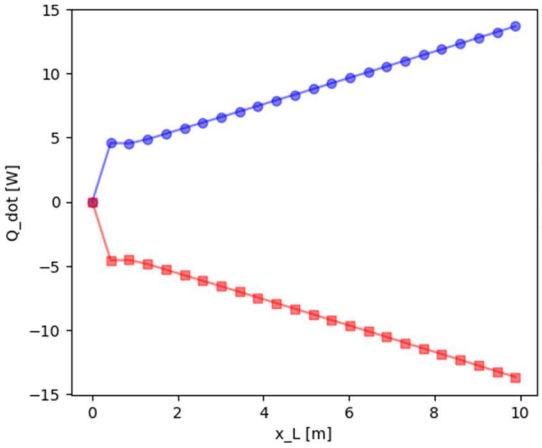

P [kPa]
x_L [m]

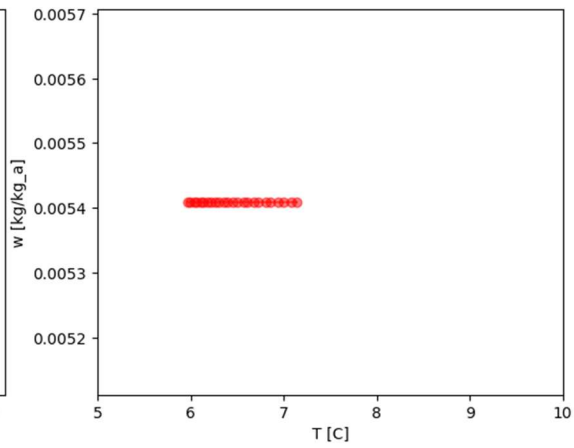

Q_dot [W]
x_L [m]

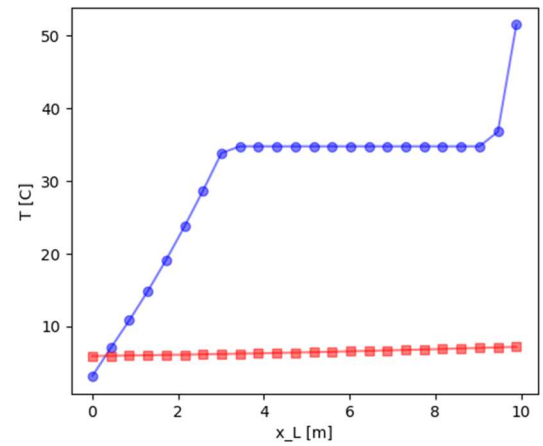

w [kg/kg_a]
T [C]

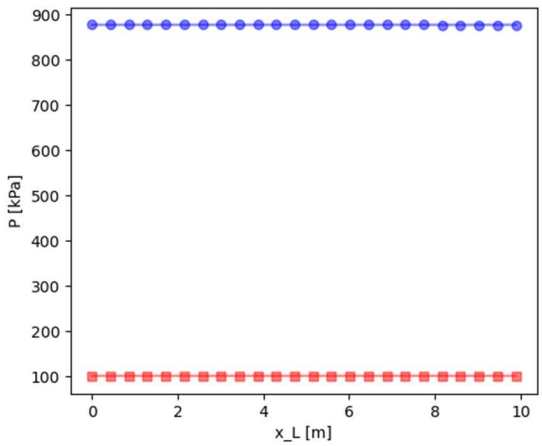

T [C]
x_L [m]

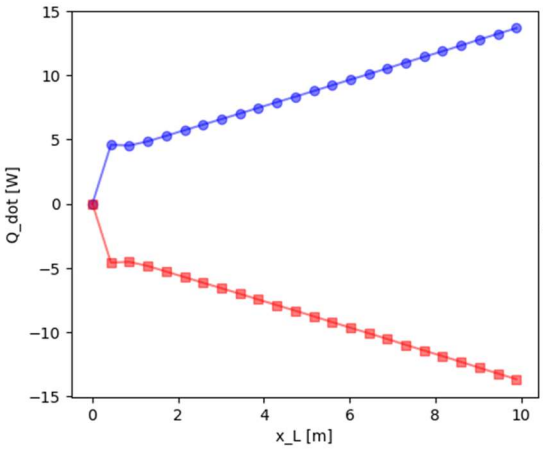

P [kPa]
x_L [m]

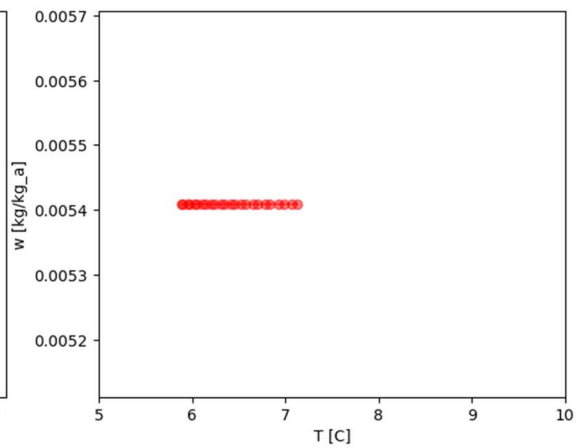

Q_dot [W]
x_L [m]

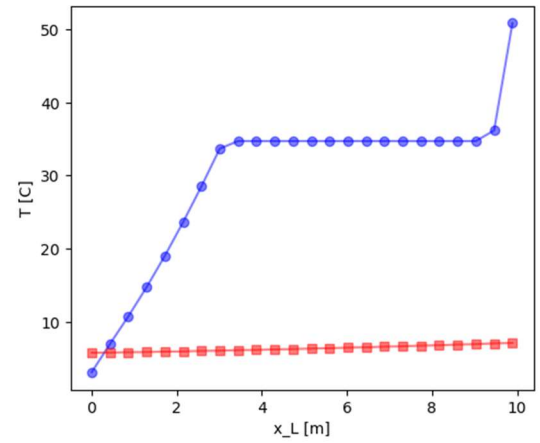

w [kg/kg_a]
T [C]

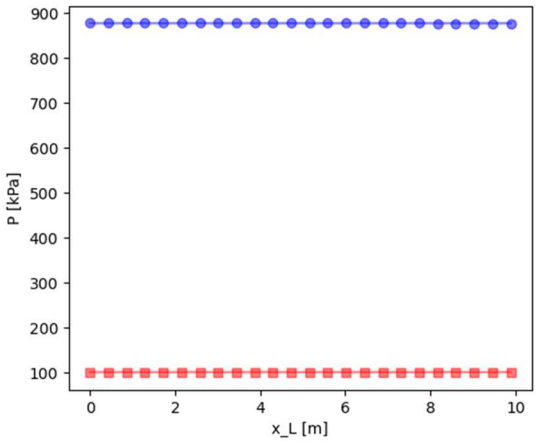

T [C]
x_L [m]

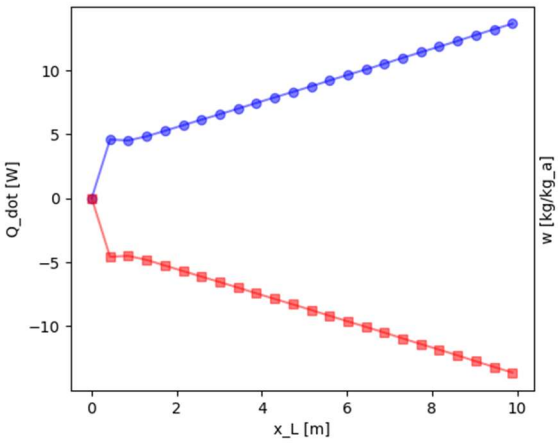

P [kPa]
x_L [m]

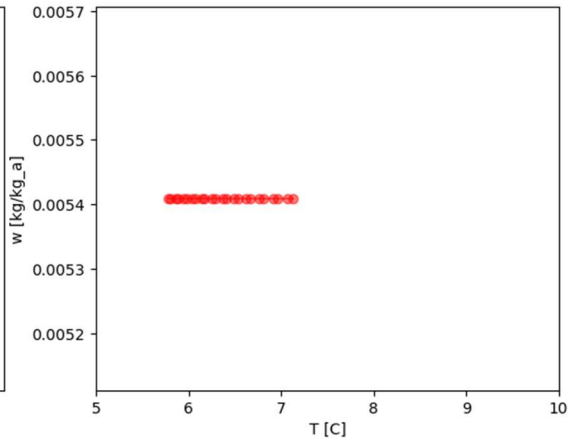

Q_dot [W]
x_L [m]

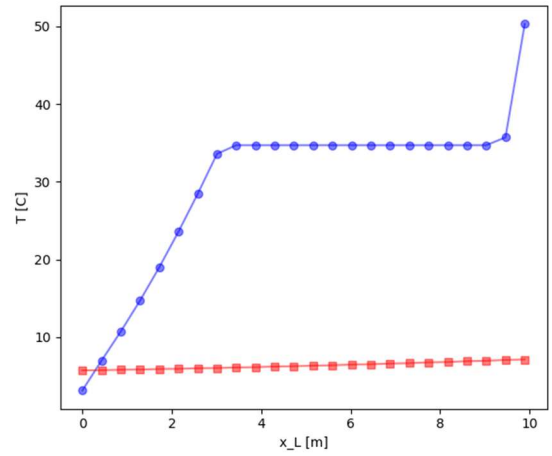

w [kg/kg_a]
T [C]

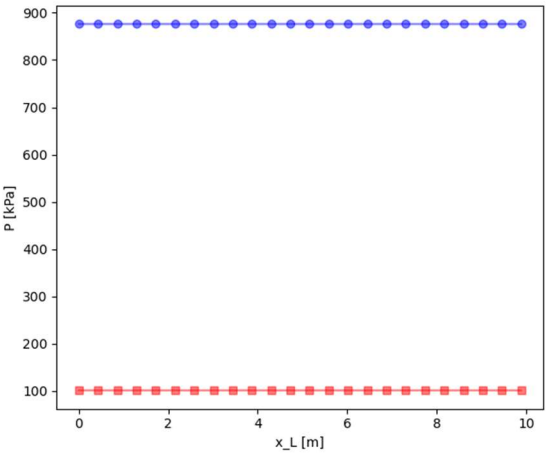

T [C]
x_L [m]

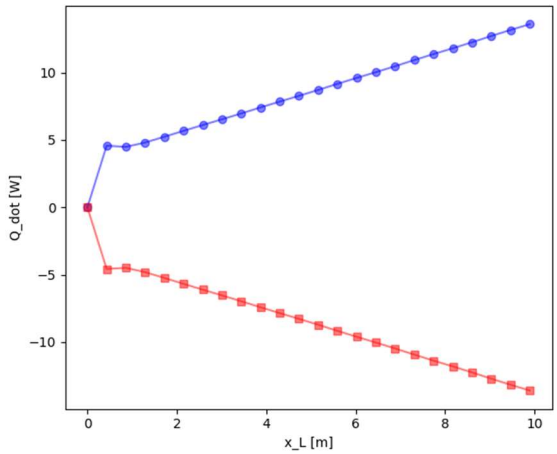

P [kPa]
x_L [m]

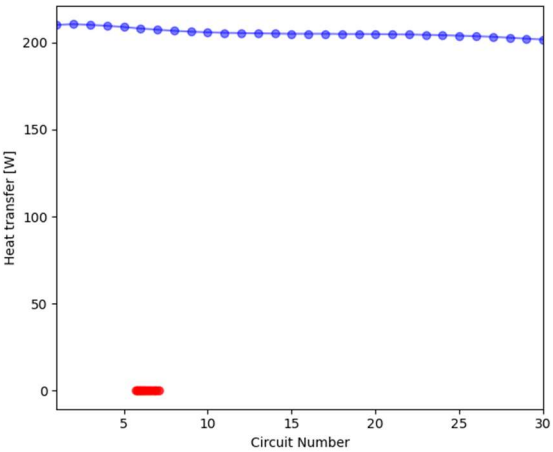

Q_dot [W]
x_L [m]


w [kg/kg_a]
T [C]


T [C]
x_L [m]


P [kPa]
x_L [m]


Q_dot [W]
x_L [m]


Heat transfer [W]
Circuit Number

Appendix C: Heat pump system model

This section shows the Python model that represents a residential heat pump system in which the modelled fin and tube heat exchanger are integrated. The code was used for all operating conditions of the heat pump systems and the one shown in this section is for operating condition 3.

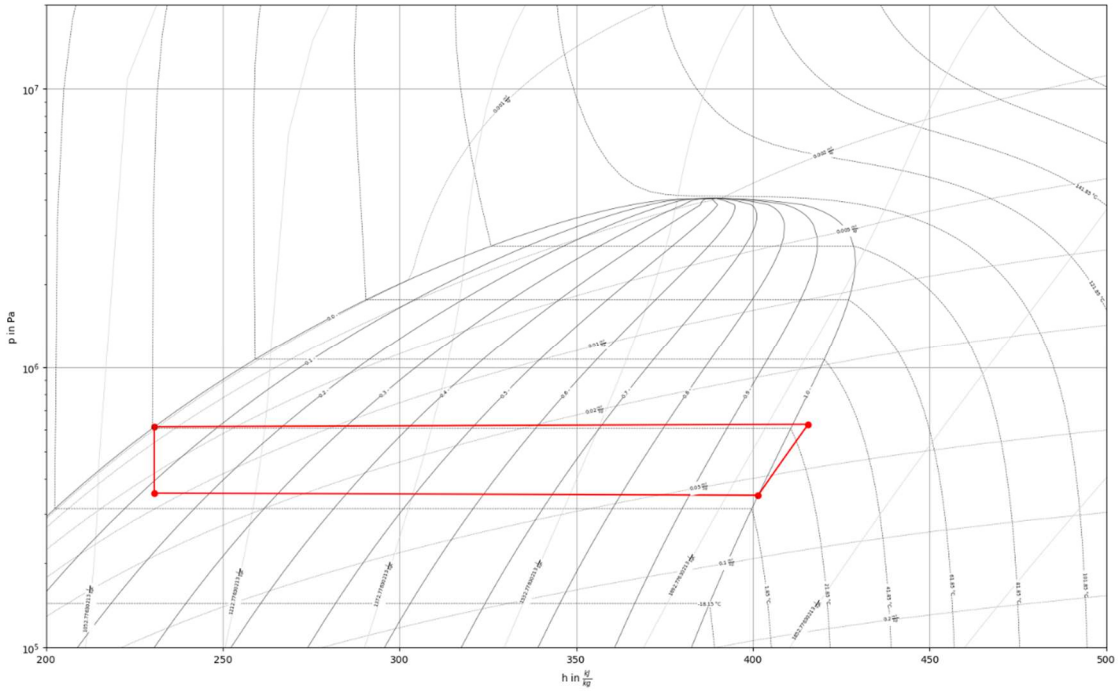
```
"""
Heat pump cycle model.ipynb
Written by
Sehobai Sehobai
August 2023
"""
"""
# This model uses TESPy modules to generate heat pump model
# Francesco Witte & Ilja Tuschy: 21 March 2020: 'TESPy:
Thermal Engineering Systems in Python',
The Journal of open source softwares,
"""


from tespy.networks import Network
from tespy.components import (
    CycleCloser, Compressor, Valve, HeatExchangerSimple
)
from tespy.connections import Connection
from fluprodia import FluidPropertyDiagram
import numpy as np

# 1.Set up a plant or cycle

# create a network object with R134a as fluid
fluid_list = ['R410a']
my_plant = Network(fluids=fluid_list)

# set the units system for temperatures to °C, pressure to Pa, enthalpy to kJ/kg and mass flow rate
to kg/s
my_plant.set_attr(T_unit='C', p_unit='Pa', h_unit='kJ / kg',m_unit = 'kg / s')

# 2.Set up components which makes up a plant or cycle

#closing the cycle
cc = CycleCloser('cycle closer')

# assigning a condenser
co = HeatExchangerSimple('condenser')

# setting an evaporator
ev = HeatExchangerSimple('evaporator')

#assigning an expansion valve
va = Valve('expansion valve')

#assigning a compressor
cp = Compressor('compressor')

# 3. Establish connections between components

# connections of heat pump
c1 = Connection(cc, 'out1', ev, 'in1', label='1') #evaporator inlet
c2 = Connection(ev, 'out1', cp, 'in1', label='2') #evaporator outlet and inlet to compressor
c3 = Connection(cp, 'out1', co, 'in1', label='3') #compressor outlet and inlet to condenser
c4 = Connection(co, 'out1', va, 'in1', label='4') #condenser outlet and inlet to the expansion
valve
c0 = Connection(va, 'out1', cc, 'in1', label='0') #expansion valve outlet

# Addition of all connections to your network
my_plant.add_conns(c1, c2, c3, c4, c0)
```

```
# 4. Set components and connections parameters

co.set_attr(pr=0.98) #condenser pressure ratio (pr = p_out/p_in) and heat transferred to the
environment
ev.set_attr(pr=0.9829) #evaporator pressure ratio (pr = p_out/p_in)
cp.set_attr(eta_s=0.85) #compressor efficiency

c1.set_attr(m=0.02639) #setting mass flow rate at point 1 (evaporator inlet)
c2.set_attr(T=34.6, x=1.0, fluid={'R410a': 1}) #setting refrigerant temperature and quality at
evaporator outlet
c4.set_attr(T=22.30, x=0.0) #setting refrigerant temperature and quality at the condenser outlet

# 5.Calculations

my_plant.solve(mode='design')
my_plant.print_results()

print(f'COP = {abs(co.Q.val) / cp.P.val}')

# 6. Pressure vs enthalpy (logph) plot

#saving results into the directory
result_dict = {}
result_dict.update({ev.label : ev.get_plotting_data()[1]})
result_dict.update({cp.label : cp.get_plotting_data()[1]})
result_dict.update({co.label : co.get_plotting_data()[1]})
result_dict.update({va.label : va.get_plotting_data()[1]})

#set up the diagram
diagram = FluidPropertyDiagram('R410a')
diagram.set_unit_system(T='°C', p='Pa', h='kJ/kg')

#Setting diagram limits and calculating isolines
for key, data in result_dict.items():
        result_dict[key]['datapoints'] = diagram.calc_individual_isoline(**data)

diagram.set_limits(x_min=200, x_max=500, y_min=1e5, y_max=2e7)
diagram.calc_isolines()
diagram.draw_isolines('logph')

for key in result_dict.keys():
    datapoints = result_dict[key]['datapoints']
    diagram.ax.plot(datapoints['h'],datapoints['p'], color='#ff0000')
    diagram.ax.scatter(datapoints['h'][0],datapoints['p'][0], color='#ff0000')
diagram.save('R410a_logph.svg')
```

RESULTS:

```
iter    | residual | massflow | pressure | enthalpy | fluid
--------+----------+----------+----------+----------+---------
1       | 4.84e+05 | 2.78e+00 | 2.30e+06 | 5.33e+05 | 0.00e+00
2       | 1.34e+05 | 1.73e-16 | 2.42e+06 | 3.18e+05 | 0.00e+00
3       | 2.56e+04 | 0.00e+00 | 4.58e+05 | 9.65e+03 | 0.00e+00
4       | 3.26e+03 | 0.00e+00 | 5.06e+04 | 5.33e+02 | 0.00e+00
5       | 6.70e+01 | 0.00e+00 | 1.19e+03 | 1.62e+01 | 0.00e+00
6       | 3.43e-02 | 0.00e+00 | 5.91e-01 | 7.37e-03 | 0.00e+00
7       | 6.98e-07 | 0.00e+00 | 1.86e-02 | 1.79e-03 | 0.00e+00
--------+----------+----------+----------+----------+---------
Total iterations: 7, Calculation time: 0.1 s, Iterations per second: 113.25
##### RESULTS (HeatExchangerSimple) #####
+------------+-----------+----------+----------+-----+-----+------+------+--------+
|            |         Q |       pr |     zeta |   D |   L |   ks |   kA |   Tamb |
|------------+-----------+----------+----------+-----+-----+------+------+--------|
| condenser  | -4.88e+03 | 9.80e-01 | 1.28e+09 | nan | nan |  nan |  nan |    nan |
| evaporator |  4.51e+03 | 9.83e-01 | 3.27e+08 | nan | nan |  nan |  nan |    nan |
+------------+-----------+----------+----------+-----+-----+------+------+--------+
##### RESULTS (CycleCloser) #####
+--------------+------------------+-------------------+
|              |   mass_deviation |   fluid_deviation |
|--------------+------------------+-------------------|
| cycle closer |         0.00e+00 |          0.00e+00 |
```

```
+--------------+------------------+-------------------+
##### RESULTS (Valve) #####
+-----------------+----------+----------+
|                 |       pr |     zeta |
|-----------------+----------+----------|
| expansion valve | 5.80e-01 | 1.09e+11 |
+-----------------+----------+----------+
##### RESULTS (Compressor) #####
+------------+----------+----------+----------+--------+
|            |        P |    eta_s |       pr |   igva |
|------------+----------+----------+----------+--------|
| compressor | 3.74e+02 | 8.50e-01 | 1.79e+00 |    nan |
+------------+----------+----------+----------+--------+
##### RESULTS (Connection) #####
+----+-----------+-----------+-----------+-----------+
|    |         m |         p |         h |         T |
|----+-----------+-----------+-----------+-----------|
|  1 | 2.639e-02 | 3.557e+05 | 2.307e+02 | 5.497e+00 |
|  2 | 2.639e-02 | 3.497e+05 | 4.015e+02 | 5.000e+00 |
|  3 | 2.639e-02 | 6.260e+05 | 4.156e+02 | 2.728e+01 |
|  4 | 2.639e-02 | 6.135e+05 | 2.307e+02 | 2.230e+01 |
|  0 | 2.639e-02 | 3.557e+05 | 2.307e+02 | 5.497e+00 |
+----+-----------+-----------+-----------+-----------+
COP = 5.500045226097071
```




```
#Ts Plotting

#saving results into the directory
result_dict = {}
result_dict.update({ev.label : ev.get_plotting_data()[1]})
result_dict.update({cp.label : cp.get_plotting_data()[1]})
result_dict.update({co.label : co.get_plotting_data()[1]})
result_dict.update({va.label : va.get_plotting_data()[1]})

#set up the diagram
diagram = FluidPropertyDiagram('R410a')
diagram.set_unit_system(T='°C', p='Pa', h='kJ/kg')

#Setting diagram limits and calculating isolines
for key, data in result_dict.items():
        result_dict[key]['datapoints'] = diagram.calc_individual_isoline(**data)

diagram.set_limits(x_min=1000, x_max=2000, y_min=0, y_max=100)
diagram.calc_isolines()
diagram.draw_isolines('Ts')
```

```
for key in result_dict.keys():
    datapoints = result_dict[key]['datapoints']
    diagram.ax.plot(datapoints['s'],datapoints['T'], color='#ff0000')
    diagram.ax.scatter(datapoints['s'][0],datapoints['T'][0], color='#ff0000')
diagram.save('R410a_logph.svg')
```

RESULTS: