NATURAL REFRIGERANTS IN HEAT PUMPS UPGRADING WASTE HEAT FOR DISTRICT HEATING IN FINLAND

Lappeenranta–Lahti University of Technology LUT

Degree Programme in Energy Technology, Bachelor's thesis

2025

Darius Stanisor

Examiners: Professor Antti Uusitalo

Professor Xiang Gou

Abstract

Lappeenranta–Lahti University of Technology LUT

LUT School of Energy Systems

Degree Programme in Energy Technology

In co-operation with partner university: HEBEI University of Technology

Darius Stanisor

Natural Refrigerants in Heat Pumps Upgrading Waste Heat for District Heating in Finland

Bachelor's thesis

2025

25 pages, 4 figures, 5 tables and 2 appendices

Examiners: Professor Antti Uusitalo and Professor Xiang Gou

Keywords: district heating, heat pump, refrigerant, natural refrigerant, waste heat

As the pressure of climate change increases, countries are looking to decarbonize their systems. Finland's energy industry is its largest polluter, and district heating (DH) its most popular residential heating solution. As DH remains largely fossil-fueled, lower environmental impact alternatives are being considered. One option is replacing DH power plants with heat pumps using industrial waste heat, but the environmental effects of the refrigerants used in the heat pump cycle also need to be considered in order to ensure sustainable operation. Natural refrigerants are the strongest contender for this position.

This study defined a cascade heat pump model and employed an iterative optimization algorithm to assess the performance of various refrigerant combinations across a range of heat source and heat sink temperatures. The heat source was an industrial waste heat stream and the heat sink was a Finnish DH supply line. The most common types of industrial streams in Finland containing waste heat that can be used for upgrading into district heat were found to come from the pulp and paper industry and food and beverage industry, with the selected temperature between 10 and 65°C. The typical Finnish district heat supply line temperature was found to range from 65 to 115°C.

The results showed that the decrease in performance when changing a synthetic refrigerant like R245fa with a natural refrigerant such as R600 is practically negligible, while achieving a significant reduction in global warming potential (GWP). Water (R718), ammonia (R717), and n-butane (R600) were the best performing natural refrigerants for the chosen system.

Symbols and abbreviations

Latin characters

h	specific enthalpy	J/kg
p	pressure	Pa
Q	heat	J
q	heat rate	W
q_m	mass flow rate	kg/s
s	specific entropy	J/(kgK)
T	temperature	K
W	work	J
X	steam quality	

Superscripts

cond	condenser
comp	compressor
evap	evaporator
sink	heat sink
source	heat source

Abbreviations

ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
CHP	combined heat and power
COP	coefficient of performance
CFC	chlorofluorocarbons
DH	district heating
HCFC	hydrochlorofluorocarbons
HFC	hydrofluorocarbons
HTC	high temperature cycle
LTC	low temperature cycle

Table of contents

List of Figures

List of Tables

1 Introduction

Finland aims to become carbon neutral by the year 2035. With 28.6 million tonnes of CO2 equivalent in 2023, the Finnish energy industry accounted for 70% of the country's greenhouse gas emissions, far more than all of the other sectors combined (Official Statistics of Finland OSF, 2025). This means that, for Finland to achieve its ambitious goal, major changes will have to be made in the way it produces its energy. District heating (DH) is currently the most commonly used method of heating in Finland and still expanding. Most Finnish DH networks today produce the heat their customers use with multiple power plants, which can be either combined heat and power (CHP) producers or dedicated heat producers (Finnish Energy, 2025a). These power plants rely mostly on non-renewable resources, such as fossil fuels, leading to environmental damage. Another option quickly gaining popularity for DH production in Finland is electric boilers, but their efficiency is often inferior to other electric options. One promising alternative to polluting power plants and inefficient boilers is using heat pumps to upgrade waste heat from industrial processes neighboring the DH network. If used with environmentally-friendly refrigerants and sustainably-produced electricity, heat pumps could significantly reduce the greenhouse gas emissions of the DH sector and bring down the environmental damage caused by energy production.

Previous studies have outlined some of the benefits and drawbacks of using natural refrigerants in heat pumps, their properties and feasibility from several perspectives. Bolaji & Huan (2013) presented natural refrigerants as the ideal long-term solution for heat pumps and refrigeration systems, having zero ozone depletion potential (ODP) and low to no global warming potential (GWP). Abas et al. (2018) discussed the Paris Climate Agreement and the Kigali amendment to the Montreal Protocol, which drive a transition away from high-GWP refrigerants. Heredia-Aricapa et al. (2020) provided a review of low-GWP refrigerant mixtures and found that most of those with similar energy performance to the high-GWP refrigerants they were supposed to replace still exceeded limits set by environmental regulations. Calm's (2008) historical review divided the evolution of refrigerants into four generations and points to a renewed interest in natural refrigerants due to environmental concerns, after having been first phased out around the 1930's by fluorochemicals, which presented better safety and durability. Jiang et al. (2022) noted that natural refrigerants are also becoming more common in high-temperature heat pump applications and are already the prevalent choice in research on the subject. In their survey on large-scale electric heat pumps used for DH in Europe, David et al. (2017) found that about 19% of the systems that were part of the survey already use natural refrigerants, namely ammonia and carbon dioxide.

Heat pumps in DH is also a well-studied topic. Ommen, Markussen & Elmegaard (2014) studied the optimal heat to power distribution of a combined heat and power plant, comparing five generic configurations of heat pumps in DH systems with varying network temperatures and fuels. The transition to heat pump based DH has also been analyzed from an economic perspective, such as that offered by Bach et al. (2016), who used the Balmorel model to determine optimal dispatch and assess the competitiveness of heat pumps connected to different parts of the network. Heat pumps used to upgrade waste heat from industrial processes for DH have been studied by Uusitalo et al. (2020-05-14), who implemented a thermodynamic analysis alongside compressor design analysis to find the refrigerant allowing for the most efficient configuration.

Although many studies covered natural refrigerants, heat pumps upgrading waste heat and heat pumps used for DH, no study covered the optimization of a heat pump upgrading industrial waste heat for a DH system while accounting for environmental factors. Finding the optimal refrigerant combination for both energy efficiency and minimal GWP is of critical importance if the heat pump is replacing the previous DH system because of environmental concerns.

The main research questions guiding the work of this thesis are: How much does changing the working fluid of a heat pump from a synthetic refrigerant to a natural refrigerant affect the performance of the system? How should a natural refrigerant be selected when considering such a change? In order to answer the questions, a cascade heat pump upgrading industrial waste heat for district heating will be defined and the typical operating conditions for such a system located in Finland will be identified. The performance of the heat pump will be modeled and then a series of refrigerant combinations will be tested for various operating temperatures with the objective of finding the best performing natural refrigerants.

2 Background

To find the best natural refrigerants for use in the heat pump model, a list of possible refrigerant candidates was first developed. To design the heat pump model, it was first necessary to determine the operating conditions of a HP upgrading waste heat for DH.

2.1 Refrigerants

Refrigerants are the working fluids used by heat pumps to carry the heat from a heat source to a heat sink. They have a large impact on heat pump performance and design, but also on its environmental effects. As pointed out by Bolaji & Huan (2013), refrigerants leak out, either in small quantities during operation, or, more significantly, during maintenance and disposal. Once they are released in the atmosphere, they can contribute to climate change, acting as a greenhouse gas. This contribution is quantified through the global warming potential (GWP) of the refrigerant, which is a measure of how well it absorbs infrared radiation relative to carbon dioxide, which has a GWP of 1. GWP is evaluated for a given time period, usually 100 years. Another metric used to assess the environmental friendliness of a refrigerant is the ozone depletion potential (ODP). ODP is a relative metric which compares how damaging a refrigerant is to the ozone layer compared to R-11, or trichlorofluoromethane, which is used as a benchmark with an ODP of 1 (UNEP et al., 2006). The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) also provides a useful classification for refrigerants in their Standard 34, which assigns refrigerants a short code comprised of two parts. The first part divides the toxicity into class A, for refrigerants of lower toxicity, and class B, for those of higher toxicity. The second part of the code assesses flammability on a scale of 4 levels, namely 1, 2L, 2 and 3, with 1 meaning non-flammable and 3 meaning highly flammable, with a heat of combustion greater than or equal to 19,000 kJ/kg (Bell, Domanski, et al., 2019).

Properties like GWP and ODP matter today, but that was not always the case. Calm (2008) divides the evolution of refrigerants into four generations. The first generation appeared roughly around the 1830s, after heat pumps first entered wide-spread use and only prioritized functionality, using even hazardous or flammable refrigerants for the sake of efficiency. At this stage, there was a lot of experimentation and the first accounts of heat pumps using natural refrigerants appeared around this time. Around the 1930s, the focus shifted to safety and durability, thus starting the second generation, which was dominated by chlorofluorocarbons (CFCs) and hydrochlorofluorocarbons (HCFCs). Years of producing CFCs and releasing it into the atmosphere resulted in serious damage to the Earth's ozone layer, prompting the Vienna convention in 1987 and the resulting Montreal protocol, which banned CFCs and marked the beginning of the third generation, which used hydrofluorocarbons (HFCs) alongside HCFCs. The fourth and current generation of refrigerants begun around the year 2011, when the European Parliament banned fluorochemicals, also termed “F-Gas” refrigerants. This generation is defined by a focus on low-GWP refrigerants, of which the natural refrigerants have proved the most promising so far. Despite the large selection of refrigerants that abide by these restrictions, many researchers agree that an ideal refrigerant fluid or mixture which can compete with the performance of previous popular choices is yet to be found.

Despite the many regulations and restrictions against fluorochemicals and high-GWP refrigerants, the most commonly used refrigerant in heat pumps is R134a, which David et al. (2017) found to be used by over 90% of the systems accounted for in the survey. According to the same survey, Heredia-Aricapa et al. (2020) and Jiang et al. (2022), other synthetic refrigerants in use today include R410a, and R245fa. Among natural refrigerants, ammonia is already a popular choice for large-scale industrial applications, hydrocarbons such as propane and isobutane are used in some domestic refrigerators, carbon dioxide is regaining popularity in air conditioning and refrigeration applications, while water has been used throughout the history of refrigerants (Bolaji & Huan, 2013). All previously mentioned refrigerants, along with a few others recommended by Gaur, Fitiwi & Curtis (2021) as possible replacements for R134a, were considered in the calculations presented in later sections, and they are all listed in Table 1 together with their relevant environmental properties. The properties are compiled from the works of Heredia-Aricapa et al. (2020), Bolaji & Huan (2013), Jiang et al. (2022), Gaur, Fitiwi & Curtis (2021) and Calm & Hourahan (2011).

Table 1: Refrigerant candidates for the heat pump model.

Refrigerant code	Common name	Chemical formula	GWP (100y)	ASHRAE-34
R134a	-	$C_2H_2F_4$	1300	A1
R410a	-	CH_2F_2 / C_2HF_5	2088	A1
R245fa	-	$C_3H_3F_5$	858	B1
R170	Ethane	C_2H_6	20	A3
R290	Propane	C_3H_8	20	A3
R600	n-Butane	C_4H_{10}	20	A3
R600a	Isobutane	C_4H_{10}	20	A3
R1270	Propylene	C_3H_6	20	A3
R717	Ammonia	NH_3	0	B2L
R744	Carbon dioxide	CO_2	1	A1
R718	Water	H_2O	0	A1

The ODP was omitted from Table 1 because all the refrigerants have an ODP of 0, as they are either natural substances or HFCs. From these properties alone, it appears the best choices are carbon dioxide and water, with an ASHRAE classification of A1 and little to no GWP.

2.2 District heating and industrial waste heat

The heat pump model is designed as a standard vapor-compression refrigeration cycle between a heat source and a heat sink. Since the heat pump is designed to upgrade waste for DH, the heat source is a residual fluid stream from an industrial process, while the heat sink is the hot water supply line of a DH network. There are various benefits to using heat pumps in DH, such as efficiency, since an electric boiler would require far more energy to get the water to the same temperature compared to a heat pump, as David et al. (2017) argues in their study. Østergaard & Andersen (2016) note that heat pumps allow tapping into low-temperature energy sources, which come at a lower cost than higher temperature ones and are also less exploited. Ommen, Markussen & Elmegaard (2014) further argue that, for existing CHP power plants, integrating heat pumps into the DH network effectively decouples CHP production constraints, allowing for more efficient operation of both heat and electricity production. Besides the efficiency and economic benefits, heat pumps in DH systems can efficiently utilize electricity from renewable sources and provide demand response to the power system due to the thermal inertia of buildings (Abas et al., 2018).

The temperatures of the heat source and heat sink can vary depending on a number of factors. According to Finnish Energy (2025b), Finnish district heating supply lines have temperatures between 65 and 115°C, which is the chosen range for the temperature of the heat sink. The heat source, chosen to be industrial waste heat, can span a wider range of temperatures, depending on the process it originates from, compared to the heat sink temperature. In the survey conducted by David et al. (2017), 28 heat pumps used industrial waste heat with temperatures ranging from 12 to 46°C. Rangnekar's study from 2023 identifies several industrial processes which generate waste heat carried by water with temperatures in the range of 40 to 90°C. As such, the minimum of the temperature range for the heat source of the heat pump model was chosen to be 10°C and the maximum 65°C. The maximum temperature for the heat source is the same as the minimum temperature of the heat sink because the heat sink temperature can vary throughout the year while the heat source does not, and the heat pump can only function while the heat sink is hotter than the heat source.

Creating a DH network running on upgraded industrial waste heat requires the DH network and industrial facility that produces the waste heat to have been constructed close to each other prior to the adding of the heat pump. Because of this, it is important to acknowledge the relevant industrial sectors for such an application. According to Rangnekar (2023), waste heat comes in one of three temperature ranges: below 100°C, between 100 and 200°C, and above 200°C. The Finnish industries which produce the most waste heat under 100°C are, according to the same study, the pulp and paper industry, with a relatively small fraction of waste heat produced in this temperature range but a far higher overall waste heat production rate than other industries, followed by the food and beverage industry, which produces more waste heat under 100°C than above, and is relatively prevalent in Finland as an industry. Although these industries are likely to produce waste heat at a desirable temperature for a DH application, the food and chemical industries are mentioned in the same study as generators of hot water in the 40-90°C range as a byproduct, which is even closer to the studied heat source temperature range.

3 Methods

To find the best heat pump configuration, a Matlab script was used to calculate the coefficient of performance (COP) of a cascade heat pump and loop through all possible configurations to find the natural refrigerants best suited for any given configuration. The script which calculates the COP of one heat pump configuration, with all input variables defined, is called the heat pump model. The heat pump model is thus the core of the iterative, brute-force search algorithm for finding the ideal refrigerants for any heat source and sink temperatures.

3.1 Heat pump model

The cascade heat pump is comprised of 2 identical cycles, the low temperature cycle (LTC) and high temperature cycle (HTC) , connected by a cascade heat exchanger. In the LTC evaporator, heat is removed from the heat source, while in the HTC condenser, heat is added to the heat source. The cascade heat exchanger connects the LTC condenser with the HTC evaporator. The compressor of each cycle uses work to increase the pressure and temperature of the refrigerant, while the expansion valve of each cycle allow the working fluid to decrease its pressure and temperature. For each cycle, 4 state points are defined as follows: state point 1 is after the evaporator and before the compressor; state point 2 is after the compressor and before the condenser; state point 3 is after the condenser and before the expansion valve; state point 4 is after the expansion valve and before the evaporator. The layout of these elements and state points are presented in Figure 1.

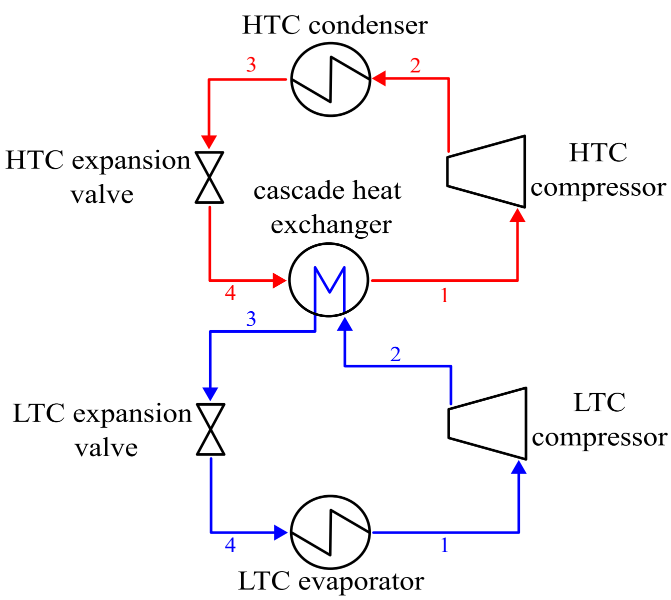


Figure 1: Heat pump layout.

An example configuration of this cycle, operating with water as refrigerants for both cycles, with the heat source at 40°C and heat sink at 100°C, is presented as plotted by the Matlab model in Figure 2.

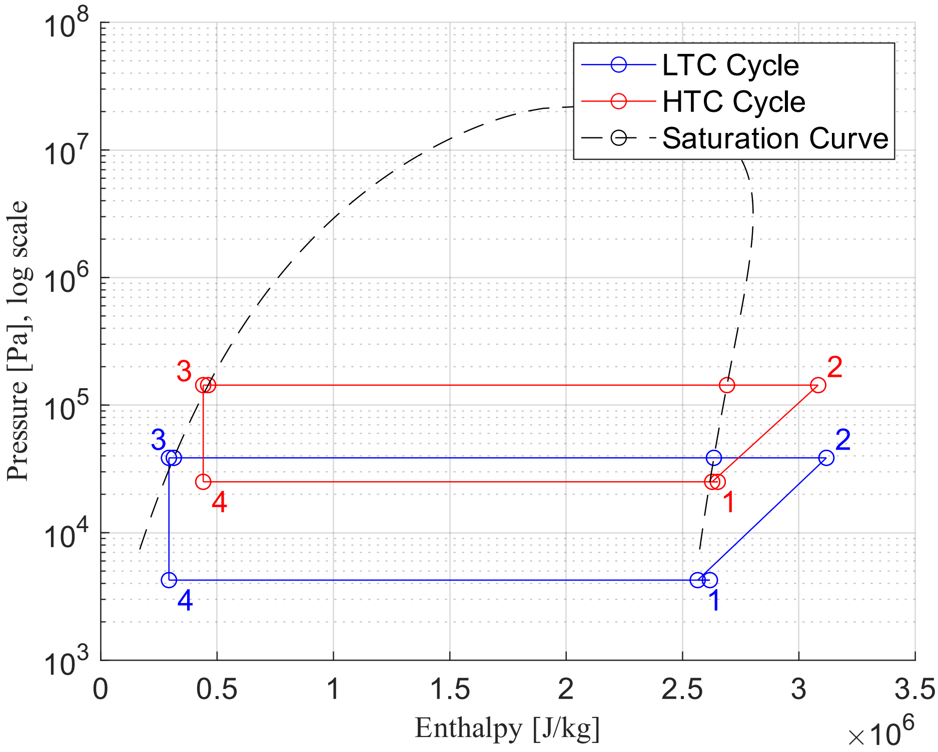


Figure 2: Log p-h diagram of cascade heat pump cycle.

Besides the choice of refrigerants and heat source and sink temperatures, there are also a number of standard process parameters used as input parameters for the calculations, which were chosen to be similar to those found in literature. The temperature of the LTC refrigerant in the LTC evaporator has to be lower than that of the heat source for heat to flow into the cycle; likewise, the temperature of the HTC refrigerant in the HTC condenser needs to be higher than that of the heat sink for heat to flow out of the cycle. For the heat exchanger to work, the temperature of the LTC fluid must be higher than that of the HTC fluid within the cascade heat exchanger. All of the three aforementioned temperature differences are assumed equal to 10 K for the calculations, same as the value used by Rangnekar (2023). The degree of superheating at the evaporator outlet and the degree of subcooling at the condenser outlet were both chosen to be 5 K, similar to the values used by Ommen, Markussen & Elmegaard (2014) and Heredia-Aricapa et al. (2020). The isentropic efficiencies of both the LTC and HTC compressors were considered to be 75%, which is just below the 80% used by Ommen, Markussen & Elmegaard (2014). The heat rate entering the LTC evaporator is also to find the refrigerant flow rates, and its value was chosen to be 1 MW, same as the one used by Uusitalo et al. (2020-05-14) in their calculations for a similar cycle.

Knowing the heat source and sink temperatures and the previously explained temperature differences, the LTC refrigerant temperature in the LTC evaporator can be calculated as:

$$T_{evap,LTC} = T_{source} - 10 \tag{1}$$

where $T_{evap,LTC}$ is the LTC refrigerant temperature in the LTC evaporator [K], T_{source} is the temperature of the heat source [K] and the difference between the two is the chosen value of 10 K. The HTC refrigerant temperature in the HTC condenser can be calculated as:

$$T_{cond,HTC} = T_{sink} + 10 \tag{2}$$

where $T_{cond,HTC}$ is the HTC refrigerant temperature in the HTC condenser [K], T_{sink} is the temperature of the heat sink [K] and T_{diff} and the difference between the two is the same as that between $T_{evap,LTC}$ and T_{source}.

Assuming the cascade heat exchanger has a temperature equal to the mean of the heat source and sink temperatures, and that the difference between refrigerants in the heat exchanger is 10 K, the temperature of the LTC refrigerant should be half of that value over the mean temperature of the heat source and sink, while the HTC refrigerant should be half of that value below the same mean temperature. This is expressed as:

$$T_{cond,LTC} = \frac{T_{source} + T_{sink}}{2} + \frac{10}{2} \tag{3}$$

where $T_{cond,LTC}$ is the LTC refrigerant temperature in the LTC condenser [K]; and:

$$T_{evap,HTC} = \frac{T_{source} + T_{sink}}{2} - \frac{10}{2} \tag{4}$$

where $T_{evap,HTC}$ is the HTC refrigerant temperature in the LTC evaporator [K].

Calculating the process COP requires finding the enthalpies at all 4 state points. To calculate the fluid properties at various state points, CoolProp was used in the Matlab script, as it is a reliable extensive and open-source fluid property library trusted by many in the scientific and engineering community (Bell, Wronski, et al., 2014). The known values, unknowns obtained through direct calculation and unknowns obtained with CoolProp for each state point are presented in Table 2.

Table 2: Main state point calculations.

State point	Known	Unknown, solved through calculations	Unknown, solved with CoolProp
cond	T_{cond} $Q_{cond} = 0$		p_{cond}
evap	T_{evap} $Q_{evap} = 1$		p_{evap}
1	$p_1 = p_{evap}$ $Q_1 = 1$	$T_1 = T_{evap} + 5$	$h_1,\ s_1$
2s	$p_{2s} = p_{cond}$ $s_{2s} = s_1$		h_{2s}
2	$p_2 = p_{cond}$	$h_2 = h_1 + (h_{2s} - h_1) \div 0.75$	$T_2,\ s_2$
3	$p_3 = p_{cond}$ $Q_3 = 0$	$T_3 = T_{cond} - 5$	$h_3,\ s_3$
4	$p_4 = p_{evap}$ $T_4 = T_{evap}$ $h_4 = h_3$		s_4

where h is specific enthalpy of the refrigerant [J/kg], s is the specific entropy of the refrigerant [J/kgK], p is pressure of the refrigerant [Pa] and Q is the steam quality, measured from 0 to 1, where 0 is saturated liquid and 1 is saturated vapor. The calculations for the LTC and HTC are identical, the only difference being that HTC values are used instead of LTC ones. The model needs to calculate the enthalpy of the fluid in each cycle at state point 2s, which is at the outlet of an ideal isentropic compressor, only as an intermediary value that is used to solve the enthalpy increase after the real compression, this being state point 2, using the isentropic efficiency of the compressors. The pressures of the fluids in the evaporator and condenser are also only calculated as intermediary values needed for later calculations.

Knowing the enthalpies of the refrigerants at all state points and the heat rate entering the evaporator, the mass flow rate of the LTC fluid is calculated as:

$$q_{m,\ LTC} = \frac{q_{evap,\ LTC}}{h_{1,\ LTC} - h_{4,\ LTC}} \tag{5}$$

where $q_{m,\ LTC}$ is the mass flow rate of the LTC refrigerant [kg/s] and $q_{evap,\ LTC}$ is the heat rate entering the LTC evaporator [W]. The mass flow rate of the HTC refrigerant can also be found:

$$q_{m,\ HTC} = q_{m,\ LTC} \frac{h_{2,\ LTC} - h_{3,\ LTC}}{h_{1,\ HTC} - h_{4,\ HTC}} \tag{6}$$

The overall work used by the process is the sum of the work exerted upon the fluid by each compressor:

$$W = q_{m,\ LTC}(h_{2,\ LTC} - h_{1,\ LTC}) + q_{m,\ HTC}(h_{2,\ HTC} - h_{1,\ HTC}) \tag{7}$$

where W is the total work into the system [J].

The useful energy transfer of the system is dictated by the heat added to the heat sink, found as:

$$Q_{cond,\ HTC} = q_{m,\ HTC}(h_{2,\ HTC} - h_{3,\ HTC}) \tag{8}$$

where $Q_{cond,\ HTC}$ is the heat leaving the HTC condenser.

COP can be calculated from the useful heat and overall work:

$$COP = \frac{Q_{cond,\ HTC}}{W} \tag{9}$$

Although the main purpose of the model is to find the COP, the state point parameters of the fluids can also be used to create diagrams of the process, such as that previously shown in Figure 2. Drawing this diagram involves the calculation of the fluid parameters at 3 additional, intermediary state points in each cycle: state point i1 is in the evaporator, before superheating; state point i2 is in the condenser, after cooling; state point i3 is in the condenser, before subcooling. The calculation procedure for these is similar to that of the main state point parameters, except there is no need for further calculations, as shown in Table 3.

Table 3: Intermediary state point calculations.

State point	Known	Unknown, solved with CoolProp
i1	$T_{i1} = T_4$ $p_{i1} = p_4$ $Q_{i1} = 1$	h_{i1}, s_{i1}
i2	$T_{i2} = T_{cond}$ $p_{i2} = p_2$ $Q_{i2} = 1$	h_{i2}, s_{i2}
i3	$T_{i3} = T_{cond}$ $p_{i3} = p_2$ $Q_{i3} = 0$	h_{i3}, s_{i3}

Just like the main state point properties, the intermediary state point property calculations are identical for the LTC and HTC. With the main and intermediary state point properties all known, the process is easily plotted in Matlab using the 'plot' function and some tweaks to various settings for visual clarity. The Matlab script implementation of the heat pump model, which calculates the COP and plots the process for a given LTC refrigerant, HTC refrigerant, heat source temperature and heat sink temperature, is presented in full in Appendix 1.

3.2 Iterative optimization algorithm

To find the best natural refrigerants in a heat pump upgrading waste heat for district heating, a broader approach was needed to analyze all of the refrigerant candidates for each of the heat source and sink temperatures in the possible range. This approach was based on iterative, brute-force search loops using the previously presented heat pump model to find the COP for each configuration.

First, the script defines the input parameters. It sets up the ranges for the heat source and sink temperatures and establishes key process parameters such as temperature differences, superheat, subcooling, compressor efficiency, and the heat rate entering the evaporator. It also specifies the lists of candidate refrigerants, separating natural refrigerants from synthetic ones.

Next, the script performs a usability check for each refrigerant. Using the maximum condensing temperatures calculated from the temperature ranges, the critical temperature of each refrigerant is retrieved via CoolProp. Refrigerants with critical temperatures higher than the maximum condensing requirements are deemed usable in either the low-temperature cycle (LTC) or the high-temperature cycle (HTC).

Then, the script implements four nested 'for' loops to iterate over every valid combination of refrigerants and temperature pairs. For each combination, the integrated heat pump model calculates the COP. The script only considers combinations where the heat source and sink temperatures are less than 5 K apart, as most heat pump cycles operate with temperature lifts of around 20 K, according to Ommen, Markussen & Elmegaard (2014).

Finally, the results are processed and displayed. The script records the COP values and refrigerant selections in a result array during each iteration, converts the array to a table that is exported to an Excel file, and displays the results through heat maps which visually represent the COP performance across the temperature ranges, and the best refrigerant choices for both the overall and natural refrigerant cases.

4 Results

The brute-force search algorithm outputs a total of 2 Excel files and 6 heat maps.

The output of the usability check of the refrigerant selection is an Excel file with the contents presented in Table 4. Here, usability refers to whether or not the refrigerant would be heated to temperatures above its critical point during the cycle. The refrigerants which are not usable are not taken into account for later calculations. Thus, R410a, R744, R1270 and R170 are eliminated entirely from further calculations, while R290 and R134a are only considered for the LTC. Only 5 refrigerant candidates are considered for both cycles, 4 of which are natural.

Table 4: Refrigerant critical temperatures and usability.

Refrigerant	Critical Temperature [C]	Usable in LTC	Usable in HTC
R170	32.172	No	No
R290	96.74	Yes	No
R600	151.975	Yes	Yes
R600a	134.667	Yes	Yes
R1270	91.061	No	No
R717	132.41	Yes	Yes
R744	30.9782	No	No
R718	373.946	Yes	Yes
R134a	101.06	Yes	No
R410a	71.344	No	No
R245fa	153.86	Yes	Yes

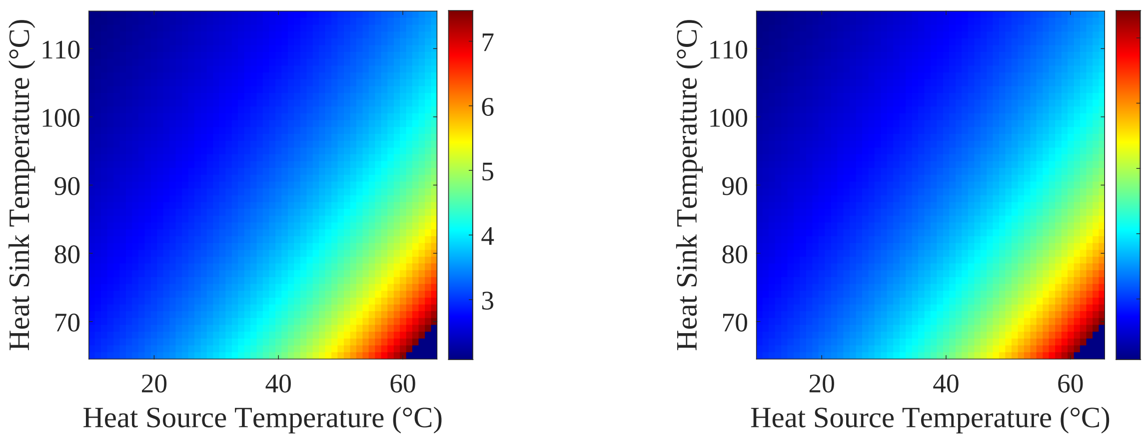
The results of the COP calculations are recorded in the second Excel file. This file contains the heat source and sink temperature, LTC and HTC refrigerants used to get the highest COP for this temperature combination as well as the COP value, and the LTC and HTC refrigerants and highest COP for the case where only natural refrigerants are considered. From this data, the number of times each refrigerant is used to achieve the highest COP for a temperature combination can be counted, presented in Table 5.

Table 5: Number of temperature combinations where refrigerants yield highest COP.

Refrigerant	LTC	HTC	LTC, natural only	HTC, natural only
R134a	464	unusable		
R245fa	1980	364		
R290	0	unusable	0	unusable
R600	0	0	1676	140
R600a	0	0	0	0
R717	421	0	559	0
R718	440	2477	606	2701

Processing the result data further, it can be noted that, for the first case, which considers synthetic and natural refrigerants, the maximum COP was 7.473, obtained at a heat source temperature of 65°C, a heat sink temperature of 70°C, and R245fa in both the LTC and HTC. When only natural refrigerants were considered, the maximum COP was 7.417, achieved at the same temperatures as the first case, with R600 in the LTC and R718 in the HTC. The average COP for the first case was 3.470 and 3.462 for the second case.

Figure 3 shows the two plots of the highest COP calculated for all heat source and heat sink temperature combinations. The heat source temperature spans the range of typical temperatures of industrial waste heat in Finland. The heat sink temperature spans the range of typical temperatures of the district heating supply line in Finland. Figure 3a represents the calculations for the first case, which tested both synthetic and natural refrigerants in the calculations. Figure 3b represents the calculations for the second case, which tested natural refrigerants only.

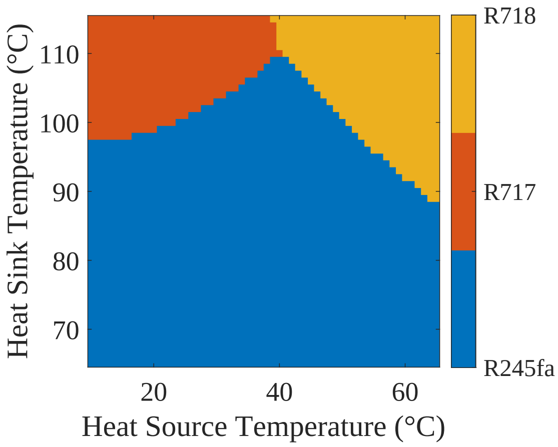


(a) Synthetic and natural refrigerants.

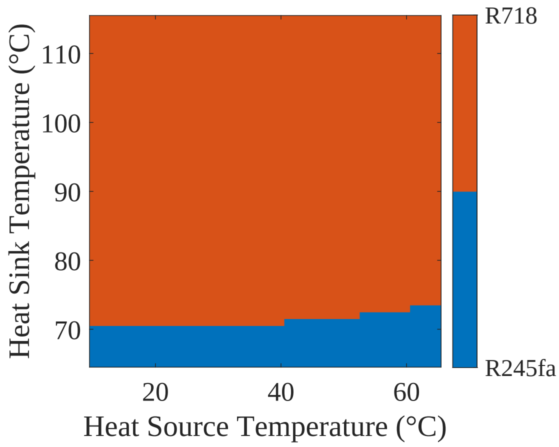
(b) Natural refrigerants only.

Figure 3: Highest COP for heat source and sink temperature combinations.

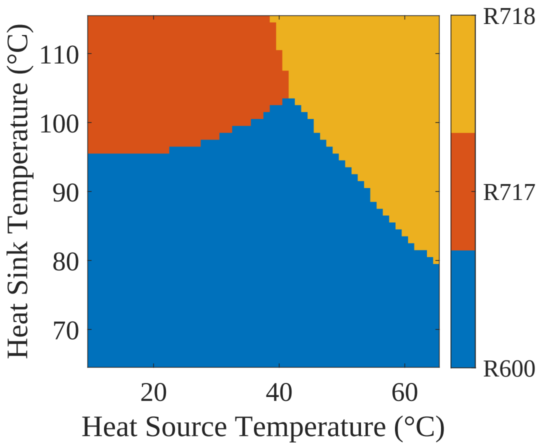
Figure 4 shows the four plots of the refrigerants used in the LTC and HTC of the heat pump cycle to obtain the highest COP for all heat source and heat sink temperature combinations. The heat source and sink temperatures cover the same range as in Figure 3. Figures 4a and 4c show the refrigerant selection for the highest possible COP for every given heat source and heat sink temperature combinations



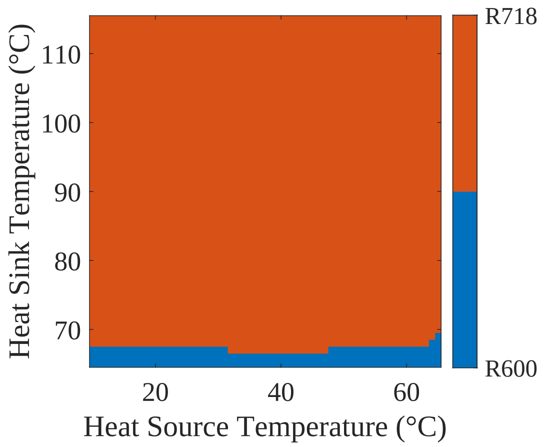
(a) Distribution of LTC refrigerants used to achieve highest COP.



(b) Distribution of HTC refrigerants used to achieve highest COP.



(c) Distribution of LTC natural refrigerants used to achieve highest COP.



(d) Distribution of HTC natural refrigerants used to achieve highest COP.

Figure 4: Refrigerant selection distribution yielding highest COP.

As evidenced by Figure 4, R245fa is the only synthetic refrigerant which is used to obtain the highest COP for any temperature combination in Figures 4a and 4b. Furthermore, the temperature combinations where the HTC refrigerant is R245fa also use R245fa in the LTC in Figures 4a and 4b. When considering only natural refrigerants, most temperature combinations that used R245fa in the first case used R600 for its similar performance. Thus, it becomes important to look at the difference in performance when switching from R245fa to R600 in the LTC only. From the results recorded in the second Excel file generated by the iterative model algorithm, it can be calculated that the average COP for all temperature combinations which use R245fa is 3.689, and 3.678 for the same temperature combinations when using R600 instead.

5 Discussion

As was expected from the thermodynamic analysis of any heat pump cycle, Figure 3 shows that higher COP values are obtained at higher temperature levels with lower temperature lifts. The highest value for COP in both cases considered was obtained at the lowest allowed temperature lift, with a value of 5°C, and highest heat source temperature, with a value of 65°C.

Figure 4 shows that water, or R718, is the best choice for nearly all temperature combinations in the HTC, but also the winner in the LTC in cases where the both heat sink and heat source are towards the high end of the temperature range. Ammonia, or R717, is the best choice for the LTC if the heat sink is at higher temperatures and the heat source is at lower temperatures. R245fa is the the best choice for most temperature combinations in the LTC, as well as for low heat sink temperatures in the HTC. When considering only natural refrigerants, the temperature combinations where R245fa was used in the first case use n-butane, or R600, instead.

When considering the switch from synthetic to natural refrigerants, at least in the case of district heating networks upgrading waste heat with heat pumps in Finland, the change in performance is practically negligible. The average decrease in COP when changing R245fa with R600 is only about 0.002%, and even less if the heat pump was using R134a before the switch, which is far more likely to be the case given R134a's popularity. At the same time, for the switch from R245fa to R600, the decrease in GWP over a 100 year time horizon is from 858 to 20, or 97%.

Although R600 offers almost identical performance compared to its synthetic competitors, it is important to note that it comes with the risk of high flammability. This is an issue that can be circumvented with careful engineering design, but most consumers are unlikely to use it. R717 has similar issues, though with increased toxicity as well. Water, or R718, is the most promising alternative for all synthetic refrigerants, having good performance, and in many cases even the best, while also having the lowest toxicity, GWP and being the most abundant of all refrigerant options.

6 Conclusions

This thesis presents the analysis and selection process of refrigerants used by a cascade heat pump upgrading industrial waste heat for a district heating in Finland. The research aimed to find the difference in performance when changing the working fluid of a heat pump from a synthetic refrigerant to a natural refrigerant and what should be considered when choosing a natural refrigerant. The results showed that the decrease in performance resulting from the switch from synthetic to natural refrigerants is entirely negligible, while the decrease in environmental impact is significant. Water, ammonia and n-butane were the best performing natural refrigerants, with water and ammonia being the better option for higher temperatures and larger temperature lifts, and n-butane being better suited for lower temperatures and lower temperature lifts. Of these, water was noted to be the most promising due to its zero environmental impact, zero toxicity, zero flammability, and high abundance.

Although this thesis succeeded in finding a number of natural refrigerants as good candidates for replacing synthetic refrigerants in the described system from a thermodynamic point of view, there are a number of factors that play into the choice of refrigerant which were not considered in this study. One of the most important is the economic viability, since the price of refrigerants can vary greatly, depending on a number of reasons, such as availability and local regulations. Another major factor are the various physical heat pump design constraints, for example, oil miscibility can cause issues with lubricants, while certain refrigerants can corrode some materials heat pumps are made of. The choice of refrigerant will be inevitably influenced by such other factors, but energy efficiency and environmental effects remain the most crucial.

As climate change worsens, so does the need for lower climate impact refrigerants increase. The fourth generation of refrigerants is being studied and developed for this purpose, and natural refrigerants are a very promising solution, given their good performance and natural abundance. As the energy efficiency associated with switching to a natural refrigerant is not a problem, future research should focus on other potential issues of some natural refrigerants, such as addressing flammability and toxicity by designing heat pumps which can mitigate such risks. Designing systems for a specific refrigerant could further lower the risk of problems arising from other factors, such as unwanted material interactions. Furthermore, it would be helpful to investigate the behavior of an integrated heat pump district heating system, which accounts for the interaction between heat source, heat sink, renewable energy sources providing electricity and the heat pump itself.

References

Abas, N., Kalair, A. R., Khan, N., Haider, A., Saleem, Z. & Saleem, M. S. (2018). Natural and synthetic refrigerants, global warming: A review. eng. *Renewable sustainable energy reviews* 90, pp. 557–569. ISSN: 1364-0321.

Bach, B., Werling, J., Ommen, T., Münster, M., Morales, J. M. & Elmegaard, B. (2016). Integration of large-scale heat pumps in the district heating systems of Greater Copenhagen. eng. *Energy (Oxford)* 107, pp. 321–334. ISSN: 0360-5442.

Bell, I. H., Domanski, P. A., McLinden, M. O. & Linteris, G. T. (2019). The hunt for non-flammable refrigerant blends to replace R-134a. eng. *International journal of refrigeration* 104, pp. 484–495. ISSN: 0140-7007.

Bell, I. H., Wronski, J., Quoilin, S. & Lemort, V. (2014). Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp. *Industrial & Engineering Chemistry Research* 53(6), pp. 2498–2508. DOI: 10.1021/ie4033999. eprint: http://pubs.acs.org/doi/pdf/10.1021/ie4033999. URL: http://pubs.acs.org/doi/abs/10.1021/ie4033999.

Bolaji, B. & Huan, Z. (2013). Ozone depletion and global warming: Case for the use of natural refrigerant - a review. eng. *Renewable sustainable energy reviews* 18, pp. 49–54. ISSN: 1364-0321.

Calm, J. M. & Hourahan, G. C. (2011). Physical, safety, and environmental data for current and alternative refrigerants. In: *Proceedings of 23rd international congress of refrigeration (ICR2011), Prague, Czech Republic*, pp. 21–26.

Calm, J. M. (2008). The next generation of refrigerants - historical review, considerations, and outlook. eng. *International journal of refrigeration* 31(7), pp. 1123–1133. ISSN: 0140-7007.

David, A., Mathiesen, B., Averfalk, H., Werner, S. & Lund, H. (Apr. 2017). Heat Roadmap Europe: Large-Scale Electric Heat Pumps in District Heating Systems. *Energies* 10, p. 578. DOI: 10.3390/en10040578.

Finnish Energy (2025a). *District heating and cooling*. [Website]. [Cited: 2025-03-05]. URL: https://energia.fi/en/energy-sector-in-finland/energy-production/district-heating-and-cooling/.

Finnish Energy (2025b). *District heating networks*. [Website]. [Cited: 2025-01-29]. URL: https://energia.fi/en/energy-sector-in-finland/energy-networks/district-heating-networks/.

Gaur, A. S., Fitiwi, D. Z. & Curtis, J. (2021). Heat pumps and our low-carbon future: A comprehensive review. eng. *Energy research social science* 71, pp. 101764–. ISSN: 2214-6296.

Heredia-Aricapa, Y., Belman-Flores, J., Mota-Babiloni, A., Serrano-Arellano, J. & García-Pabón, J. J. (2020). Overview of low GWP mixtures for the replacement of HFC refrigerants: R134a, R404A and R410A. eng. *International journal of refrigeration* 111, pp. 113–123. ISSN: 0140-7007.

Jiang, J., Hu, B., Wang, R., Deng, N., Cao, F. & Wang, C.-C. (2022). A review and perspective on industry high-temperature heat pumps. eng. *Renewable sustainable energy reviews* 161, pp. 112106–. ISSN: 1364-0321.

Official Statistics of Finland OSF (2025). *Greenhouse gases*. [Website]. [Cited: 2025-01-15]. URL: https://stat.fi/en/publication/clmpwhj34iw9n0cuntl6ngb08.

Ommen, T., Markussen, W. B. & Elmegaard, B. (2014). Heat pumps in combined heat and power systems. eng. *Energy (Oxford)* 76, pp. 989–1000. ISSN: 0360-5442.

Østergaard, P. A. & Andersen, A. N. (2016). Booster heat pumps and central heat pumps in district heating. eng. *Applied energy* 184, pp. 1374–1388. ISSN: 0306-2619.

Rangnekar, S. (2023). *Estimation of waste heat potential for application in high temperature heat pumps in Finland*. eng.

UNEP, U. et al. (2006). Handbook for the Montreal protocol on substances that deplete the ozone layer. *UNEP/Earthprint*.

Uusitalo, A., Turunen-Saaresti, T., Honkatukia, J., Tiainen, J. & Jaatinen-Värri, A. (2020-05-14). Numerical analysis of working fluids for large scale centrifugal compressor driven cascade heat pumps upgrading waste heat. eng. *Applied Energy*.

Appendix 1 Matlab script for the heat pump model

```
% Heat source temperature [K]
T_source = 40 + 273.15;
% Heat sink temperature [K]
T_sink = 100 + 273.15;
% LTC refrigerant
refrigerant_LTC = 'R718';
% HTC refrigerant
refrigerant_HTC = 'R718';
% Temperature difference between: heat source and LTC refrigerant in the
    LTC evaporator;
% the LTC and HTC refrigerants in the cascade heat exchanger;
% heat sink and HTC refrigerant in the HTC condenser [K]
T_diff = 10;
% Degree of superheating before compressors [K]
T_superheat = 5;
% Degree of subcooling before condensers [K]
T_subcool = 5;
% Isentropic efficiency of compressors
eta_isentropic = 0.75;
% Heat rate to LTC evaporator [W]
q_evaporator_LTC = 10^6;


% LTC evaporator and HTC condenser temperatures can be calculated using
    the temperature difference
T_evaporator_LTC = T_source - T_diff;
T_condenser_HTC = T_sink + T_diff;

% The heat pump is made up of 2 standard heat pump cycles connected by a
% cascade heat exchanger. The temperatures of the LTC and HTC
    refrigerants are
% calculated around the mean of the heat sink and heat source
    temperatures.
T_condenser_LTC = (T_evaporator_LTC + T_condenser_HTC) / 2 + T_diff / 2;
T_evaporator_HTC = (T_evaporator_LTC + T_condenser_HTC) / 2 - T_diff / 2;

% 1-2 is a real compression
% 2-3 is a condensation process preceded by cooling and followed by
    subcooling
% 3-4 is an isenthalpic expansion
% 4-1 is an evaporation process followed by superheating

% State point 1 is after the evaporator and before the compressor.
% State point 2 is after the compressor and before the condenser.
```

```
% State point 3 is after the condenser and before the expansion valve.
% State point 4 is after the expansion valve and before the evaporator.

% State point i1 is in the evaporator, before superheating
% State point i2 is in the condenser, after cooling
% State point i3 is in the condenser, before subcooling

% Defining pressures in the evaporator and condenser of each cycle
p_evaporator_LTC = py.CoolProp.CoolProp.PropsSI('P', 'T', T_evaporator_
    LTC, 'Q', 1, refrigerant_LTC);
p_condenser_LTC = py.CoolProp.CoolProp.PropsSI('P', 'T', T_condenser_LTC,
     'Q', 0, refrigerant_LTC);
p_evaporator_HTC = py.CoolProp.CoolProp.PropsSI('P', 'T', T_evaporator_
    HTC, 'Q', 1, refrigerant_HTC);
p_condenser_HTC = py.CoolProp.CoolProp.PropsSI('P', 'T', T_condenser_HTC,
     'Q', 0, refrigerant_HTC);

% State point properties for the Low Temperature Cycle (LTC)
T1_LTC = T_evaporator_LTC + T_superheat;
p1_LTC = p_evaporator_LTC;
h1_LTC = py.CoolProp.CoolProp.PropsSI('H', 'T', T1_LTC, 'P', p1_LTC,
    refrigerant_LTC);
s1_LTC = py.CoolProp.CoolProp.PropsSI('S', 'T', T1_LTC, 'P', p1_LTC,
    refrigerant_LTC);

h2s_LTC = py.CoolProp.CoolProp.PropsSI('H', 'P', p_condenser_LTC, 'S', s
    1_LTC, refrigerant_LTC);
h2_LTC = h1_LTC + (h2s_LTC - h1_LTC) / eta_isentropic;
p2_LTC = p_condenser_LTC;
T2_LTC = py.CoolProp.CoolProp.PropsSI('T', 'H', h2_LTC, 'P', p_condenser_
    LTC, refrigerant_LTC);
s2_LTC = py.CoolProp.CoolProp.PropsSI('S', 'H', h2_LTC, 'P', p_condenser_
    LTC, refrigerant_LTC);

T3_LTC = T_condenser_LTC - T_subcool;
p3_LTC = p2_LTC;
h3_LTC = py.CoolProp.CoolProp.PropsSI('H', 'Q', 0, 'T', T3_LTC,
    refrigerant_LTC);
s3_LTC = py.CoolProp.CoolProp.PropsSI('S', 'Q', 0, 'T', T3_LTC,
    refrigerant_LTC);

T4_LTC = T_evaporator_LTC;
p4_LTC = p1_LTC;
h4_LTC = h3_LTC;
s4_LTC = py.CoolProp.CoolProp.PropsSI('S', 'P', p4_LTC, 'H', h4_LTC,
    refrigerant_LTC);
```

```
% State point properties for the High Temperature Cycle (HTC)
T1_HTC = T_evaporator_HTC + T_superheat;
p1_HTC = p_evaporator_HTC;
h1_HTC = py.CoolProp.CoolProp.PropsSI('H', 'T', T1_HTC, 'P', p1_HTC,
    refrigerant_HTC);
s1_HTC = py.CoolProp.CoolProp.PropsSI('S', 'T', T1_HTC, 'P', p1_HTC,
    refrigerant_HTC);

h2s_HTC = py.CoolProp.CoolProp.PropsSI('H', 'P', p_condenser_HTC, 'S', s
    1_HTC, refrigerant_HTC);
h2_HTC = h1_HTC + (h2s_HTC - h1_HTC) / eta_isentropic;
p2_HTC = p_condenser_HTC;
T2_HTC = py.CoolProp.CoolProp.PropsSI('T', 'H', h2_HTC, 'P', p2_HTC,
    refrigerant_HTC);
s2_HTC = py.CoolProp.CoolProp.PropsSI('S', 'H', h2_HTC, 'P', p2_HTC,
    refrigerant_HTC);

T3_HTC = T_condenser_HTC - T_subcool;
p3_HTC = p2_HTC;
h3_HTC = py.CoolProp.CoolProp.PropsSI('H', 'Q', 0, 'T', T3_HTC,
    refrigerant_HTC);
s3_HTC = py.CoolProp.CoolProp.PropsSI('S', 'Q', 0, 'T', T3_HTC,
    refrigerant_HTC);

T4_LTC = T_evaporator_HTC;
p4_HTC = p1_HTC;
h4_HTC = h3_HTC;
s4_HTC = py.CoolProp.CoolProp.PropsSI('S', 'P', p4_HTC, 'H', h4_HTC,
    refrigerant_HTC);

% Find refrigerant mass flow rates from evaporator heat rate
qm_LTC = q_evaporator_LTC / (h1_LTC - h4_LTC);
qm_HTC = qm_LTC * (h2_LTC - h3_LTC) / (h1_HTC - h4_HTC);

% Calculate COP
W = qm_LTC * (h2_LTC - h1_LTC ) + qm_HTC * (h2_HTC - h1_HTC);
Q_condenser_HTC = qm_HTC * (h2_HTC - h3_HTC);
COP = Q_condenser_HTC / W

%Intermediate state points, for plotting
Ti1_LTC = T4_LTC;
pi1_LTC = p4_LTC;
hi1_LTC = py.CoolProp.CoolProp.PropsSI('H', 'T', Ti1_LTC, 'Q', 1,
    refrigerant_LTC);
si1_LTC = py.CoolProp.CoolProp.PropsSI('S', 'T', Ti1_LTC, 'Q', 1,
```

```
    refrigerant_LTC);

Ti2_LTC = T_condenser_LTC;
pi2_LTC = p2_LTC;
hi2_LTC = py.CoolProp.CoolProp.PropsSI('H', 'T', Ti2_LTC, 'Q', 1,
    refrigerant_LTC);
si2_LTC = py.CoolProp.CoolProp.PropsSI('S', 'T', Ti2_LTC, 'Q', 1,
    refrigerant_LTC);

Ti3_LTC = T_condenser_LTC;
pi3_LTC = p2_LTC;
hi3_LTC = py.CoolProp.CoolProp.PropsSI('H', 'T', Ti3_LTC, 'Q', 0,
    refrigerant_LTC);
si3_LTC = py.CoolProp.CoolProp.PropsSI('S', 'T', Ti3_LTC, 'Q', 0,
    refrigerant_LTC);

Ti1_HTC = T4_HTC;
pi1_HTC = p4_HTC;
hi1_HTC = py.CoolProp.CoolProp.PropsSI('H', 'T', Ti1_HTC, 'Q', 1,
    refrigerant_HTC);
si1_HTC = py.CoolProp.CoolProp.PropsSI('S', 'T', Ti1_HTC, 'Q', 1,
    refrigerant_HTC);

Ti2_HTC = T_condenser_HTC;
pi2_HTC = p2_HTC;
hi2_HTC = py.CoolProp.CoolProp.PropsSI('H', 'T', Ti2_HTC, 'Q', 1,
    refrigerant_HTC);
si2_HTC = py.CoolProp.CoolProp.PropsSI('S', 'T', Ti2_HTC, 'Q', 1,
    refrigerant_HTC);

Ti3_HTC = T_condenser_HTC;
pi3_HTC = p2_HTC;
hi3_HTC = py.CoolProp.CoolProp.PropsSI('H', 'T', Ti3_HTC, 'Q', 0,
    refrigerant_HTC);
si3_HTC = py.CoolProp.CoolProp.PropsSI('S', 'T', Ti3_HTC, 'Q', 0,
    refrigerant_HTC);


% Define saturation curves for LTC
T_crit_LTC = py.CoolProp.CoolProp.PropsSI('Tcrit', refrigerant_LTC);
T_sat_LTC = linspace(T_source, T_crit_LTC, 100);
h_sat_LTC_liq = zeros(size(T_sat_LTC)); % Liquid phase enthalpy
h_sat_LTC_vap = zeros(size(T_sat_LTC)); % Vapor phase enthalpy
p_sat_LTC_liq = zeros(size(T_sat_LTC)); % Liquid phase pressure
p_sat_LTC_vap = zeros(size(T_sat_LTC)); % Vapor phase pressure
```

```
for i = 1:length(T_sat_LTC)
    p_sat_LTC_liq(i) = py.CoolProp.CoolProp.PropsSI('P', 'T', T_sat_LTC(i
        ), 'Q', 0, refrigerant_LTC);
    p_sat_LTC_vap(i) = py.CoolProp.CoolProp.PropsSI('P', 'T', T_sat_LTC(i
        ), 'Q', 1, refrigerant_LTC);
    h_sat_LTC_liq(i) = py.CoolProp.CoolProp.PropsSI('H', 'T', T_sat_LTC(i
        ), 'Q', 0, refrigerant_LTC);
    h_sat_LTC_vap(i) = py.CoolProp.CoolProp.PropsSI('H', 'T', T_sat_LTC(i
        ), 'Q', 1, refrigerant_LTC);
end

% Define saturation curves for HTC
T_crit_HTC = py.CoolProp.CoolProp.PropsSI('Tcrit', refrigerant_HTC);
T_sat_HTC = linspace(T_evaporator_HTC, T_crit_HTC, 100);
h_sat_HTC_liq = zeros(size(T_sat_HTC));
h_sat_HTC_vap = zeros(size(T_sat_HTC));
p_sat_HTC_liq = zeros(size(T_sat_HTC));
p_sat_HTC_vap = zeros(size(T_sat_HTC));

for i = 1:length(T_sat_HTC)
    p_sat_HTC_liq(i) = py.CoolProp.CoolProp.PropsSI('P', 'T', T_sat_HTC(i
        ), 'Q', 0, refrigerant_HTC);
    p_sat_HTC_vap(i) = py.CoolProp.CoolProp.PropsSI('P', 'T', T_sat_HTC(i
        ), 'Q', 1, refrigerant_HTC);
    h_sat_HTC_liq(i) = py.CoolProp.CoolProp.PropsSI('H', 'T', T_sat_HTC(i
        ), 'Q', 0, refrigerant_HTC);
    h_sat_HTC_vap(i) = py.CoolProp.CoolProp.PropsSI('H', 'T', T_sat_HTC(i
        ), 'Q', 1, refrigerant_HTC);
end

% Log p-h diagram
figure;
hold on;
set(gca, 'YScale', 'log');
set(gca, 'FontSize', 12);

% Plot LTC
plot([h1_LTC, h2_LTC], [p1_LTC, p2_LTC], '-o', 'Color', 'b', '
    HandleVisibility', 'off');
plot([h2_LTC, hi2_LTC], [p2_LTC, pi2_LTC], '-o', 'Color', 'b', '
    HandleVisibility', 'off');
plot([hi2_LTC, hi3_LTC], [pi2_LTC, pi3_LTC], '-o', 'Color', 'b', '
    HandleVisibility', 'off');
plot([hi3_LTC, h3_LTC], [pi3_LTC, p3_LTC], '-o', 'Color', 'b', '
    HandleVisibility', 'off');
plot([h3_LTC, h4_LTC], [p3_LTC, p4_LTC], '-o', 'Color', 'b', '
```

```
    HandleVisibility', 'off');
plot([h4_LTC, hi1_LTC], [p4_LTC, pi1_LTC], '-o', 'Color', 'b', '
    HandleVisibility', 'off');
plot([hi1_LTC, h1_LTC], [pi1_LTC, p1_LTC], '-o', 'Color', 'b', '
    HandleVisibility', 'off');
text(h1_LTC, p1_LTC * 0.7, ' 1', 'FontSize', 12, 'Color', 'b');
text(h2_LTC, p2_LTC * 1.4, ' 2', 'FontSize', 12, 'Color', 'b');
text(h3_LTC * 0.6, p3_LTC * 1.4, ' 3', 'FontSize', 12, 'Color', 'b');
text(h4_LTC, p4_LTC * 0.7, ' 4', 'FontSize', 12, 'Color', 'b');
plot(NaN, NaN, '-ob', 'DisplayName', 'LTC Cycle');

% Plot HTC
plot([h1_HTC, h2_HTC], [p1_HTC, p2_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
plot([h2_HTC, hi2_HTC], [p2_HTC, pi2_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
plot([hi2_HTC, hi3_HTC], [pi2_HTC, pi3_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
plot([hi3_HTC, h3_HTC], [pi3_HTC, p3_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
plot([h3_HTC, h4_HTC], [p3_HTC, p4_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
plot([h4_HTC, hi1_HTC], [p4_HTC, pi1_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
plot([hi1_HTC, h1_HTC], [pi1_HTC, p1_HTC], '-o', 'Color', 'r', '
    HandleVisibility', 'off');
text(h1_HTC, p1_HTC * 0.7, ' 1', 'FontSize', 12, 'Color', 'r');
text(h2_HTC, p2_HTC * 1.4, ' 2', 'FontSize', 12, 'Color', 'r');
text(h3_HTC * 0.65, p3_HTC * 1.3, ' 3', 'FontSize', 12, 'Color', 'r');
text(h4_HTC, p4_HTC * 0.7 , ' 4', 'FontSize', 12, 'Color', 'r');
plot(NaN, NaN, '-or', 'DisplayName', 'HTC Cycle');

if strcmp(refrigerant_LTC, refrigerant_HTC)
    % Plot saturation curve if the LTC refrigerant is the same as the HTC
        refrigerant
     plot(h_sat_LTC_liq, p_sat_LTC_liq, '--k', 'HandleVisibility', 'off');
     plot(h_sat_LTC_vap, p_sat_LTC_vap, '--k', 'HandleVisibility', 'off');
     plot(NaN, NaN, '--ok', 'DisplayName', 'Saturation Curve');

else
    % Plot saturation curves for LTC
     plot(h_sat_LTC_liq, p_sat_LTC_liq, '--b', 'HandleVisibility', 'off');
     plot(h_sat_LTC_vap, p_sat_LTC_vap, '--b', 'HandleVisibility', 'off');
     plot(NaN, NaN, '--ob', 'DisplayName', 'LTC Saturation Curve');

    % Plot saturation curves for HTC
```

```
    plot(h_sat_HTC_liq, p_sat_HTC_liq, '--r', 'HandleVisibility', 'off');
    plot(h_sat_HTC_vap, p_sat_HTC_vap, '--r', 'HandleVisibility', 'off');
    plot(NaN, NaN, '--or', 'DisplayName', 'HTC Saturation Curve');
end

title('Log p-h diagram of cascade heat pump', 'FontSize', 12);
xlabel('Enthalpy [J/kg]', 'FontSize', 12);
ylabel('Pressure [Pa], log scale', 'FontSize', 12);
legend('FontSize', 12);
grid on;
hold off;

% Export figure as PDF
exportgraphics(gcf, 'log_p_h_diagram.pdf', 'ContentType', 'vector');
```

Appendix 2 Matlab script for the iterative implementation of the model

```
clear;
close all;

% Heat source temperature range [K]
T_source_range = (10:1:65) + 273.15;
% Heat sink temperature range [K]
T_sink_range = (65:1:115) + 273.15;
% Minimum temperature difference between heat source and heat sink [K]
min_sink_source_T_diff = 5;
% Temperature difference between: heat source and LTC refrigerant in the
    LTC evaporator;
% the LTC and HTC refrigerants in the cascade heat exchanger;
% heat sink and HTC refrigerant in the HTC condenser [K]
T_diff = 10;
% Degree of superheating before compressors [K]
T_superheat = 5;
% Degree of subcooling before condensers [K]
T_subcool = 5;
% Isentropic efficiency of compressors
eta_isentropic = 0.75;
% Heat rate to LTC evaporator [W]
q_evaporator_LTC = 10^6;
% Possible refrigerants for either cycle
natural_refrigerants = {'R170', 'R290', 'R600', 'R600a', 'R1270', 'R717',
     'R744', 'R718'};
synthetic_refrigerants = {'R134a', 'R410a', 'R245fa'};

% The highest possible condensing temperature of each cycle is calculated
% for assessing the usability of the refrigerants
T_max_condenser_LTC = (T_source_range(length(T_source_range)) + T_sink_
    range(length(T_sink_range))) / 2 + T_diff / 2;
T_max_condenser_HTC = T_sink_range(length(T_sink_range)) + T_diff;

% Preallocating the arrays containing the usable refrigerants for each
% cycle and the usability table
refrigerants_LTC={};
refrigerants_HTC={};
refrigerants = [natural_refrigerants, synthetic_refrigerants];
usability = cell(length(refrigerants), 3);

for i = 1:length(refrigerants)
    refrigerant = refrigerants{i};
```

```
    usability{i, 1} = refrigerant;
    T_crit = py.CoolProp.CoolProp.PropsSI('Tcrit', refrigerant);
    usability{i, 2} = T_crit - 273.15;

    if T_crit > T_max_condenser_LTC
        usability{i, 3} = 'Yes';
        refrigerants_LTC{length(refrigerants_LTC)+1} = refrigerant;
    else
        usability{i, 3} = 'No';
    end

    if T_crit > T_max_condenser_HTC
        usability{i, 4} = 'Yes';
        refrigerants_HTC{length(refrigerants_HTC)+1} = refrigerant;
    else
        usability{i, 4} = 'No';
    end
end

% Display a table showing which refrigerant is usable for what cycle
usability_table = cell2table(usability, 'VariableNames', {'Refrigerant',
    'Critical Temperature [C]', 'Usable in LTC', 'Usable in HTC'});
usability_table.Refrigerant = string(usability_table.Refrigerant);
disp(usability_table);
writetable(usability_table, 'Usability.xlsx');

% This is implemented this way because the script is designed to be
    usable
% without prior knowledge about the properties of specific refrigerants


% Pre-allocating 'results' cell array by calculating its size (for
    increased performance)
N_rows = numel(T_source_range) * numel(T_sink_range);
N_columns = 8;
% 1 extra row is needed for the column labels on the first line of the
    array
results = cell(N_rows + 1, N_columns);
results(1, :) = {'Heat source temperature', 'Heat sink temperature', 'LTC
     refrigerant for highest COP', 'HTC refrigerant for highest COP', '
    Highest COP', 'LTC natural refrigerant for highest COP', 'HTC natural
    refrigerant for highest COP', 'Highest COP with natural refrigerants
    '};
% Index of the current row where results are to be written
row_i = 2;
```

```
for T_source = T_source_range
    for T_sink = T_sink_range
        % The heat source and heat sink temperatures must be the minimum
            temperature difference apart
        if T_sink - T_source < min_sink_source_T_diff
            continue;
        end

        % LTC evaporator and HTC condenser temperatures can be calculated
             using the temperature difference
        T_evaporator_LTC = T_source - T_diff;
        T_condenser_HTC = T_sink + T_diff;

        % The heat pump is made up of 2 standard heat pump cycles
            connected by a cascade heat exchanger.
        % The temperatures of the LTC and HTC fluids in  the heat
            exchanger are calculated around the mean of T_sink
        % the heat sink and heat source temperatures.
        T_condenser_LTC = (T_evaporator_LTC + T_condenser_HTC) / 2 + T_
            diff / 2;
        T_evaporator_HTC = (T_evaporator_LTC + T_condenser_HTC) / 2 - T_
            diff / 2;

        % Initializing the array containing the refrigerant combination
            with the highest COP for this temperature combination
        best_COP = {'0', '0', 0};
        % Initializing the array containing the natural refrigerant
            combination with the highest COP for this temperature
            combination
        best_nat_COP = {'0', '0', 0};

        % Testing all refrigerant combinations
        for refrigerant_LTC = refrigerants_LTC
            for refrigerant_HTC = refrigerants_HTC
                try
                    % Defining pressures in the evaporator and condenser
                        of each cycle
                    p_evaporator_LTC = py.CoolProp.CoolProp.PropsSI('P',
                        'T', T_evaporator_LTC, 'Q', 1, refrigerant_LTC{1})
                        ;
                    p_condenser_LTC = py.CoolProp.CoolProp.PropsSI('P', '
                        T', T_condenser_LTC, 'Q', 0, refrigerant_LTC{1});
                    p_evaporator_HTC = py.CoolProp.CoolProp.PropsSI('P',
                        'T', T_evaporator_HTC, 'Q', 1, refrigerant_HTC{1})
                        ;
                    p_condenser_HTC = py.CoolProp.CoolProp.PropsSI('P', '
```

```
    T', T_condenser_HTC, 'Q', 0, refrigerant_HTC{1});

% State points properties for the Low Temperature
    Cycle (LTC)
T1_LTC = T_evaporator_LTC + T_superheat;
p1_LTC = p_evaporator_LTC;
h1_LTC = py.CoolProp.CoolProp.PropsSI('H', 'T', T1_
    LTC, 'P', p1_LTC, refrigerant_LTC{1});
s1_LTC = py.CoolProp.CoolProp.PropsSI('S', 'T', T1_
    LTC, 'P', p1_LTC, refrigerant_LTC{1});

h2s_LTC = py.CoolProp.CoolProp.PropsSI('H', 'P', p_
    condenser_LTC, 'S', s1_LTC, refrigerant_LTC{1});
h2_LTC = h1_LTC + (h2s_LTC - h1_LTC) / eta_isentropic
    ;
p2_LTC = p_condenser_LTC;
T2_LTC = py.CoolProp.CoolProp.PropsSI('T', 'H', h2_
    LTC, 'P', p_condenser_LTC, refrigerant_LTC{1});
s2_LTC = py.CoolProp.CoolProp.PropsSI('S', 'H', h2_
    LTC, 'P', p_condenser_LTC, refrigerant_LTC{1});

T3_LTC = T_condenser_LTC - T_subcool;
p3_LTC = p2_LTC;
h3_LTC = py.CoolProp.CoolProp.PropsSI('H', 'Q', 0, 'T
    ', T3_LTC, refrigerant_LTC{1});
s3_LTC = py.CoolProp.CoolProp.PropsSI('S', 'Q', 0, 'T
    ', T3_LTC, refrigerant_LTC{1});

T4_LTC = T_evaporator_LTC;
p4_LTC = p1_LTC;
h4_LTC = h3_LTC;
s4_LTC = py.CoolProp.CoolProp.PropsSI('S', 'P', p4_
    LTC, 'H', h4_LTC, refrigerant_LTC{1});

% State points properties for the High Temperature
    Cycle (HTC)
T1_HTC = T_evaporator_HTC + T_superheat;
p1_HTC = p_evaporator_HTC;
h1_HTC = py.CoolProp.CoolProp.PropsSI('H', 'T', T1_
    HTC, 'P', p1_HTC, refrigerant_HTC{1});
s1_HTC = py.CoolProp.CoolProp.PropsSI('S', 'T', T1_
    HTC, 'P', p1_HTC, refrigerant_HTC{1});

h2s_HTC = py.CoolProp.CoolProp.PropsSI('H', 'P', p_
    condenser_HTC, 'S', s1_HTC, refrigerant_HTC{1});
h2_HTC = h1_HTC + (h2s_HTC - h1_HTC) / eta_isentropic
```

```
    ;
p2_HTC = p_condenser_HTC;
T2_HTC = py.CoolProp.CoolProp.PropsSI('T', 'H', h2_
    HTC, 'P', p2_HTC, refrigerant_HTC{1});
s2_HTC = py.CoolProp.CoolProp.PropsSI('S', 'H', h2_
    HTC, 'P', p2_HTC, refrigerant_HTC{1});

T3_HTC = T_condenser_HTC - T_subcool;
p3_HTC = p2_HTC;
h3_HTC = py.CoolProp.CoolProp.PropsSI('H', 'Q', 0, 'T
    ', T3_HTC, refrigerant_HTC{1});
s3_HTC = py.CoolProp.CoolProp.PropsSI('S', 'Q', 0, 'T
    ', T3_HTC, refrigerant_HTC{1});

T4_HTC = T_evaporator_HTC;
p4_HTC = p1_HTC;
h4_HTC = h3_HTC;
s4_HTC = py.CoolProp.CoolProp.PropsSI('S', 'P', p4_
    HTC, 'H', h4_HTC, refrigerant_HTC{1});

% Finding refrigerant mass flow rates from evaporator
     heat rate
qm_LTC = q_evaporator_LTC / (h1_LTC - h4_LTC);
qm_HTC = qm_LTC * (h2_LTC - h3_LTC) / (h1_HTC - h4_
    HTC);

% Calculating COP
W_comp = qm_LTC * (h2_LTC - h1_LTC ) + qm_HTC * (h2_
    HTC - h1_HTC);
Q_condenser_HTC = qm_HTC * (h2_HTC - h3_HTC);
COP = Q_condenser_HTC / W_comp;

% Overwriting the previous highest COP if the current
     COP is higher
if COP > best_COP{3}
    best_COP = {refrigerant_LTC{1}, refrigerant_HTC
        {1}, COP};
end

% Overwriting the previous highest natural
    refrigerant COP if the current COP is higher
% and neither cycle is using R134a as its refrigerant
if ismember(refrigerant_LTC{1}, natural_refrigerants)
     && ismember(refrigerant_HTC{1}, natural_
    refrigerants) && COP > best_nat_COP{3}
    best_nat_COP = {refrigerant_LTC{1}, refrigerant_
```

```
                            HTC{1}, COP};
                    end

                catch
                    continue;
                end
            end
        end

        % Writing results
        results(row_i, :) = [{T_source-273.15}, {T_sink-273.15}, best_COP
            (:)', best_nat_COP(:)'];
        row_i = row_i + 1;
    end
end

% Converting results to table
results_table = cell2table(results(2:row_i-1, :), 'VariableNames',
    results(1, :));
numeric_columns = {'Heat source temperature', 'Heat sink temperature', '
    Highest COP', 'Highest COP with natural refrigerants'};
results_table = convertvars(results_table, setdiff(results_table.
    Properties.VariableNames, numeric_columns), @(x) string(x));

% Writing results to an Excel file
writetable(results_table, 'Results.xlsx');

% Extracting numeric values from result array for plotting
n_T_source = cell2mat(results(2:end, 1));
n_T_sink = cell2mat(results(2:end, 2));
n_COP = cell2mat(results(2:end, 5));
n_COP_nat = cell2mat(results(2:end, 8));

% Defining a grid for heat map
[X, Y] = meshgrid(unique(n_T_source), unique(n_T_sink));

% Creating empty matrices to store COP values
Z = nan(size(X));        % For highest COP
Z_nat = nan(size(X));    % For highest COP with natural refrigerants

% Defining matrices with heat sink and source temperature combinations
% with their corresponding COP values
for i = 1:length(n_T_source)
    xi = find(X(1,:) == n_T_source(i)); % Find column index
    yi = find(Y(:,1) == n_T_sink(i));   % Find row index
    Z(yi, xi) = n_COP(i);               % Assign COP value
```

```
    Z_nat(yi, xi) = n_COP_nat(i);   % Assign COP (natural refrigerants)
end

% Heat map for highest COP
figure;
imagesc(unique(n_T_source), unique(n_T_sink), Z);
set(gca, 'YDir', 'normal', 'FontName', 'Times New Roman');
xlabel('Heat Source Temperature ( C )', 'FontSize', 12, 'FontName', '
    Times New Roman');
ylabel('Heat Sink Temperature ( C )', 'FontSize', 12, 'FontName', 'Times
    New Roman');
set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');
set(colorbar, 'FontSize', 12, 'FontName', 'Times New Roman');
colormap jet;
print(gcf, 'Heat_Map_Highest_COP.pdf', '-dpdf', '-bestfit');

% Heat map for highest COP using only natural refrigerants
figure;
imagesc(unique(n_T_source), unique(n_T_sink), Z_nat);
set(gca, 'YDir', 'normal', 'FontName', 'Times New Roman');
xlabel('Heat Source Temperature ( C )', 'FontSize', 12, 'FontName', '
    Times New Roman');
ylabel('Heat Sink Temperature ( C )', 'FontSize', 12, 'FontName', 'Times
    New Roman');
set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');
set(colorbar, 'FontSize', 12, 'FontName', 'Times New Roman');
colormap jet;
print(gcf, 'Heat_Map_Highest_COP_Natural.pdf', '-dpdf', '-bestfit');

% Extracting refrigerant values for plotting
ref_LTC = results_table.("LTC refrigerant for highest COP");
ref_HTC = results_table.("HTC refrigerant for highest COP");
ref_LTC_nat = results_table.("LTC natural refrigerant for highest COP");
ref_HTC_nat = results_table.("HTC natural refrigerant for highest COP");

% Remove entries with no COP
valid_idx = results_table.("Highest COP") > 0;
ref_LTC = ref_LTC(valid_idx);
ref_HTC = ref_HTC(valid_idx);
ref_LTC_nat = ref_LTC_nat(valid_idx);
ref_HTC_nat = ref_HTC_nat(valid_idx);
n_T_source = results_table.("Heat source temperature")(valid_idx);
n_T_sink = results_table.("Heat sink temperature")(valid_idx);

% Only consider unique refrigerants
unique_LTC = unique(ref_LTC);
```

```
unique_HTC = unique(ref_HTC);
unique_LTC_nat = unique(ref_LTC_nat);
unique_HTC_nat = unique(ref_HTC_nat);

% Create mapping from refrigerant to numeric value for coloring
ref_map_LTC = containers.Map(cellstr(unique_LTC), 1:length(unique_LTC));
ref_map_HTC = containers.Map(cellstr(unique_HTC), 1:length(unique_HTC));
ref_map_LTC_nat = containers.Map(cellstr(unique_LTC_nat), 1:length(unique
    _LTC_nat));
ref_map_HTC_nat = containers.Map(cellstr(unique_HTC_nat), 1:length(unique
    _HTC_nat));

% Initialize matrices
Z_ref_LTC = nan(size(X));
Z_ref_HTC = nan(size(X));
Z_ref_LTC_nat = nan(size(X));
Z_ref_HTC_nat = nan(size(X));

% Fill the matrices with refrigerant indices
for i = 1:length(n_T_source)
    xi = find(X(1,:) == n_T_source(i));
    yi = find(Y(:,1) == n_T_sink(i));
    Z_ref_LTC(yi, xi) = ref_map_LTC(char(ref_LTC(i)));
    Z_ref_HTC(yi, xi) = ref_map_HTC(char(ref_HTC(i)));
    Z_ref_LTC_nat(yi, xi) = ref_map_LTC_nat(char(ref_LTC_nat(i)));
    Z_ref_HTC_nat(yi, xi) = ref_map_HTC_nat(char(ref_HTC_nat(i)));
end

% Plot LTC refrigerants
figure;
h = imagesc(unique(n_T_source), unique(n_T_sink), Z_ref_LTC);
set(gca, 'YDir', 'normal', 'FontName', 'Times New Roman');
xlabel('Heat Source Temperature ( C )', 'FontSize', 12, 'FontName', '
    Times New Roman');
ylabel('Heat Sink Temperature ( C )', 'FontSize', 12, 'FontName', 'Times
    New Roman');
title('LTC Refrigerant for Highest COP', 'FontSize', 12, 'FontName', '
    Times New Roman');
colormap(lines(length(unique_LTC)));
c = colorbar('Ticks', 1:length(unique_LTC), 'TickLabels', cellstr(unique_
    LTC));
set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');
print(gcf, 'LTC_Refrigerant_Selection.pdf', '-dpdf', '-bestfit');

% Plot HTC refrigerants
figure;
```

```
imagesc(unique(n_T_source), unique(n_T_sink), Z_ref_HTC);
set(gca, 'YDir', 'normal', 'FontName', 'Times New Roman');
xlabel('Heat Source Temperature ( C )', 'FontSize', 12, 'FontName', '
    Times New Roman');
ylabel('Heat Sink Temperature ( C )', 'FontSize', 12, 'FontName', 'Times
    New Roman');
title('HTC Refrigerant for Highest COP', 'FontSize', 12, 'FontName', '
    Times New Roman');
colormap(lines(length(unique_HTC)));
colorbar('Ticks', 1:length(unique_HTC), 'TickLabels', cellstr(unique_HTC)
    );
set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');
print(gcf, 'HTC_Refrigerant_Selection.pdf', '-dpdf', '-bestfit');

% Plot LTC natural refrigerants
figure;
imagesc(unique(n_T_source), unique(n_T_sink), Z_ref_LTC_nat);
set(gca, 'YDir', 'normal', 'FontName', 'Times New Roman');
xlabel('Heat Source Temperature ( C )', 'FontSize', 12, 'FontName', '
    Times New Roman');
ylabel('Heat Sink Temperature ( C )', 'FontSize', 12, 'FontName', 'Times
    New Roman');
title('LTC Natural Refrigerant for Highest COP', 'FontSize', 12, '
    FontName', 'Times New Roman');
colormap(lines(length(unique_LTC_nat)));
colorbar('Ticks', 1:length(unique_LTC_nat), 'TickLabels', cellstr(unique_
    LTC_nat));
set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');
print(gcf, 'LTC_Natural_Refrigerant_Selection.pdf', '-dpdf', '-bestfit');

% Plot HTC natural refrigerants
figure;
imagesc(unique(n_T_source), unique(n_T_sink), Z_ref_HTC_nat);
set(gca, 'YDir', 'normal', 'FontName', 'Times New Roman');
xlabel('Heat Source Temperature ( C )', 'FontSize', 12, 'FontName', '
    Times New Roman');
ylabel('Heat Sink Temperature ( C )', 'FontSize', 12, 'FontName', 'Times
    New Roman');
title('HTC Natural Refrigerant for Highest COP', 'FontSize', 12, '
    FontName', 'Times New Roman');
colormap(lines(length(unique_HTC_nat)));
colorbar('Ticks', 1:length(unique_HTC_nat), 'TickLabels', cellstr(unique_
    HTC_nat));
set(gca, 'FontSize', 12, 'FontName', 'Times New Roman');
print(gcf, 'HTC_Natural_Refrigerant_Selection.pdf', '-dpdf', '-bestfit');
```