OSLO METROPOLITAN UNIVERSITY
STORBYUNIVERSITETET

OsloMet – Oslo Metropolitan University
Department of Civil Engineering and Energy Technology
P.O. box 4 St. Olavs plass, 0130 Oslo
Pilestredet 35, Oslo

Phone: 67 23 50 00
`www.oslomet.no`

THESIS/CANDIDATE NO.	21
AVAILABILITY	Open

MASTER'S THESIS

TITLE Implementation of Revit add-in for analysis of heating plant P&ID	DATE June 15, 2020 NO. PAGES/APPENDICES 80/5
AUTHOR Christian Velde Steneng	SUPERVISOR Arnab Chaudhuri

IN COLLABORATION WITH Erichsen & Horgen AS	CONTACT PERSON

ABSTRACT
This thesis reviews the possibility of creating an add-in for Revit by the use of the Revit API to analyze a heating plant P&ID. The programming language used to create the add-in is C#.
The add-in contains methods for dynamically calculations of COP and power output in accordance with SN-NSPEK 3031:2020 appendix K. The add-in is calculating and automatically setting the needed insulation thickness and inner diameter for the pipes in the system.
The results show that C# is a robust programming language to create an add-in for Revit, with few restrictions. The add-in created in this thesis makes it is possible to analyze a heating P&ID and investigate optimal combinations of regulation and temperatures for the system.

3 KEYWORDS

Revit API

Energy balance

Heating plant

Preface

First and foremost, I would like to thank my supervisor, Arnab Chauduri, for his time, enthusiasm, and all the valuable advice. Thanks to Thor Even Tutturen for your time, advice, and friendship during this period. I want to thank Anders Møllevik Jacobsen for giving me this challenging and exciting task. I would also like to thank my family for their support and genuine interest in my project. Lastly, I would like to thank my partner Marie for her time, support, and feedback.

Abstract

This thesis reviews the possibility of creating an add-in for Revit by the use of the Revit API to analyze a heating plant P&ID. The programming language used to create the add-in is C#.

The add-in contains methods for dynamically calculations of COP and power output in accordance with SN-NSPEK 3031:2020 appendix K. The add-in is calculating and automatically setting the needed insulation thickness and inner diameter for the pipes in the system.

The results show that C# is a robust programming language to create an add-in for Revit, with few restrictions. The add-in created in this thesis makes it is possible to analyze a heating P&ID and investigate optimal combinations of regulation and temperatures for the system.

Contents

List of Figures

List of Tables

Nomenclature

Physics Constants

p^*	Pressure constant for heat capacity calculation	$16.53MPa$
R	Specific gas constant	$0.461526kJ/(kg\cdot K)$
T^*	Temperature constant for heat capacity calculation	$1386K$

Letters

C_p	Isobaric heat capacity	$kJ/(kg\cdot K)$
D	Diameter	m
f	Friction factor	$-$
h	Specific enthalpy	kJ/kg
$\dot{m}$	mass-flow rate	kg/s
mo	Month number	$-$
n	Type of heat rejector	
P	Pressure	Pa
$\dot{Q}$	Heat transfer rate	W
Re	Reynolds number	$-$
T	Temperature	$^{\circ}C(orK)$
t	hour	h
$\dot{V}$	volumetric-flow rate	m^3/s
v	Velocity	m/s
V	Volume	m^3

Greek letters

ϵ	Roughness	mm
κ	Index for decision of pump usage	$-$
λ	Thermal conductivity	W/mK
μ	Dynamic viscosity	$Pa \cdot s$
ω	Number of hours pump shall be used	h
Φ	Heat loss per meter	W/m
ϕ	Phase change	$-$

Subscripts

adj	Adjusted
amb	Ambient
amp	Amplitude
avg	Average
h	High
HP	Heat pump
i	Counter
in	Inside
inp	User input
l	Low
$NO.$	Number
o	Outdoor
O	Outer
r	Return
S	Source
su	Supply
WDT	Winter design temperature

Chapter 1

Introduction

1.1 Background

In 2015 the United Nations composed a list with 17 sustainable development goals to extinct poverty and slow down the climate changes within the year 2030. Amongst these goals are goal 11 “Sustainable cities and communities” and goal 13 “Climate action” [1]. According to Statistics Norway, 0.8 billion metric tons of CO_2 was emitted from heating of buildings directly to the air in 2018[2].

In a PhD-thesis written by Markusson, it is stated that “Pumps account for 20% of the electrical energy use for electric motors in the world” [3]. The author also expressed,

> *“In Sweden about 2 TWh per year of electrical energy is used for pumps in residential, commercial and public buildings, that corresponds to about 3% of the total electric energy use in residential, public and commercial buildings”.*

In recent years there has been a call for more precise sizing of components for heating and cooling plants in buildings. The thesis work of Hansen 2016, reported that the building from the case-study, “Malmskriverveien 4” in Bærum, had a lot higher installed power than what was ever used in the building [4].

Due to the constant development of tools for designing buildings, like building information modeling software and high precision energy design tools like IDA-ICE, it should be possible to estimate and predict how to reduce the oversizing of components and lower the energy usage and emissions from a building.

Usually, an engineering approach means to simplify whatever that does not seems like it is too crucial for the result, which may cause inconsistencies between real-world cases and calculated cases. Generally, this approach is often adequate since the simplifications do not affect the results significantly. However, with the advent of software and computers, it should be possible to perform calculations with higher precision. Besides, sometimes a safety-factor is added to the estimates, which causes even higher inconsistencies between the actual values and the calculated.

During the design phase of a building, it is normal to use a building information modeling software, often called “BIM”. While there are several different software like this, Revit is a standard tool for this purpose. Revit makes it possible to design the building in 3D, which is very useful for the cooperation between the different branches in a project. A high-quality building model should have a very high level of detail, which means that these models should be as close as possible to the actual building. With a model this close to reality, it should be possible to utilize exact calculations.

When calculating the energy and power demand of a building, it is, however, usual to utilize an energy design tool. These tools are not the same as a BIM program nevertheless, these programs are sometimes able to import a 3D model of the building through the open-source file type called ".IFC". While it is possible to import all the information from a BIM tool to an energy design tool, a lot of "unnecessary" components are removed from the model to make the import and calculations faster.

An example of an energy design tool is "IDA-ICE". IDA-ICE is a highly accurate tool, which will produce detailed results if the model set up is correct. However, it does not consider the actual set-up of the heating plant. Furthermore, it might be different persons that are designing the heating plant than the persons that are utilizing the energy calculations. While the energy and power demand are properties that are dependent on specific criteria, the behavior of the heating plant is in a way only dependent of the power demand and the outside temperature. While figure 1.1 is highly simplified, these are the only inputs that are required in order to perform such calculations.

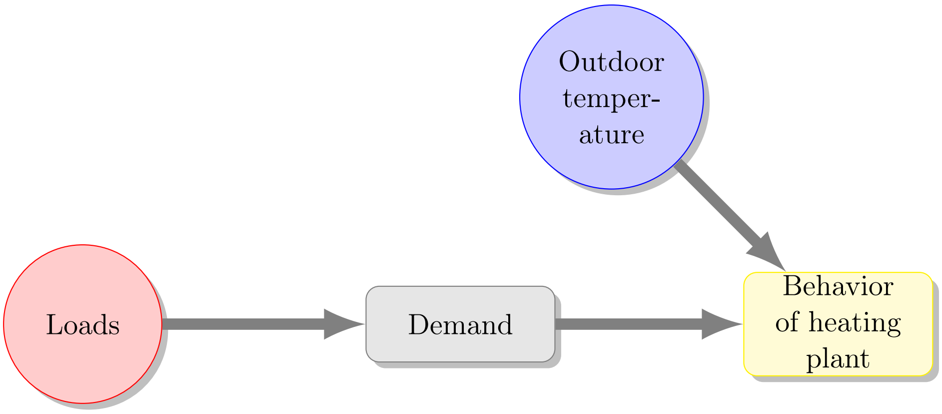


Figure 1.1: Dependencies for the heating plant

Since the loads and demand can be calculated in an energy design tool, it should be possible to simulate the behavior of the heating plant in Revit with the right inputs. This might reduce bad design, oversizing, and make it possible for the designer to evaluate the setup and read the results quickly and seamlessly.

Revit has an application programming interface (API), which allows the user to create instructions for the core software to execute. This makes it possible to create external commands (Add-ins) that operates according to the developers' instructions.

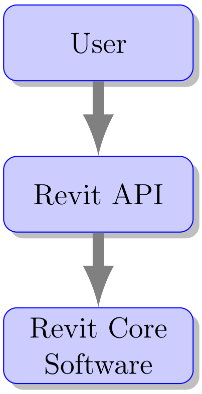


Figure 1.2: Connection between the user, the Revit API and the Revit core software [5]

1.2 Problem Statement

In this work, we aim to develop an "add-in" so that the following question can be addressed.

> *Is there any way to calculate and show the energy-flow in a heating plant for buildings with high precision in Revit?*

1.3 Tools

1.3.1 Requirements

To create an add-in, it is crucial to know what such a task requires. It is also sensible to develop a list of requirements that the result should meet. Listed below are the requirements set for this project.

- It needs to be user-friendly.
- It needs to have a high precision
- The developed software needs to be written in a way that is easily readable for others and modular in nature so it can be further developed by other persons than the writer.

To make the program user-friendly, it is a target that the user should insert and perform as few clicks or writings as possible. Nonetheless, some inputs will be required from the user in order not to make a program suitable for only one case.

To simulate the behavior and energy flow in a heating plant, precision is of utmost importance. The tool will have little to none value if the result is far off from what is happening in the system during operation.

Chapter 2

Governing equations and method

In the following chapter the governing equation and the method will be presented.

2.1 Physical properties and mathematics

The physical properties of the materials included in the system will have a significant impact on the calculation. Any error from the fundamental equations will follow and impair the result. To reduce error, there have been taken several actions to precisely calculate the physical properties

2.1.1 Energy balance

To create a mathematical model of a heating plant one can consider a basic system in steady-state and perform an energy balance of the system, as shown in equation 2.1.

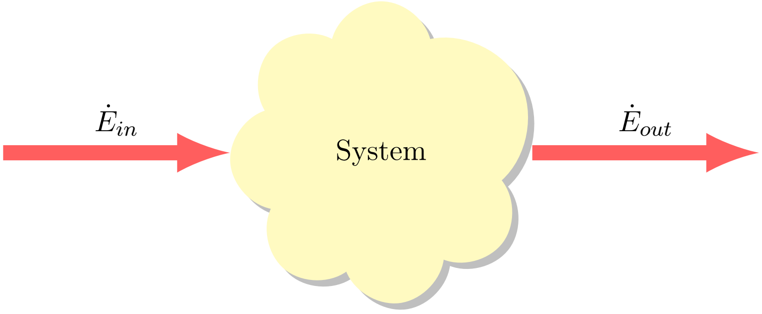


Figure 2.1: Energy balance of a simple steady state system.

$$\dot{E}_{in} = \dot{E}_{out} \tag{2.1}$$

In this case, $\dot{E}_{out}$ represents the energy demand of the building, in terms of heating or cooling and $\dot{E}_{in}$ represents the energy delivered to the building by the heat- or cooling source. This means that the energy coming to the system is the same as the energy leaving the system. Equation 2.1 can be further expanded to include stored energy as presented in equation 2.2.

$$\dot{E}_{in} - \dot{E}_{out} = \dot{E}_{stored} \tag{2.2}$$

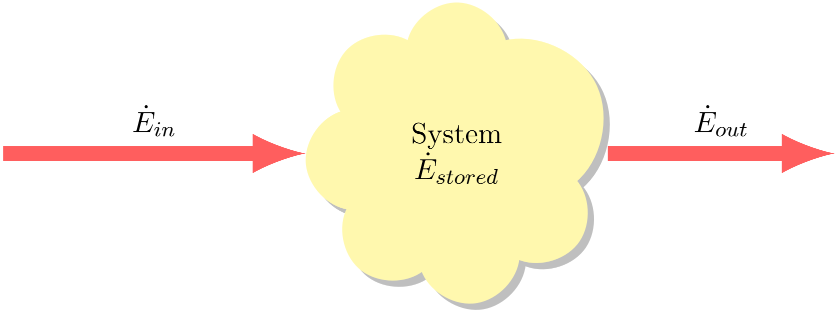


Figure 2.2: Energy balance of a simple system with energy storage.

In this system, there is still no energy transferred to the surroundings. This means that the system is, in a way, a perfect system. In reality, there is always a transfer of energy to the surroundings, which means that it is essential to calculate this transfer as well.

2.1.2 Power

To create a system that represents a heating plant for a building, pumps and heat sources should be added to the system. By using the system from figure 2.2, it is also possible to calculate both the power output and input of the system.

To calculate the required mass flow-rate of water in a system, it is crucial to know the power- demand and the temperature difference over the component.

$$\dot{Q} = \dot{m} C_p \Delta T \tag{2.3}$$

Equation 2.3 shows that there are possibly four unknown factors to calculate the power output of the system. Fortunately, the supposed temperature difference is usually known, and the required power output should be calculated in an energy design program like IDA-ICE. The specific heat capacity for water will be further explained in subsection 2.1.3. This makes the mass-flowrate of the fluid the only unknown factor in this equation.

The change of the temperature difference specific for each component will be further explained in section 2.2.

2.1.3 Specific heat capacity for water at constant pressure

The specific heat capacity of a system is normally defined as how much energy that is required to raise the temperature in the system by one Kelvin. This is given by,

$$C = \frac{Q}{\Delta T} \tag{2.4}$$

By considering a more specific system, for example, one kilogram of water, one can limit the system to a particular volume or specific pressure, which makes it possible to measure how much energy is needed to raise a certain mass of the system with one Kelvin.

By using this definition, it becomes clear that the specific heat capacity for water at constant pressure varies with both temperature and pressure. As mentioned in section 2.1, it is a requirement for the program that the calculations should consider such dependencies. This means that it is a need for obtaining the isobaric heat capacity for water at each temperature and each pressure. Typically, these values are tabulated values. In order to implement such values to a program, the tables are to be loaded and accessible. Furthermore, it would be necessary to interpolate for each value that is not listed. This way of obtaining the specific heat capacity can both be time-consuming and, depending on the table; the error might be substantial. It would also require tables for each value of pressure.

However, “The International Association for the Properties of Water and Steam” has developed a method to calculate different properties for water by using Gibbs free energy.

This resulted in equations for five different regions. The validity of region 1 is presented in equation 2.5 and 2.6 [6]. Since these values are well within the range of pressure and temperatures of a conventional plumbing system, there is no need for implementing equations of other regions [7].

$$273.15K \leq T \leq 623.15K \tag{2.5}$$

$$P_{sat}(T) \leq p \leq 100MPa \tag{2.6}$$

The IAPWS fundamental equation for Gibbs free energy is

$$\frac{g(p,T)}{RT} = \gamma(\xi,\tau) = \sum_{i=1}^{34}(7.1-\xi)^{I_i}(\tau-1.222)^{J_i} \tag{2.7}$$

By second order partial derivation with respect to τ, the formula becomes

$$\gamma_{\tau\tau} = \left[\frac{\partial^2\gamma}{\partial\tau^2}\right]_{\xi} \tag{2.8}$$

Which is the equation for isobaric heat capacity for water.

The governing equations for calculation of isobaric heat capacity for water is then given by

$$\frac{C_p(\xi,\tau)}{R} = -\tau^2\gamma_{\tau\tau} \tag{2.9}$$

With,

$$\gamma_{\tau\tau} = \sum_{i=1}^{34} n_i(7.1-\xi)^{I_i} J_i(J_i-1)(\tau-1.222)^{J_i-2} \tag{2.10}$$

Where,

$$\xi = \frac{p}{p^*} \tag{2.11}$$

$$\tau = \frac{T^*}{T} \tag{2.12}$$

For region 1, p^* is 16.53 MPa, T^* is 1386K. Values for n_i, I_i, J_i and i are given in appendix A.

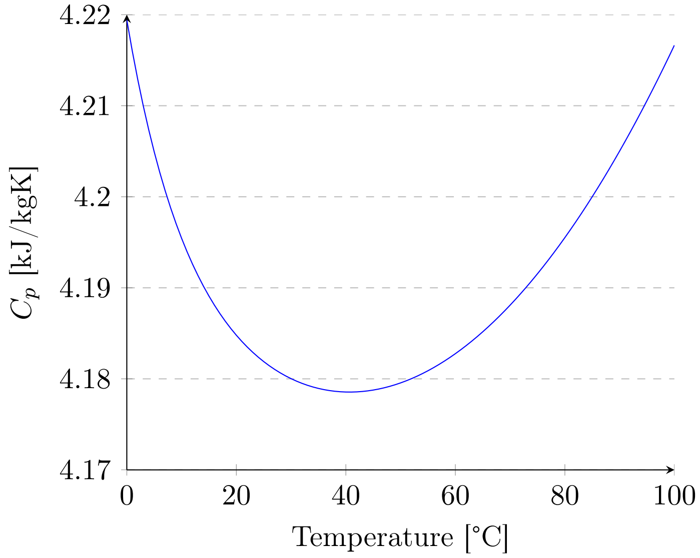


Figure 2.3: Plot of the specific heat capacity for H_2O with pressure at 1 atm and temperature ranging from 0°C to 100°C

Figure 2.3 shows the results of the calculations using the equations presented with pressure at 1 atm and temperature ranging from 0°C to 100 °C. As expected, the heat capacity is varying. It is, however, essential to note the scale, which shows that the difference is minimal. The result and the consequences will be further discussed in chapter 4. As the isobaric heat capacity is varying, equation 2.3 can be rewritten to

$$\dot{Q} = \sum_{i=T_{low}}^{T_{high}} \dot{m} C_p(i,p) \tag{2.13}$$

2.1.4 Dynamic viscosity for water

The dynamic viscosity for water is dependent on both temperature and pressure. However, pressure has a shallow impact. In the book "Fluid dynamics" by Çengel and Cimbala, it is stated that "For liquids, the dynamic viscosity is practically independent of the impact of pressure except at extremely high pressures" [8]. Since the pressures at the heating plant usually never exceed 10atm, it is possible to calculate the dynamic viscosity for water at a certain temperature by using the formula for dynamic viscosity for water at 1atm by

$$\mu = a10^{b/(T-c)} \tag{2.14}$$

For water, the values $a = 2.414 \times 10^{-5} N \cdot s/m^2$, $b = 247.8K$ and $c = 140K$ [8].

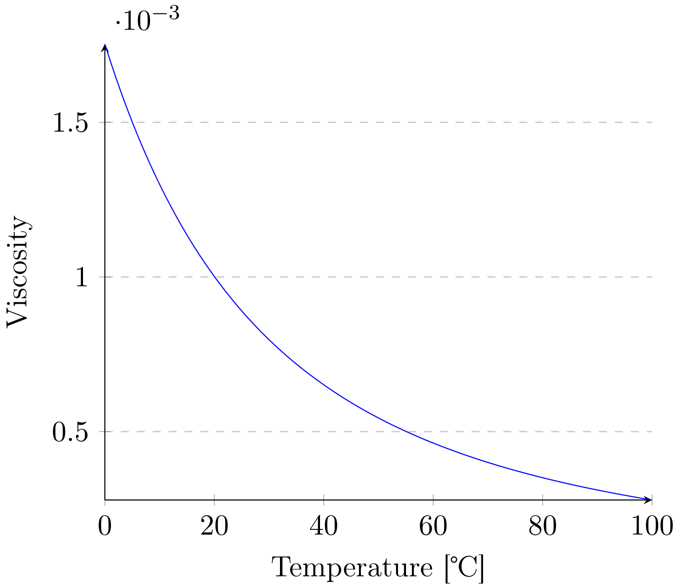


Figure 2.4: Dynamic viscosity for H_2O at temperature ranging from 0°C to 100°C

2.1.5 Density for water

The density for a substance is usually defined as how much a certain amount of volume the substance weights. Commonly, this is defined as how many kilograms the substance weights per cubic meter as

$$\rho = \frac{m}{V} \tag{2.15}$$

Since temperature can affect the volume of substances, the density of a material is temperature-dependent. By knowing this, it becomes essential to calculate the density for water for a specific temperature. Kell [9] has developed a formula to calculate the density for air-free water at 1 atm. As previously stated, water is incompressible, and the variation of density caused by pressure is minimal [8]. This makes it safe to apply the formula developed by Kell to this application as,

$$\begin{aligned}\rho_{wtr} = {} & (999.83952 + 16.945176T - 7.9870401 \times 10^{-3}T^2 - 46.170461 \times 10^{-6}T^3 \\ & + 105.56302 \times 10^{-9}T^4 - 280.54253 \times 10^{-12}T^5)/(1 + 16.897850 \times 10^{-3}T)\end{aligned} \tag{2.16}$$

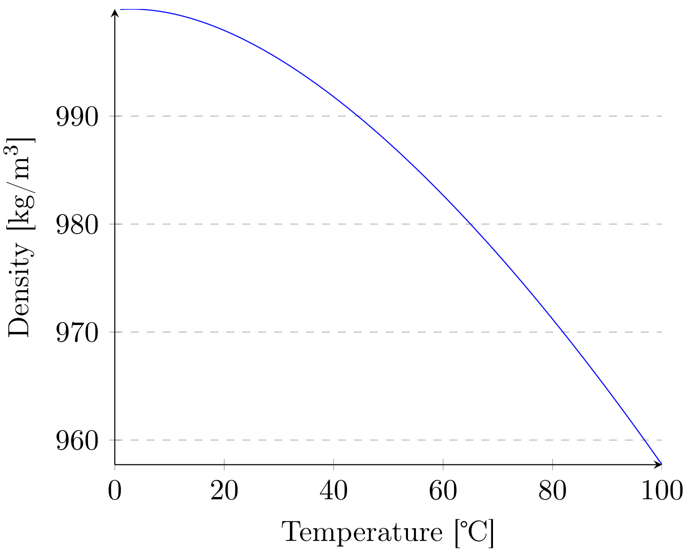


Figure 2.5: Density for H_2O at 1 atm with temperature ranging from 0℃ to 100℃

However, if the pressure should be considered, as with isobaric heat capacity, the IAPWS relation is also applicable to calculate the density for water at a given pressure.

$$V = \left[\frac{\partial g}{\partial T}\right]_T \tag{2.17}$$

By partial derivation with respect to ξ,

$$\gamma_\xi = \left[\frac{\partial \gamma}{\partial \xi}\right]_\tau \tag{2.18}$$

Which leads to

$$V(\xi, \tau)\frac{p}{RT} = \xi\gamma_\xi \tag{2.19}$$

$$\gamma_\xi = \sum_{i=1}^{34} -n_i I_i (7.1 - \xi)^{I_i - 1}(\tau - 1.222)^{J_i} \tag{2.20}$$

Lastly,

$$\rho_{wtr} = \frac{1}{V} \tag{2.21}$$

2.1.6 Heat transfer to surroundings

As explained in section 2.1.1, there will always be a certain amount of energy transfer to the surroundings of the system. If the system is considered as the actual heating plant, and the surroundings, in this case, is the volume of the building, there is intentional heat transfer. However, during the transportation of the fluid, it is never desirable to transfer heat from for example pipes to the surroundings. To reduce this heat transfer, a normal practice is to insulate the pipes where the fluid is transferred. The heat transfer from the pipes to the surroundings per meter can be calculated, as shown in equation 2.22.

$$U' = \frac{\pi}{\frac{1}{D_i h_i} + \sum\left(\frac{ln(D_{k+1}/D_k)}{2\lambda_k}\right) + \frac{1}{D_o h_o}} \tag{2.22}$$

Depending on the pipe diameter, thickness, material, and insulation type, it is possible to calculate the heat loss per meter length of the pipe by

$$\Phi = U'(\Delta T) \tag{2.23}$$

Where ΔT is the temperature difference between the ambient air and the fluid in the pipe, by using this relation, it is possible to calculate the needed insulation thickness for a pipe.

From the buildings and regulations act, it is stated that

> *"Pipes, equipment and ducts connected to the building's heating system shall be insulated. The thickness of the insulation shall be economically optimal, calculated in accordance with a Norwegian standard or an equivalent European standard" [10]*

As stated, the standard NS-EN 12828:2012+A1:2014 can be used. In the standard, certain requirements for different types of pipes are listed [11]. From class 5, the requirements to the insulation thickness are

$$U' \leq \begin{cases} 1.5D_o + 0.16(W/mK) & D_o \leq 0.4m \\ 0.49(W/m^2K) & otherwise \\ \end{cases} \tag{2.24}$$

Notice the difference in the units in equation 2.24.

2.1.7 Pressure loss

Apart from the heat transferred to the surroundings, there is also another factor that needs to be considered when designing a heating plant. This factor is pressure loss due to friction inside the pipes.

Generally, the pressure loss can be described as shown in equation 2.25.

$$\Delta P_L = f\frac{L}{D}\frac{\rho V^2}{2} \tag{2.25}$$

As shown in the equation, there are several factors that needs to be calculated. First, the average velocity in the pipe needs to be calculated which is done by,

$$v_{avg} = \frac{\dot{V}}{\pi D^2/4} \tag{2.26}$$

When the average velocity of the fluid is known, the Reynolds number is calculated by,

$$Re = \frac{\rho v_{avg} D}{\mu} \tag{2.27}$$

The result of the Reynolds number can be used to determine if the flow is laminar or turbulent. As a general rule, the flow is laminar if the Reynolds number is $\lesssim 2300$ and turbulent if the Reynolds number is $\gtrsim 4000$ [8].

When the flow is laminar, the friction factor is independent of the roughness of a pipe and can be expressed as

$$f = \frac{Re}{64} \tag{2.28}$$

It is, however, rare that laminar flow occurs within a system with a relatively high volumetric flow rate and water as fluid. Therefore it becomes vital to calculate the friction factor for turbulent flow as well. When the flow is turbulent, the friction factor can be calculated by

$$\frac{1}{\sqrt{f}} = -2.0\log\left(\frac{\epsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right) \tag{2.29}$$

As seen, equation 2.29 is implicit with f on both sides of the equation. This means that in order to solve the equation, it requires iteration. A function for finding the friction factor can then be expressed as

$$F(f) = -2.0 \log\left(\frac{\epsilon/D}{3.7} + \frac{2.51}{Re\sqrt{f}}\right) - \frac{1}{\sqrt{f}} \tag{2.30}$$

It becomes clear that in order to find the roots of the function, some kind of root finding algorithm needs to be applied. The "Bisection method" is applicable in this case. This method is described as

$$x_n = \frac{1}{2}(x_a + x_b) \tag{2.31}$$

The method then checks if $f(x_1)$ is greater or lower than 0, and updates the bound. If $f(x_1) > 0$, $x_a = f(x_1)$ and if $f(x_1) > 0$, $x_a = f(x_1)$ [12]. To apply this method into an algorithm, it becomes clear that this method also needs a stopping condition in order to break the loop. In the book "Applied Mathematics and Modeling for Chemical Engineers" by Rice and Do, it is stated the drawback of this method is that product of the first guesses needs to be negative. It is also stated that the rate of convergence is slow compared to other methods. On the other hand, they argue that "The method is very simple" and "The method always converges" [13].

To apply this algorithm, it is clear that some initial values for x_a and x_b is needed. Figure 2.6 shows the solution of equation 2.30 for a steel pipe with a roughness of 0.002mm, diameter of 0.05m, and water at 20°C. By varying the f from 10^{-5} to 1, it becomes clear that these values will fulfill the criteria for using the "Bisection method". According to the plot in figure 2.6, an upper limit of 1 is arguably very high. However, since this is a plot for a specific case, the solution is indeed varying.

In the algorithm, the stopping condition can be described as

```
while abs(xa -xb) > condition
```

A lower condition will result in higher precision, but this also means that the algorithm will have a higher amount of iterations, thus increasing the time it takes to solve the equation.

By applying this algorithm on the same example as above, and a condition of 10^{-7}, the criteria are met within 24 iterations, as shown in table 2.1.

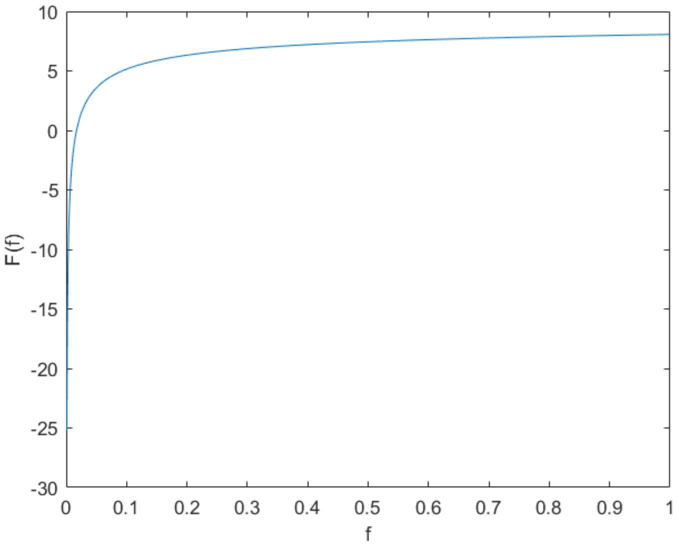


Figure 2.6: Plot of the friction factor

x_a	x_b	$f(x)$
0,0001	1	0,50005
0,0001	0,50005	0,250075
0,0001	0,250075	0,1250875
0,0001	0,1250875	0,06259375
0,0001	0,06259375	0,03134687
0,0001	0,03134687	0,01572344
0,01572344	0,03134687	0,02353516
0,01572344	0,02353516	0,0196293
0,01572344	0,0196293	0,01767637
0,01572344	0,01767637	0,0166999
0,0166999	0,01767637	0,01718813
0,0166999	0,01718813	0,01694402
0,01694402	0,01718813	0,01706608
0,01706608	0,01718813	0,01712711
0,01712711	0,01718813	0,01715762
0,01715762	0,01718813	0,01717288
0,01717288	0,01718813	0,01718051
0,01718051	0,01718813	0,01718432
0,01718051	0,01718432	0,01718241
0,01718241	0,01718432	0,01718337
0,01718337	0,01718432	0,01718384
0,01718384	0,01718432	0,01718408
0,01718408	0,01718432	0,0171842
0,01718408	0,0171842	0,01718414

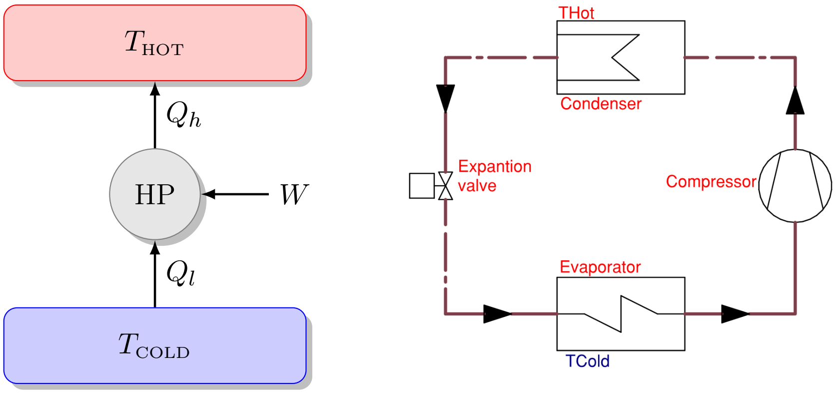
Table 2.1: Results from the test case using the Bisection algorithm

2.2 Components

This section will introduce the various components that are used for calculation in the program. This list is shortened, for the reason that there are a lot of possibilities on how to construct such a system. For this reason the components below are the components that are considered as crucial in terms of calculating the energy flow in the system.

2.2.1 Heat pump

The heat pump's task is to raise the water temperature in the system. Generally, a heat pump is a reasonable way to convert electrical energy to thermal energy due to the coefficient of power. Because of the working principle of a heat pump, it makes it possible to achieve a higher amount of thermal energy delivered to the system than what is required by electrical work.



(a) Working principle of a heat pump (b) P&ID of a heat pump

Figure 2.7: Principle of heat pump and diagram for heat pump

As shown in figure 2.7a, the heat pump needs a cold and a hot reservoir in order to work. By knowing the temperatures of the reservoirs, it is possible to calculate the Carnot coefficient of power, as shown in equation 2.32.

$$COP_{h,rev} = \frac{1}{1 - \frac{T_l}{T_h}} \tag{2.32}$$

Equation 2.32 makes it possible to calculate the coefficient of power and the needed power input to convert the desired output. However, the Carnot coefficient of power is highly theoretical, and the calculated COP will always be higher than the actual values.

To incorporate real values, the enthalpy of the working fluid can be used. By knowing the change of enthalpy of the working fluid over the condenser and the compressor, the COP of a heat pump can then be expressed as shown in equation 2.33 [14].

$$COP = \frac{\Delta h_{condenser}}{\Delta h_{compressor}} \tag{2.33}$$

Enthalpy is dependent on both temperature, pressure, and the phase of the fluid. In addition to this, the enthalpy is varying with different working fluid. This makes a precise calculation of the COP almost an impossible task since there are many different variables.

Another way of estimating the COP of a heat pump is described in SN-NSPEK 3031:2020, appendix K. In this approach, there are pre-defined setpoints with adjacent values for temperature, COP, and power performance. The method in this specification interpolates the values when the result is in between the setpoints. It is also stated that when the source temperature is lower than the lowest setpoint, the power should be set to zero, and COP is set to the lowest COP. The same applies to both power and COP when the source temperature is higher than the highest set point, except that the power will be the highest value, instead of zero.

In the standard, it is stated that this method should be used in the early phase of a project, which is what this tool is meant for.

The specification differs between types of heat pumps and how they are operating. Generally, it varies between ON/OFF operating heat pumps, inverter operated heat pumps, as well as what kind of reservoir the heat pump uses. As a simplification, ON/OFF operating heat pumps are analyzed. As cold reservoir, both boreholes and the outside air will be implemented in the program.

The relation for heat pumps with boreholes as source and ON/OFF operation is presented in table 2.2.

T_S	T_{s1} = -5°C		T_{s2} = 0°C		T_{s3} = 5°C	
T_{su}	P_{HP}	COP	P_{HP}	COP	P_{HP}	COP
T_{su} = 35°C	$0.92 \times P_{HP,inp}$	$0.78 \times COP_{inp}$	$P_{HP,inp}$	COP_{inp}	$1.15 \times P_{HP,inp}$	$1.10 \times COP_{inp}$
T_{su} = 55°C	$0.82 \times P_{HP,inp}$	$0.43\times COP_{inp}$	$0.91 \times P_{HP,inp}$	$0.58 \times COP_{inp}$	$1.00 \times P_{HP,inp}$	$0.73 \times COP_{inp}$

Table 2.2: Values for the relation of power, COP and temperature for heat pump with different source temperatures from borehole and ON/OFF operation [15]

According to the table, the COP and the power output are dependent on both the source temperature and the supply temperature. The source temperature for boreholes can be found by using the formula for calculating the supply temperature of the borehole, which is shown in equation 2.34.

$$T_S = T_{S,avg} - T_{S,amp} \cos\left[\frac{2\pi}{12}(mo - \phi - 1)\right] \tag{2.34}$$

In this case, the average temperature of the borehole, and the amplitude temperature depends on the water flow, the use of the well and the average temperature for the location of the borehole. The quantity ϕ is described as “phase cange” and should be set to zero unless otherwise known. These values are described in SN-NSPEK 3031:2020.

It is stated in the standard that July is supposed to be the hottest month and January is the coldest [15], which corresponds to the results in figure 2.8. After knowing the source temperature, the maximum output power from the heat pump can be calculated.

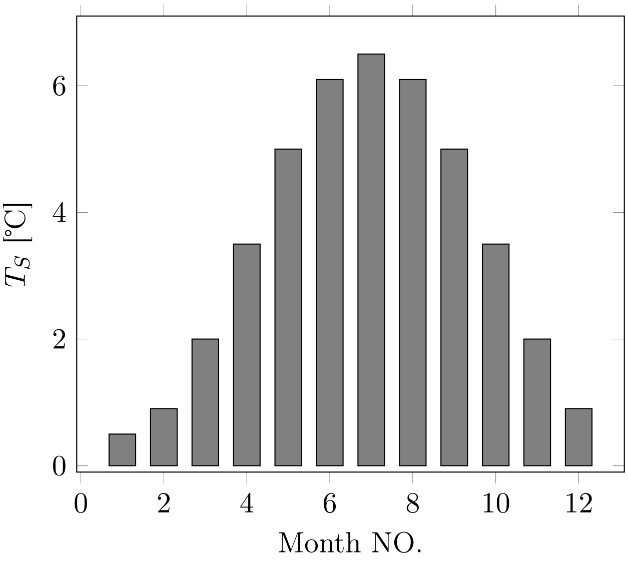


Figure 2.8: Supply temperature for borehole with average temperatures over 5°C, for heating, undersized borehole and large stream of ground water

As previously mentioned, this needs to implement linear interpolation given by

$$y = y_0 + (x - x_0)\frac{(y_1 - y_0)}{(x_1 - x_0)} \tag{2.35}$$

To calculate the power output, the interpolation is done in three steps. The first step calculates the power output for supply temperatures at 35°C. Because the calculations are performed on ON/OFF operating heat pumps, and are performed hourly, the power output is an average of the power output from the heat pump. In reality, the heat pump will deliver max power, and turn itself on and off within this hour. However, this simplification is done due to how much larger the calculations needs to be in order to consider minutes or possibly seconds.

$$P_{HP,35}(t) = \begin{cases} 0.92P_{HP,inp} + (T_S(t)+5)\dfrac{(P_{HP,inp} - 0.92P_{HP,inp})}{5} & -5 \leq T_S(t) < 0 \\ P_{HP,inp} + (T_S(t))\dfrac{(1.15P_{HP,inp} - P_{HP,inp})}{5} & 0 \leq T_S(t) < 5 \\ 1.15P_{HP,inp} & T_S(t) \geq 5 \\ 0 & otherwise \end{cases} \tag{2.36}$$

After the power output of 35°C is calculated, the next step is to calculate the power output with supply temperature at 55°C.

$$P_{HP,55}(t) = \begin{cases} 0.82P_{HP,inp} + (T_S(t)+5)\dfrac{(0.91P_{HP,inp} - 0.82P_{HP,inp})}{5} & -5 \le T_S(t) < 0 \\ P_{HP,inp} + (T_S(t))\dfrac{(P_{HP,inp} - 0.91P_{HP,inp})}{5} & 0 \le T_S(t) < 5 \\ P_{HP,inp} & T_S \ge 5 \\ 0 & otherwise \end{cases} \tag{2.37}$$

When the two powers are known, the last step is to calculate the power output with the desired supply temperature

$$P_{HP}(t) = \begin{cases} P_{HP,35}(t) + (T_{su}(t) - 35)\dfrac{(P_{55}(t) - P_{35}(t))}{(35-55)} & 35 < T_{su} < 55 \\ P_{HP,35} & T_{su} \le 35 \end{cases} \tag{2.38}$$

The COP for the heatpump is calculated in the same way as the power, except that according to the standard, the COP will not drop to zero when the source temperature is below 5°C.

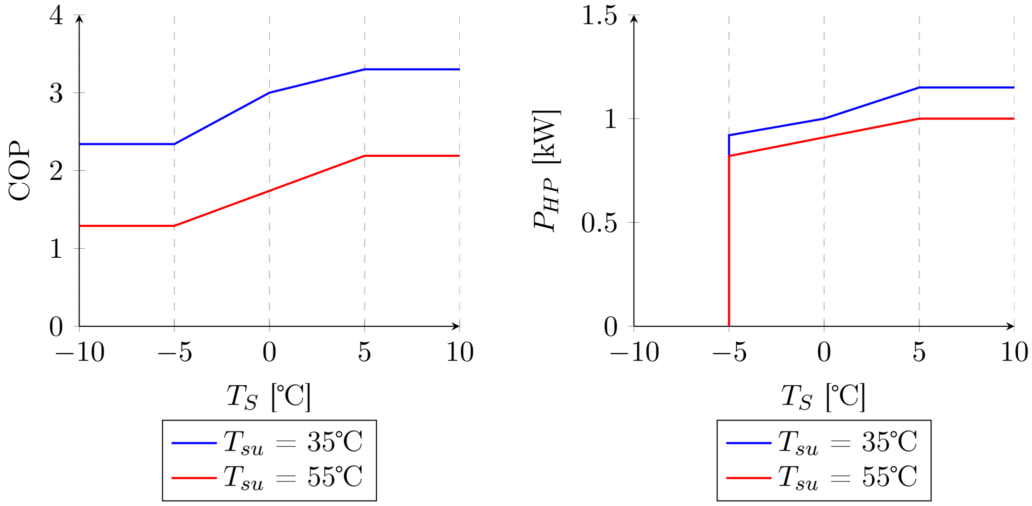
$$COP_{35}(t) = \begin{cases} 0.78COP_{inp} + (T_S(t)+5)\dfrac{(COP_{inp} - 0.78COP_{inp})}{5} & -5 \le T_S(t) < 0 \\ COP_{inp} + (T_S(t))\dfrac{(1.1COP_{inp} - COP_{inp})}{5} & 0 \le T_S(t) < 5 \\ 1.1COP_{inp} & T_S(t) \ge 5 \\ 0.78COP_{inp} & otherwise \end{cases} \tag{2.39}$$

$$COP_{55}(t) = \begin{cases} 0.43COP_{inp} + (T_S(t)+5)\dfrac{(0.73COP_{inp} - 0.43COP_{inp})}{5} & -5 \le T_S(t) < 0 \\ 0.73COP_{inp} + (T_S(t))\dfrac{(0.73COP_{inp} - 0.58COP_{inp})}{5} & 0 \le T_S(t) < 5 \\ 0.73COP_{inp} & T_S(t) \ge 5 \\ 0.43COP_{inp} & otherwise \end{cases} \tag{2.40}$$

$$COP_{HP}(t) = COP_{HP,35}(t) + (T_{su}(t) - 55)\frac{(COP_{55}(t) - COP_{35}(t))}{(55-35)} \tag{2.41}$$

$$COP_{HP}(t) = \begin{cases} COP_{HP,35}(t) + (T_{su}(t) - 35)\dfrac{(COP_{55}(t) - COP_{35}(t))}{(55-35)} & 35 < T_{su} < 55 \\ COP_{HP,35} & T_{su} < 35 \end{cases} \tag{2.42}$$

By using this relation, it is possible to calculate the COP and the power-output from the heat pump for any given supply temperature and source temperature. Figure 2.9 shows the relation for a reference COP at 3. As stated in the specification, the COP stays stationary with source temperatures above 5°Cand source temperature below -5°C. Figure 2.9a shows the same relation to power, with a reference power output at 1kW from the heat pump. In this case, when the source temperature drops below minimum source temperature, the power becomes 0. This means that the heat pump should be turned off during these circumstances.



(a) Cop for heat pump with boreholes as source

(b) Power for heat pump with boreholes as source

Figure 2.9: COP and power for heat pump with boreholes as source, plotted after specifications from SN-NSPEK 3031:2020 [15]

When using outdoor air as a source, table 2.3 applies.

T_S	T_{s1} = -15°C		T_{s2} = -2°C		T_{s3} = 7°C	
T_{su}	P_{HP}	COP	P_{HP}	COP	P_{HP}	COP
T_{su} = 35°C	0.55 × $P_{HP,inp}$	0.48 × COP_{inp}	0.73× $P_{HP,inp}$	0.71 × COP_{inp}	$P_{HP,inp}$	COP_{inp}
T_{su} = 55°C	0.44 × $P_{HP,inp}$	0.32× COP_{inp}	0.65 × $P_{HP,inp}$	0.45 × COP_{inp}	0.89 × $P_{HP,inp}$	0.68 × COP_{inp}

Table 2.3: Values for the relation of power, COP and temperature for heat pump with different source temperatures from the outdoor air and ON/OFF operation [15]

Again, the source temperature needs to be known in order to calculate the power-output and the COP for the heat pump. Since the heat-pump uses the outdoor air as reservoir, the outdoor temperature from the input file is used for calculations. An example of the outdoor temperature during a year for Oslo is presented in figure 2.10.

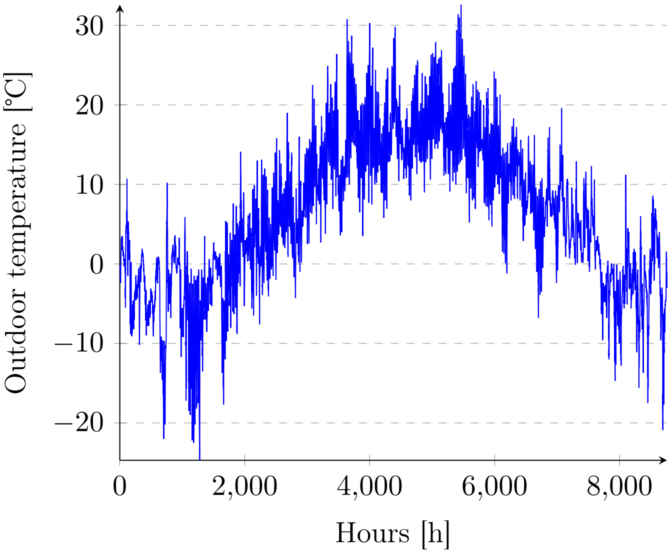


Figure 2.10: Outdoor air temperature for Oslo during a year

Again, the calculations for the COP and the power output follows the same logic as before. Note, however, that the input now is the highest possible output.

$$P_{HP,35}(t) = \begin{cases} 0.55P_{HP,inp} + (T_S(t)+15)\dfrac{(0.73P_{HP,inp} - 0.55P_{HP,inp})}{13} & -15 \leq T_S(t) < -2 \\ 0.73P_{HP,inp} + (T_S(t)+2)\dfrac{(P_{HP,inp} - 0.73P_{HP,inp})}{9} & -2 \leq T_S(t) < 7 \\ P_{HP,inp} & T_S(t) \geq 7 \\ 0 & otherwise \end{cases} \tag{2.43}$$

$$P_{HP,55}(t) = \begin{cases} 0.44P_{HP} + (T_S(t)+15)\dfrac{(0.65P_{HP} - 0.82P_{HP})}{13} & -15 \leq T_S(t) < -2 \\ 0.65P_{HP,inp} + (T_S(t)+2)\dfrac{(0.89P_{HP,inp} - 0.65P_{HP,inp})}{9} & -2 \leq T_S(t) < 7 \\ 0.89P_{HP,inp} & T_S \geq 7 \\ 0 & otherwise \end{cases} \tag{2.44}$$

$$P_{HP}(t) = \begin{cases} P_{HP,35}(t) + (T_{su}(t) - 35)\dfrac{(P_{55}(t) - P_{35}(t))}{(55-35)} & 35 < T_{su} \leq 55 \\ P_{HP,35} & T_{su} \leq 35 \end{cases} \tag{2.45}$$

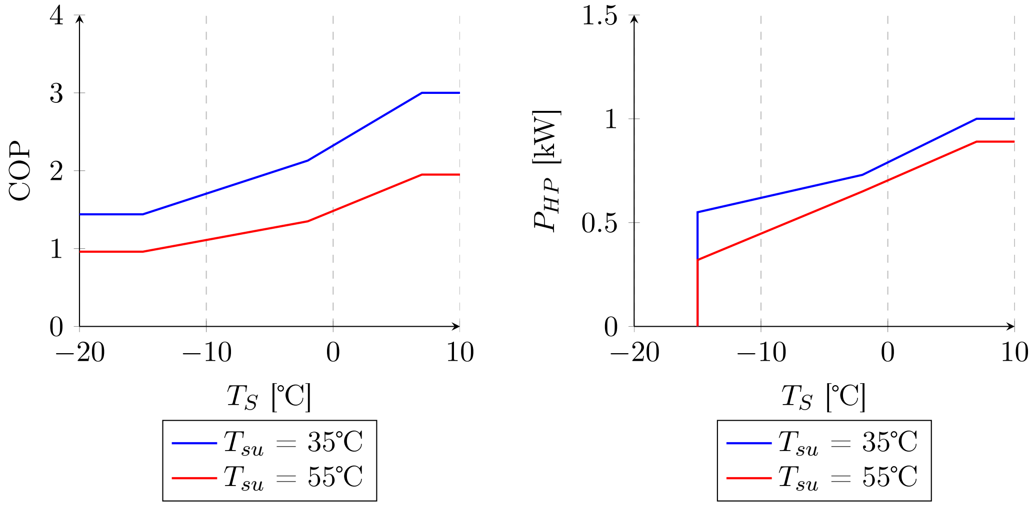
$$COP_{35}(t) = \begin{cases} 0.48COP_{inp} + (T_S(t)+15)\dfrac{(0.71COP_{inp} - 0.48COP_{inp})}{13} & -15 \leq T_S(t) < -2 \\ 0.71COP_{inp} + (T_S(t)+2)\dfrac{(COP_{inp} - 0.71COP_{inp})}{9} & -2 \leq T_S(t) < 7 \\ COP_{inp} & T_S(t) \geq 7 \\ 0.48COP_{inp} & otherwise \end{cases} \tag{2.46}$$

$$COP_{55}(t) = \begin{cases} 0.32COP_{inp} + (T_S(t)+15)\dfrac{(0.45COP_{inp} - 0.32COP_{inp})}{13} & -15 \leq T_S(t) < -2 \\ 0.45COP_{inp} + (T_S(t)+2)\dfrac{(0.68COP_{inp} - 0.45COP_{inp})}{9} & -2 \leq T_S(t) < 7 \\ 0.68COP_{inp} & T_S(t) \geq 7 \\ 0.32COP_{inp} & otherwise \end{cases} \tag{2.47}$$

$$COP_{HP}(t) = COP_{HP,35}(t) + (T_{su}(t) - 35)\frac{(COP_{55}(t) - COP_{35}(t))}{(55-35)} \tag{2.48}$$

$$COP_{HP}(t) = \begin{cases} COP_{HP,35}(t) + (T_{su}(t) - 35)\dfrac{(COP_{55}(t) - COP_{35}(t))}{(55-35)} & 35 < T_{su} \leq 55 \\ COP_{HP,35} & T_{su} \leq 35 \end{cases} \tag{2.49}$$

These calculations lead to a slightly different result, as presented in figure 2.11.



(a) COP for heat pump with outdoor air as source

(b) Power for heat pump with outdoor air as source

Figure 2.11: COP and power for heat pump with outdoor air as source, plotted after specifications from SN-NSPEK 3031:2020 [15]

2.2.2 Outdoor temperature compensation

In the book "Vannbaserte oppvarings- og kjølesystemer", Zijdemans states that by assuming that the power demand for a building is proportional to the outdoor temperature, outdoor temperature compensation curves can be constructed [16]. By doing so, it is possible to raise or lower the supply temperature for the fluid relative to the outside temperature. As seen in section 2.2.1, a lower supply temperature will increase the COP for the heat pump. By following this curve for supply and return temperatures, the system can achieve the same power output with lower temperatures and higher volumetric flow rates. The formula for creating those curves are shown in equation 2.50 and 2.51.

$$T_{su} = T_{in} + \Delta T_{WDT} \left(\frac{T_{in} - T_o}{T_{in} - T_{WDT}} \right)^{\frac{1}{n}} + 0.5(T_{su,WDT} - T_{r,WDT}) \frac{T_{in} - T_o}{T_{in} - T_{WDT}} \tag{2.50}$$

$$T_r = T_{su} - (T_{su,WDT} - T_{r,WDT}) \frac{T_{in} - T_o}{T_{in} - T_{WDT}} \tag{2.51}$$

As seen in the equations above, the needed inputs are the winter design temperature(WDT), supply and return temperature at winter design temperature, the temperature inside the building, and the temperature outside at the given time.

In figure 2.12 such a curve is constructed with WDT at -20°C, a supply temperature at 40°C and a return temperature at 30°C. The setpoint temperature inside the building is 22°C. Appendix B, gives factors for different heat rejectors. However, the factor n for heat pump is not included in the list. As a simplification it is set to 1 on this project.

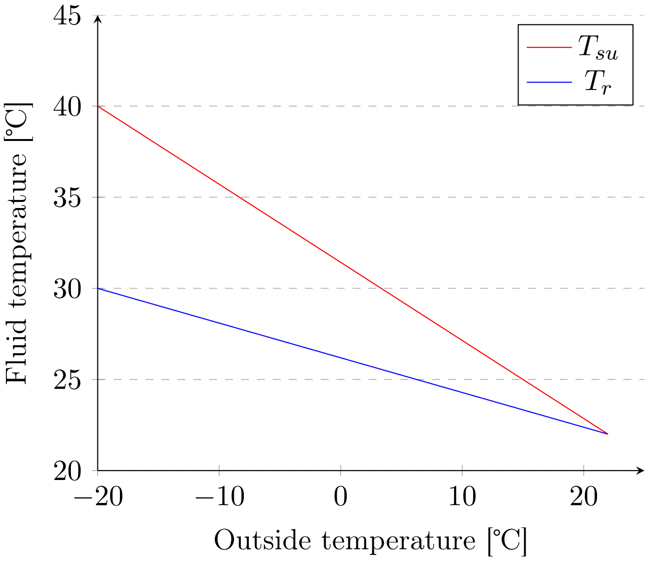


Figure 2.12: Outdoor temperature compensation curve for a heat pump

2.2.3 Pumps

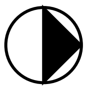


Figure 2.13: Symbol for a pump[17]

The pump is responsible for the water flow in the system. To calculate the absolute minimum needed water flow, one can use equation 2.3 to calculate the mass flow provided by the pump. By knowing the density of the water at the given temperature, it is possible to calculate the volumetric flow over the pump.

According to SN-NSPEK 3031:2020, the needed power for the pump can be calculated with pressure drop for the circuit, volumetric flow, and the total efficiency for the pump [15].

$$P_{pump} = \frac{\Delta p \dot{V}}{\eta_{tot}} \tag{2.52}$$

In the supply circuit, it is sensible to install two pumps with an alternating operation to reduce wear and have a “backup” if one of the pumps should terminate. In the book “Varmenormen” it is stated that pumps can alternate with a cycle of six months [17]. A mathematical description of this alternation can be expressed as,

$$\kappa(t) = \sin\left(\frac{2\pi}{\omega}t\right) \tag{2.53}$$

Where ω is the cycle in hours, in this case, ω is 336, which is the number of hours for two weeks. However, if a specific project should require different cycles, formula 2.53 is still applicable by changing ω to an appropriate number. When $\kappa(t)$ is known, the selection of which pump that should be in operation can be expressed as

$$Pump_{NO.} = \begin{cases} 1 & \kappa(t) \leq 0 \\ 2 & otherwise \end{cases} \tag{2.54}$$

2.2.4 Heat exchanger

Normally, by using a heat exchanger, there is a need for transferring heat without mixing two fluids together. There are a number of examples of types and the usage of heat exchangers. However, in this thesis, the heat exchanger is used for peak load.

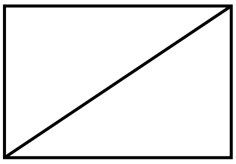


Figure 2.14: Symbol for heat exchanger [17]

2.2.5 Thermal energy storage tanks

To reduce wear on the heat pump, a common practice is to include one or several thermal energy storage tanks to the system. This will add the possibility for the heat pump to turn on and off during load hours.

When the heat pump is in operation, pump 1 will have a higher mass flow-rate than pump 2. This will “push” the hot water to the tank, while the mass-flowrate provided by pump 2 is the actual need for the system. It is worth noticing that during this time of operation, the heat pump will provide a higher power output than what is demanded by the building.

When the heat pump is turned off, the magnetic solenoid is closed. Again this means that pump 1 is not in operation. Pump 2 is supplied with hot water from the thermal energy storage tank.

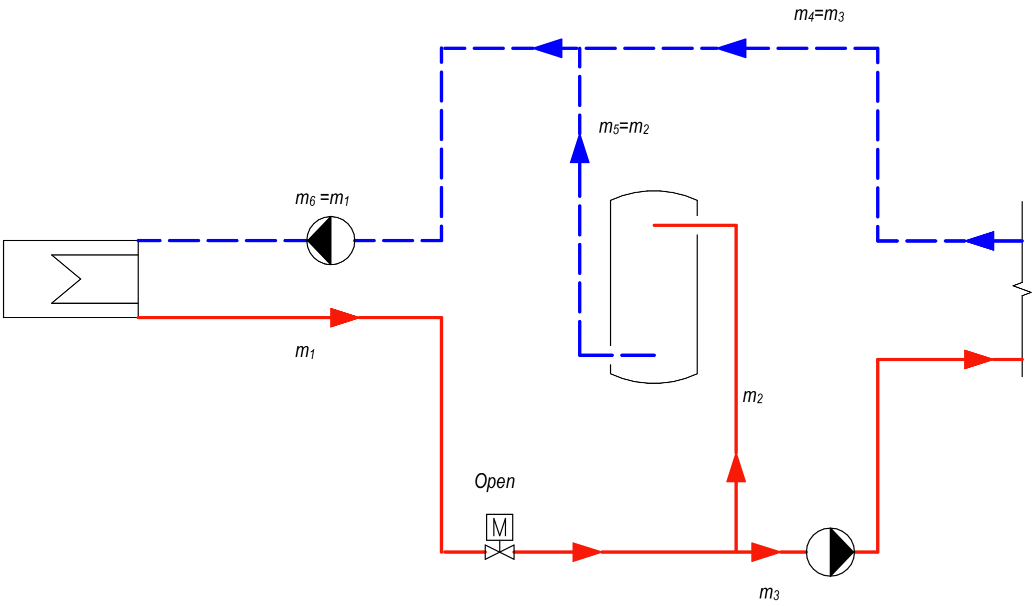


Figure 2.15: Fluid flow when the heat pump is in operation

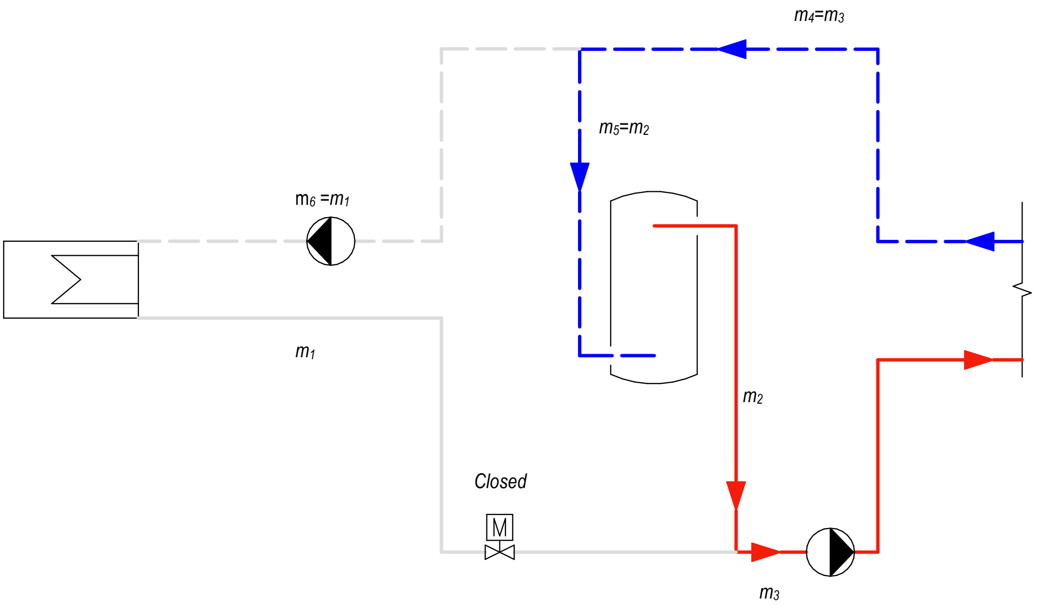


Figure 2.16: Fluid flow when the heat pump is not in operation

2.2.6 The test system

The P&ID used for testing of the program is presented in figure 2.17

The system consists of a heat pump, two supply pumps, one return pump, two magnetic solenoid that switches on or off if the supplied power is to low, a heat exchanger for peak load, and a thermal energy storage tank.

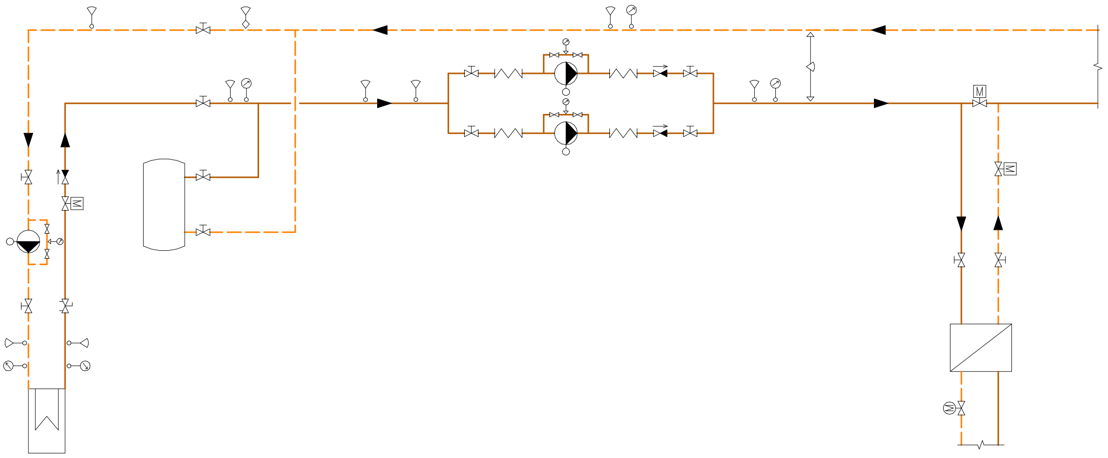


Figure 2.17: P&ID for the test system

This system is a simplification compared to large systems. However, this system is representative of a real system if there is only one heat pump. The usage of a heat exchanger as peak power might also be disputable. Nevertheless, this is chosen to verify and check the program, and its response to certain conditions, like if the supply temperature is lower than what it is supposed to be.

2.3 Method

The method used in this thesis is based on quantitative research. This approach means that the researcher gets comparable information about the object of interest in a systematic way, to express the information in the form of numbers or graphs. The results from the data can then be analyzed in order to find patterns and outcomes [18].

In this thesis, the chosen programming language is C# (pronounced "c sharp"). C# is a strongly typed programming language, which means that all the data must have a type. By forcing the programmer to choose a type for each data, the programmer can take active actions to improve speed since the types take up different sizes [19]. Besides, it also reduces the chance of unexpected results. For example, it is not possible to add a string and integer without first converting the string to an integer. Table 2.4 shows the data types used in this specific project.

Type	**Size (bytes)**	**Range**
byte	1	0 to 255
bool	1	true/false
int	4	-2,147,483,648 to 2,147,483,647
double	8	-1.79769313486232E308 to 1.79769313486232E308
string		Characters

Table 2.4: Data types, its size and its range used in this project [19]

Generally, there are several ways to create an add-in for Revit like, for example, Dynamo. C# is chosen because it is one of the most powerful programming languages for creating add-ins for Revit in terms of restrictions. There are, however, drawbacks by using this language, including the need for higher-level knowledge of programming compared to other usable resources like "Dynamo". Additionally, AutoDesk, the developer of Revit, has created a tutorial on how to create a Revit add-in for C#, which is easy to understand and very useful for beginners.

The development environment chosen for writing the code is Microsoft Visual Studio, which is also the recommended integrated development environment by Ky [19].

2.3.1 Implementation of the Add-in

In order to create a script that is useable in Revit, the first step is to generate a "Class library project". This makes it possible for the add-in and the Revit API to pass information to each other. The next step is to add references to RevitAPI.dll and RevitAPIUI.dll. When these steps are completed, the script is ready for programming [5].

Once the code is finished, the next step is to incorporate the script in Revit. To incorporate the script in Revit, it is necessary to create an "Add-in manifest". Among other things the manifest contains information about the name and location of the script and a unique ID for the script.

Visual studio has provided a program to create an ID, called "GUIDGEN.EXE". It is essential to create a unique ID for every script made for Revit to run them.

Once these steps are completed, the script will load into Revit, and the script is ready to be executed by the user.

2.3.2 Development

As stated in chapter 1, the script has to fulfill a handful of criteria in order to be useful. In this section, the different criteria and the general working principle of the code will be explained.

A sensible order to develop the add-in is by dividing the tasks into three different processes. The processes listed below will be further explained throughout this chapter.

- Pre-processing
- Processing
- Post-processing

In order to calculate parameters for the system in figure 2.17, some further information is necessary in order to get results. In this case, the input parameters provided by the user needs to be:

- Rated power output for the heat pump
- Rated COP for the heat pump
- Outdoor temperature
- Power demand
- Desired supply temperature at WDT
- Desired return temperature at WDT
- Hour NO.
- Source

To create a useful code, it is important to have at least some knowledge about Revit and its design. Revit is a parametric database, which means that every element in the database is related to each other. Whenever an object is added to the project, it is also added to the database [20]. In Revit, there are several types of elements. However, these elements can be divided into three main types. As shown in figure 2.18, these three types are: "Model elements", "Datum elements" and "View specific elements". The three main element types have different properties. While the model elements represent the models' actual 3D geometry and the datum elements represent the models' reference planes and points, the View specific elements are elements that only are available in the specific view.

Usually, a P&ID is created in a view called “Drafting view”, which means that the elements in the view are not a part of the building model [21]. This means that every element relevant for the add-in is in the category “View-specific”. Because nothing in the actual 3D model is linked with the P&ID, the way of identifying the elements is challenging. On the other hand, it means that if the add-in should behave inexplicably or delete something, it will not alter or possibly ruin the actual 3D-model.

Since all of the elements in the view are “View-specific”, the add-in needs a way to identify components in the P&ID. Furthermore, every element is a kind of element called “Detail line”, which means that there is not an easy way to get proper information about the elements without some processing. The first task is to identify each element and know what it is supposed to do. This is done by knowing the element name and what kind of component this name is representing. In the example file given, the element names and its function are presented in table 2.5.

Element name	**Representing**
EH_VARME RETUR	Return pipe
EH_VARME TUR	Supply pipe
MCX3111NOR_PUM01_001	Pump
MCX3110NOR_VAL04_010	Magnetic solenoid
MCX3130NOR_HEA09_001	Thermal energy storage tank
Skillelinje	Divide
MCX3130NOR_HEA12_001	Heat exchanger
EH_VV	Hot water pipe
EH_VVC	Hot water circulation pipe
MCX3110NOR_VAL03_001	Motorised valve
MCX3010NOR_AIR01_002	Dry cooler

Table 2.5: Element names and what component they are representing

In a Revit project, there are different sets of parameters called “Project” parameters, “Shared parameters” and “Global parameters”. As the name implies, the project parameters are project-specific, while the shared parameters can be used in multiple families and projects. To ensure that the add-in does not alter or change any other parameters, shared parameters are loaded into the project. The parameters can be used to store information about the element. For example, it is possible to write information about temperature or diameter in the parameter.

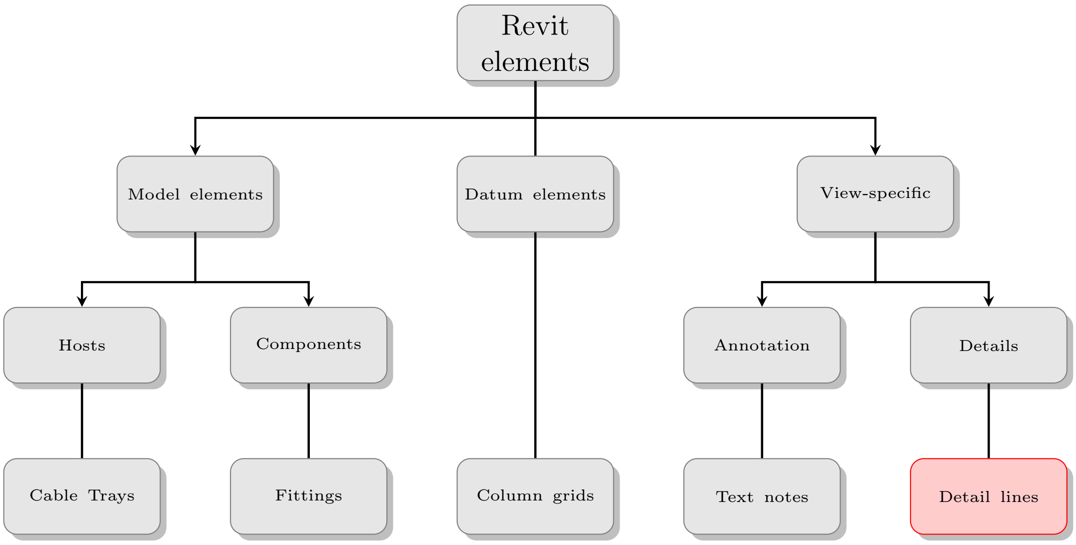


Figure 2.18: Element types in Revit [20]

Pre-processing

In the first process, Pre-processing loads the Add-in to Revit and handles the input to the script.

The very first part of the script is invoking Revit to load the add-in. Further, the script defines that the menu bar should contain the add-in. This part of the script is not doing any calculations. However, if the button is pressed, it invokes another script that includes information on what task that should be performed. In this thesis, the chosen name for the Tab is "EneCalc", the same for the button, while the ribbon panel text is "Master's project". The chosen logo for the button is the OsloMet logo.

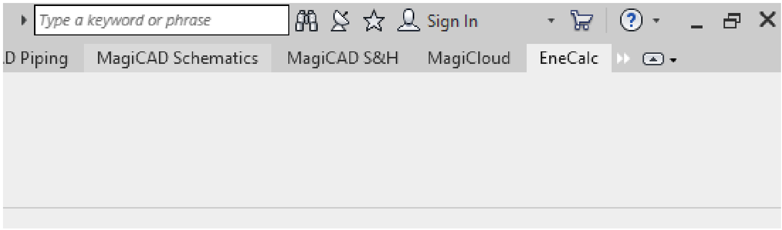


Figure 2.19: Tab with the name implemented

As previously mentioned, parameters can be used to store specific information about an element. In this case, each element needs to have parameters to store the information about the element from the calculations. In the add-in, the way of adding parameters is first to create shared parameters, and then add them to the project as project parameters, which is the first step of the actual pre-processing. In Revit, the parameters act as

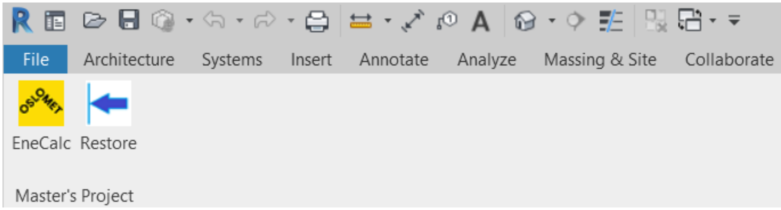


Figure 2.20: Execute button and restore button

physical properties, which means that it is important to add information about what kind of unit the properties have. This is a one time job, so the script needs to check if the shared parameters with the desired name exist and then create them if they do not exist, as presented in figure 2.21.

Parameter name	**Parameter Type**	**Unit**
ENE_Temperature	PipingTemperature	°C
ENE_Flow_Rate	PipingFlow	L/s
ENE_COP	Number	-
ENE_Identifier	Text	-
ENE_Power	HVACPower	kW
ENE_Diameter	BarDiameter	mm
ENE_Insulation	PipeInsulationThickness	mm

Table 2.6: Added parameters to the project

The calculations require some user-specified values. In the add-in, a user form for the inputs has been created. The form with its inputs is presented in figure 2.22, and the required inputs are specified below.

- WDT
- Supply temperature at WDT
- Return temperature at WDT
- Rated COP
- Rated power
- Source (Borehole/air)

The outdoor temperature compensation characteristic needs a setpoint temperature for the building, the supply and return temperature, and the winter design temperature,

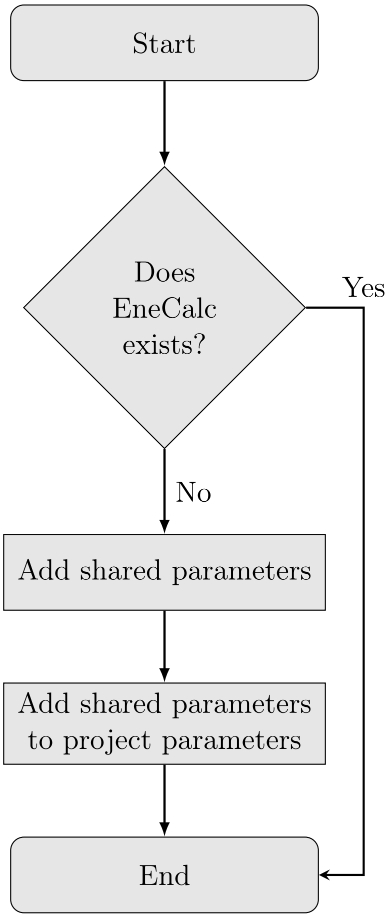


Figure 2.21: Flowchart for the process of adding parameters to the project

as shown in equation 2.50 and 2.51. All the values are inserted by the user, except the indoor/ambient temperature, which is simplified to be 22℃ in the entire project.

The next performed task is reading user-provided input. In this case, the pre-calculated power demand should be provided from IDA-ICE or some other energy simulation tool. The values should then be exported to excel, which will be read by the script. To speed up the process, the script will read each value and pass it over to a list. The list has a parameter for each read value from the file. This handling ensures that there is only a need for obtaining the values once, and it will clear up memory for the rest of the process. To read the data from the sheet, the script needs to know exactly what kind of information each column contains. There are several ways of obtaining such information, but in this script, there are two ways that have been considered. The first way is to specify what kind of information the column contains. The advantage of using this method is that as long as the setup is constructed in a good way, the chance of error from the program is low. However, this method requires exact inputs from the user. If

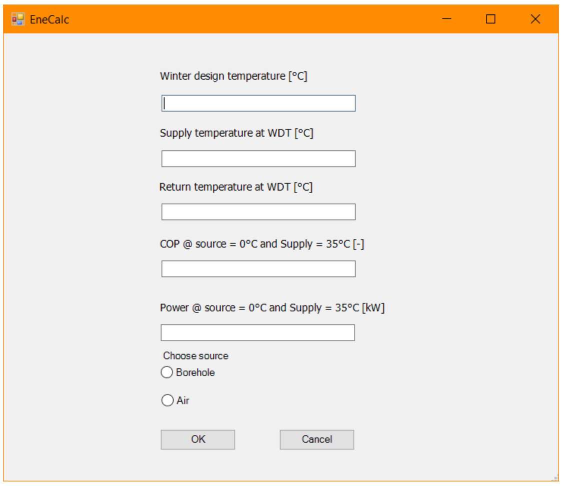


Figure 2.22: The user form with input values

the user switches values between columns, the rest of the script will read this column as something else than what it actually is. The other method is to program the script to read the header for each column. This will create a more dynamically sheet for the user, where there is no need for limiting data to certain columns. On the other hand, this method is more sensitive to what the headers contain. For example, if the script is programmed to search for a keyword called "Power" and the user has written "Powert", the script will ignore the whole column. Both of these methods have great weaknesses, and there should be other ways to handle this problem. However, since this task is limited, the way this script will work is by using the first method. Which is forcing each column to contain specific information. The column order is given by table 2.7.

Column nr	**1**	**2**	**3**
Input	Hour NO.	Outdoor temperature	Power
Unit	h	℃	kW

Table 2.7: Added parameters to the project

In C# there are no predefined functions for reading values from a .xlsx or .xlsm file. However, listed below is the logic for how the script reads and handles the data from the file.

```
Define file path
Define sheet number
Read file path
Open file
Open sheet
Check how many rows that contains data
```

```
7  For i = 1 to 3
8      For each row
9          Check data type
10         If type is double
11             Store to list
12         Else
13             Skip to next row
14         End if
15     End for
16 End for
17 Close file
18 Delete any connection to the file and file path
```

In the code above, the first lines handle the file placement and prepare the script to read input. Since an excel file may have several sheets, the code needs input on which sheet number to look into. Line 6 checks how many rows the file contains. This is in order not to limit the script to a certain time constraint. In other words, this means that there is not a restriction on how many values are needed in order to make a valid calculation. In order to create reliable calculations it is recommended that this number is at least 8760 which is the number of hours in a year. However, the script is designed to operate with values varying from one to a very high but finite amount of rows. It is worth noting that a very high amount of data will impair the performance. Furthermore, it has not been tested.

Any excess information is unnecessary and should not be included. The time parameter could be excepted, but it is an excellent way to keep track of the time of the year, especially if the user desires to perform a calculation for just some months.

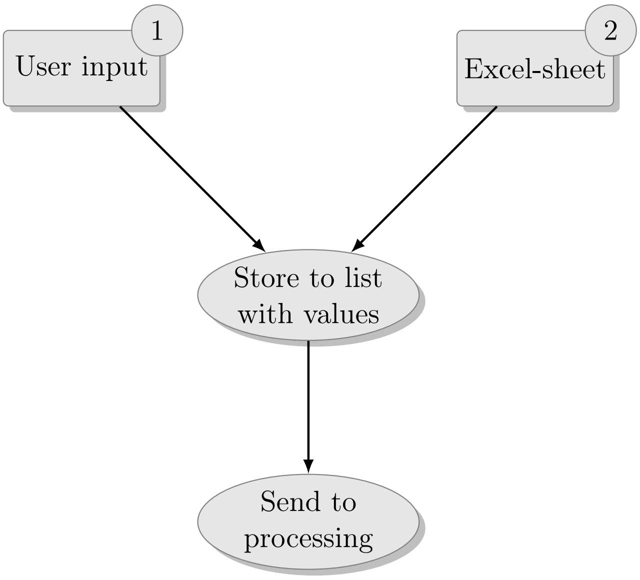


Figure 2.23: Flowchart for the pre-processing part

Processing

After the given parameters have been read and stored, the next step is to start the calculation.

The calculations are executed for each hour from the excel-file. The logic in this code is that the system has a specifically given power demand for each timestep, which is given from the excel input. As seen in the test-setup, some different components need to be treated differently, and in the right order to calculate accurate results. The performed calculations for each circuit are presented in table 2.8, and marked with an x if it is performed on the circuit.

	Unit	**Supply**	**Return**	**Peak**
T	℃	x	x	x
ρ	kg/m^3	x	x	x
$\dot{m}$	kg/s	x	x	x
C_p	kJ/kgK	x	x	x
μ	Pa s	x	x	x
$\dot{V}$	l/s	x	x	x
P_{hp}	kW	x		
COP	-	x		
P_{peak}	kW			x
ΔP	Pa/m	x	x	x
D_i	mm	x	x	x
Insulation thickness	mm	x	x	x
Conductivity	W/mK	x	x	x
Heat transfer	W/m	x	x	x

Table 2.8: Calculations performed on circuits

First, the desired supply and return temperature are calculated according to equation 2.50 and 2.51. Simultaneously as the temperatures are calculated, the isobaric heat capacity is calculated for each temperature. By knowing these values, the required mass flow rate for the desired power-output can be calculated. However, if the power demand is deficient, the mass flow rate is set to 0 as

$$\dot{m}(t) = \begin{cases} 0 & P_{demand}(t) < 10^{-2} \\ \dfrac{P_{demand}(t)}{C_p(T,t,p)\Delta T(t)} & otherwise \end{cases} \tag{2.55}$$

If boreholes are selected, the next part calculates the borehole temperature, given by equation 2.34. Since this equation is dependent on the month number, the number of hours in a month is approximated to be 730. By dividing the number of hours with 730 and then using the ceiling formula, the result will be the month number as

$$mo = \left\lceil \frac{t}{730} \right\rceil \tag{2.56}$$

When the source temperature and supply temperature is known, the maximum power output from the heat pump can be calculated. The calculations are done according to the equations in section 2.2.1 and in accordance with the selected source.

After obtaining the maximum power output from the heat pump, it is necessary to check whether the output meets the power demand for the building or not. According to the calculations from SN-NSPEK 3031:2020, it is under certain conditions possible to achieve a COP lower than one. If the COP is one, or lower, the heat pump is not operating efficiently, and the operation is terminated. The power delivered by the heat pump is expressed as

$$P_{HP,adj}(t) = \begin{cases} P_{demand}(t) & P_{demand}(t) \leq P_{HP}(t) \\ 0 & COP(t) \leq 1 \end{cases} \tag{2.57}$$

After adjusting the power, the program adjusts the COP. Even though it is stated in SN-NSPEK 3031:2020 that the COP stays stationary below a specific temperature [15], given the definition for COP from equation 2.33, the COP does not have any relevance if the heat pump is not in operation, which is why the COP is set to zero if the power output is zero, as

$$COP_{HP,adj}(t) = \begin{cases} COP_{HP}(t) & 0 < P_{HP,adj}(t) \\ 0 & otherwise \end{cases} \tag{2.58}$$

The program then calculates the actual supply temperature from the heat pump. Since the standard claims that the highest possible achievable supply temperature from the heat pump is 55°C, the program adjusts the temperature to 55°C, even though the power output is sufficient for raising the power higher [15].

$$T_{su,adj}(t) = \begin{cases} \dfrac{P_{hp,adj}(t)}{\dot{m}(t)C_p(T,t,p)} + T_r(t) & T_{su}(t) \leq 55 \\ 55 & otherwise \end{cases} \tag{2.59}$$

If the supply temperature is set to 55°C, there is a need to adjust the power output from the heat pump accordingly as

$$P_{hp,adj,2}(t) = \begin{cases} \dot{m}C_p(T,t,p)(55 - T_r(t)) & 55 < T_{su} \\ P_{hp,output}(t) & otherwise \end{cases} \tag{2.60}$$

When the adjusted supply temperature is calculated, the peak-load power can be calculated as

$$P_{peak}(t) = \dot{m}(t)C_p(T,t,p)(T_{su}(t) - T_{su,adj}(t)) \tag{2.61}$$

The advantage of this equation is that the peak power automatically becomes zero if the heat pump power output is sufficient. When the peak power is known, the peak temperature is calculated as

$$T_{su,peak}(t) = \begin{cases} T_{su}(t) & T_{su,adjusted}(t) < T_{su}(t) \\ T_{amb} & otherwise \end{cases} \tag{2.62}$$

It is, of course, not the case that the temperature will drop down to ambient temperature instantly. However, since the program does not calculate response time, the fluid temperature is set to ambient to show that nothing is happening in the pipe. Furthermore, should the fluid stay stationary in the pipe for a long time, the temperature will drop to ambient temperature.

By using the temperature for the peak supply, it is possible to calculate the mass flowrate for the peak supply by,

$$\dot{m}_{peak}(t) = \begin{cases} \dot{m}(t) & T_{su,adjusted}(t) < T_{su}(t) \\ 0 & otherwise \end{cases} \tag{2.63}$$

This equation is dependent to avoid division by zero. The working principle of how the add-in calculates temperatures, power, and COP is presented in figure 2.24. Note that this is performed for each hour.

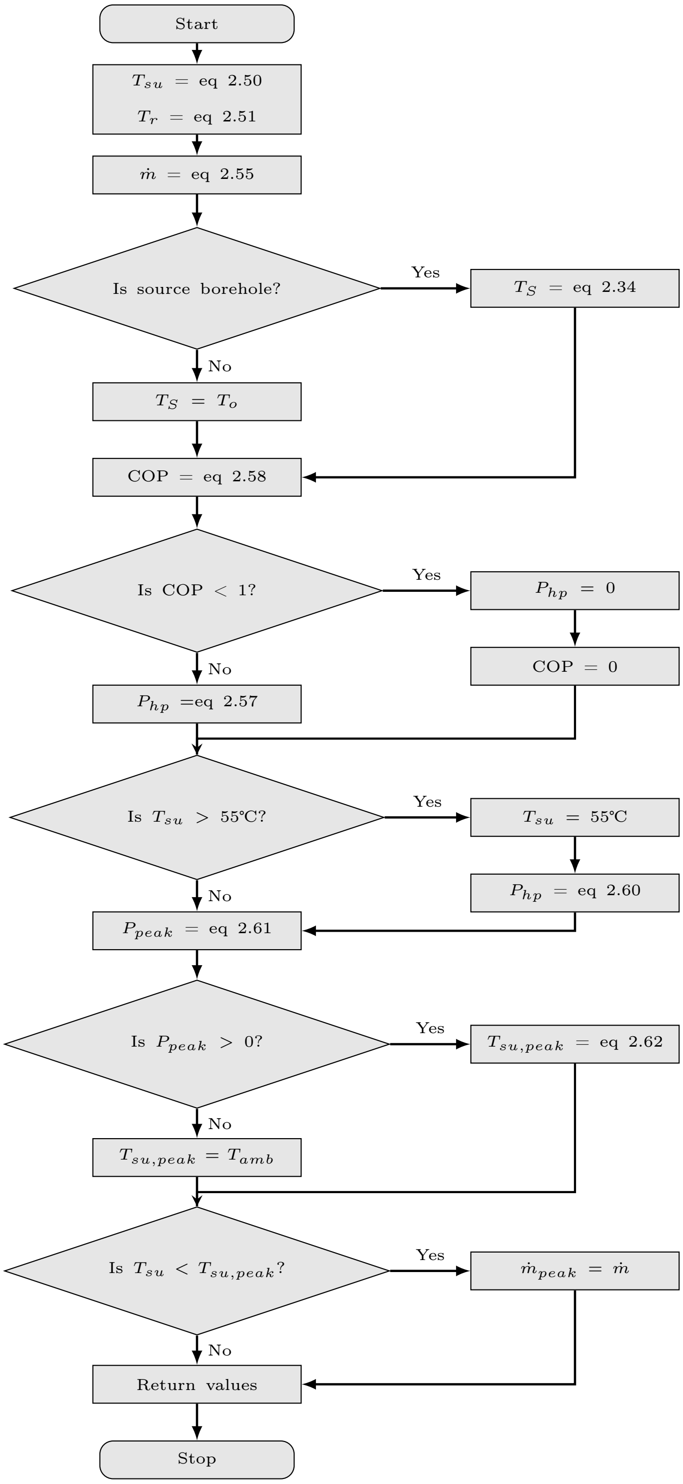


Figure 2.24: Flowchart for the algorithm for temperatures, power and COP

When all the temperatures and mass flowrates in the system are known, it is possible to calculate both the dynamic viscosity and the density of the water for each pipe. All of these calculations are performed with the List<T> class in C#, which ensures both speed and reliability. By using the System.Linq, it also opens up for using "lambda expression" to find specific values without having to "loop" through the whole list.

Usually, when the diameter of the pipe is determined, it is common to calculate the pressure drop per meter for the volumetric flow at winter design temperature. However, as seen in equation 2.52. A high pressure drop per meter leads to higher power demand for the pumps. Therefore, the program looks up the time of the year where the volumetric flow is highest and sizes the pipes according to the volumetric flow. The determination of the highest volumentric flow is done by using a lambda expression as presented below.

```
double maxFlowSupply = hourSupplies.Max(s => s.volumetricFlowRateSupply);
double maxFlowReturn = hourSupplies.Max(s => s.volumetricFlowRateReturn);
```

If one should require to determine the size in another way, for example by the highest volumetric flow rate at winter design temperature, a lambda expression can be used as

```
double WDT = double.Parse(UserForm.txtWDT); \\Winter design temperature
List<HourSupply> x = hourSupplies.FindAll(s => s.outsideTemperature == WDT);
double maxFlowSupply = x.Max(s => s.volumetricFlowRateSupply);
```

This example also shows how the user input are treated. The winter design temperature is inserted into a text box, which is treating the input as a string. The script then needs to convert the user input into a double for use later.

As seen in the example above, it is necessary to have calculated the parameters for every hour before it is possible to determine the size of the pipe. The first calculation is done in order to determine the size of the pipes for the supply, return, and peak circuit. The calculation is performed according to the equations in section 2.1.7. The pressure loss algorithm is presented in figure 2.25, and the sizing algorithm is presented in figure 2.26. A list of diameters for steel pipes has been implemented in the add-in. By doing so, it is possible to apply an indexer to the list. By applying a counter to the algorithm, one can loop through the list until the criteria are satisfied. The algorithm also has a stop function if the largest diameter should not meet the criteria given. The roughness for steel pipes used in the algorithm is 0.002mm [8].

After the size for each of the circuits are determined, the next step is to calculate the pressure loss for each hour. As previously mentioned, this calculation has to be done separately, and again, the program is looping through each hour.

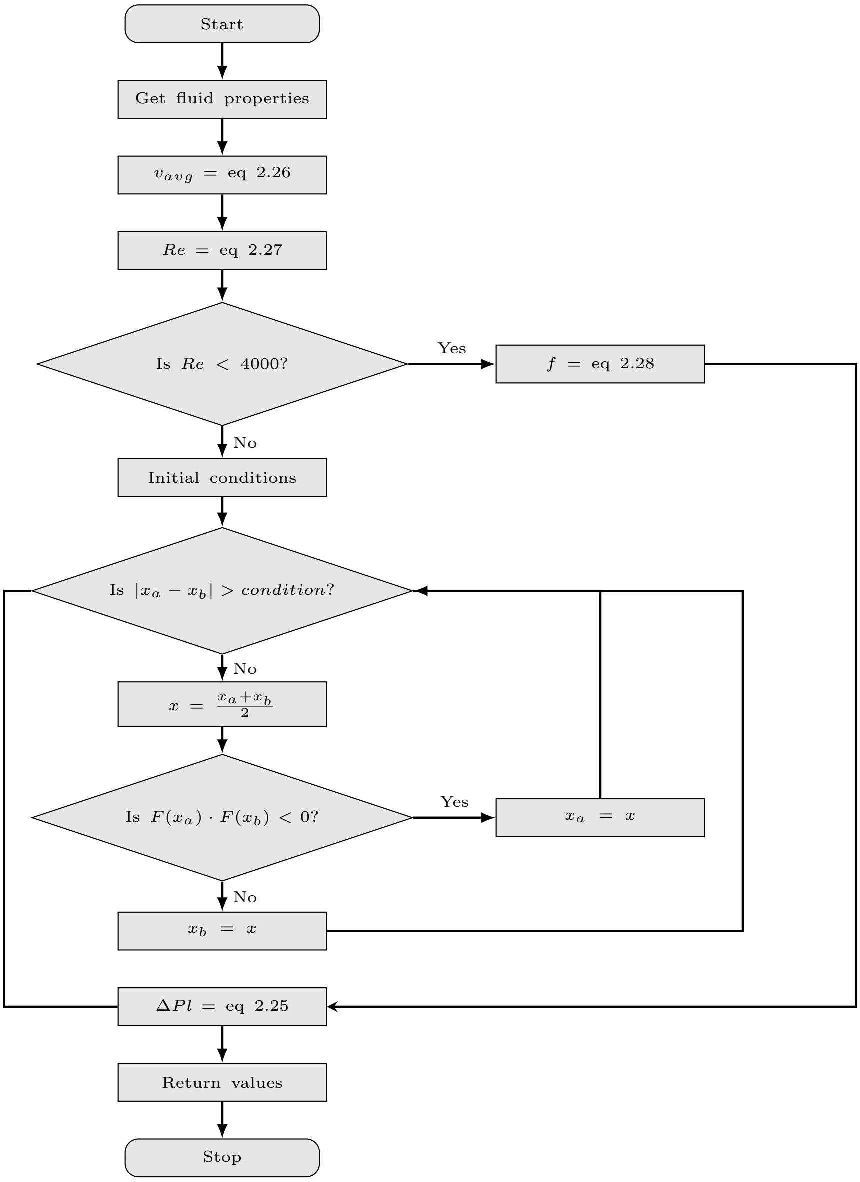


Figure 2.25: Flowchart for the algorithm for pressure loss calculation

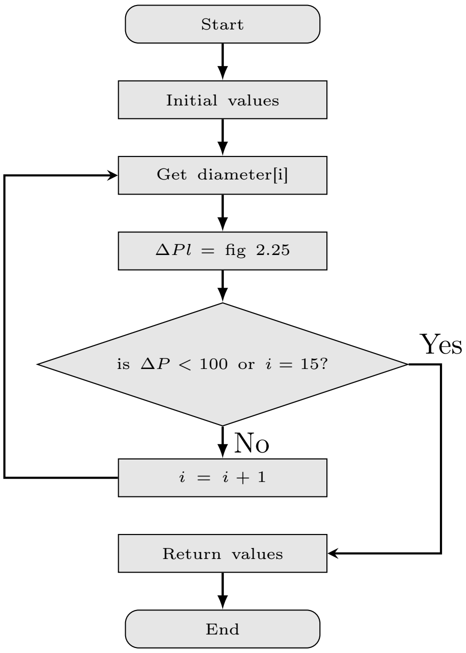


Figure 2.26: Flowchart for the algorithm for the determination of the pipe diameter

When the pipe diameter and the temperatures are known, it is possible to calculate the heat transfer per meter for the pipe, by using equation 2.23. In this part, the program uses the calculated diameter for the pipes and has a set value for the maximum U', according to equation 2.24. Again, a list is included with different thicknesses for insulation in the add-in. The script will then loop through until the requirements are met or if the highest value for insulation thickness has been reached.

Since the U' value are independent of the temperature, this is done for the first hour, but the heat transfer rate with the calculated thickness of the insulation is calculated for each hour. The thermal conductivity for steel used in this program is 15.1 W/mK(AISI 302) [22], and the thermal conductivity for the insulation is 0.036W/mK at 50℃ [23].

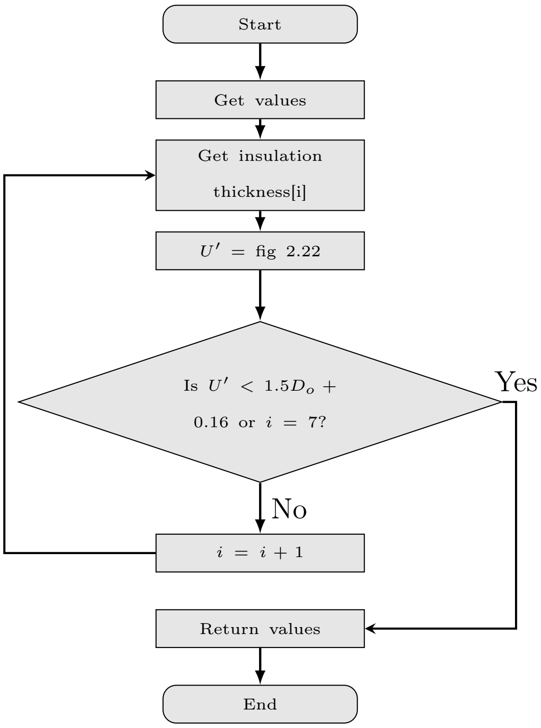


Figure 2.27: Flowchart for the algorithm for determination of insulation thickness

After all the values have been calculated the results are written to a .CSV-file for the user to read. CSV stands for "'Comma separated values", which means that the values usually are separated with commas. However, since comma is the standard decimal separator in Norway, the separator chosen for this particular file is a semicolon, which ensures that there are not any problems with values with decimals.

Post-Processing

The post-processing in this program handles the presentation of the results. The idea in this part is to display the results visually easily. By doing so, it becomes easy for the user to understand and see the behavior of the system and the calculated results. In Revit, there is a function that allows for "overriding" displayed graphics. This function alters the display of the elements without physically modifying the elements. In this project, the temperature decides the color of the elements. In order to make a temperature-scale, it is created a function on a basis from equation 2.67 and 2.68. In this case, ΔT is the difference between the lowest calculated value and the highest throughout the year, except where the water flow is 0. When there is no flow in the pipes, the color is set to gray.

$$\Delta T = T_{su,max} - T_{r,min} \tag{2.64}$$

$$\Delta T(t) = \frac{T_x(t) - T_{r,min}}{\Delta T} \tag{2.65}$$

$$Red = \begin{cases} 0 & \Delta T(t) < \dfrac{\Delta T}{2} \\ 1020\left(\Delta T(t)) - \dfrac{\Delta T}{2}\right) & \dfrac{\Delta T}{2} \leq \Delta T(t) < \dfrac{3\Delta T}{4} \\ 255 & otherwise \end{cases} \tag{2.66}$$

$$Green = \begin{cases} 1020\Delta T(t) & \Delta T(t) < \dfrac{\Delta T}{4} \\ -1020\left(\Delta T(t) - \dfrac{3\Delta T}{4}\right) + 255 & \dfrac{4\Delta T}{3} < \Delta T \\ 255 & otherwise \end{cases} \tag{2.67}$$

$$Blue = \begin{cases} 255 & \Delta T(t) < \dfrac{\Delta T}{4} \\ -1020\left(\Delta T(t) - \dfrac{\Delta T}{2}\right) & \dfrac{\Delta T}{2} \leq \Delta T(t) < \dfrac{3\Delta T}{4} \\ 0 & otherwise \end{cases} \tag{2.68}$$

This method ensures that the scale is between the lowest and highest possible temperature in the system. When it comes to water flow, the way of displaying the flow is by the use of thickness. Again, the same logic is used here, where the largest flow is eight, and the lowest is 1. The reason why eight is chosen as the highest quantity is that values

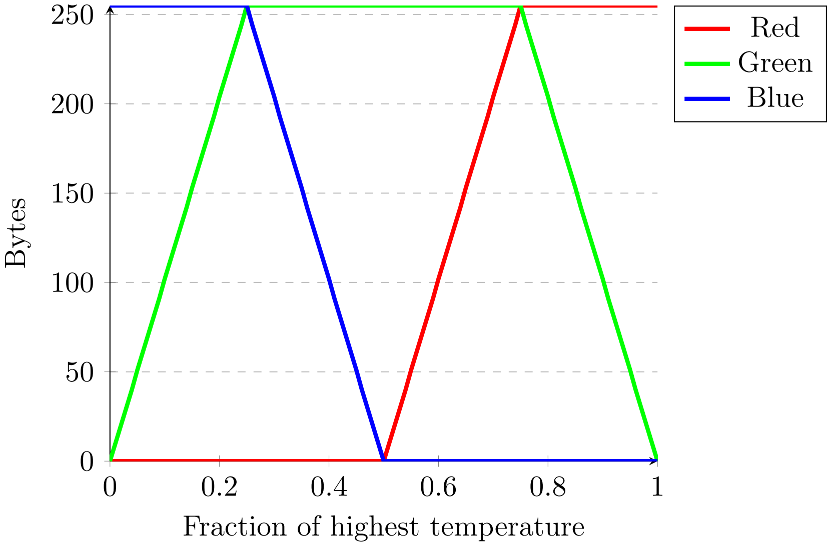


Figure 2.28: Values for red, green and blue relative to the temperature in the system

above eight make the lines disproportionally thick, which might lead to lines crossing or blocking other lines and impairs the vision of the lines for the user.

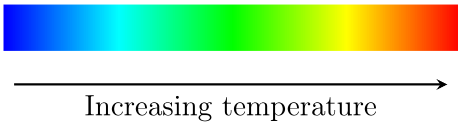


Figure 2.29: Color scale for temperatures

$$Thickness = 8\left(\frac{\dot{V}(t)}{\Delta\dot{V}}\right) \tag{2.69}$$

When the calculations have been executed, the user is prompted to fill in a specific hour number, which is used to display the results in Revit. The values for temperature, diameter, volumetric flowrate, and insulation thickness are added to each correspondent element in the P&ID. Note that Revit uses Imperial units as default. In order to display the units as desired, a method to convert the unit to Revit's internal units have to be applied.

```
UnitUtils.ConvertToInternalUnits(tReturn, DisplayUnitType.DUT_CELSIUS);
```

As shown above, the first input is the calculated value; in this case, the return temperature. The next input is what unit that is desirable. Since the temperature in the add-in is calculated in ℃, the "DisplayUnitType" has to be Celsius. In cases where the desired unit type cannot be obtained, the parameter has to be set as "number". In this case, Revit will not convert the input to internal units.

The user also has to fill in specifications on the pipes. Since all the elements in the view are "detail lines" there are ways to find out if the pipes or components are connected to each other. The program is able to distinguish different types of elements by their name, but it cannot differ between a supply pipe with a peak load or a supply pipe without peak load. To distinguish further, the user needs to fill in information about the element in the parameter called "ENE_Identifier".

Code	**Function**
x	Pipes whithout flow. Any of theese wil be gray and the temperature is set to ambient temperature.
peakSupply	The supply pipe to the peak load
peakReturn	The return pipe from the peak load
peakValve	The pipe where the solenoid is
p1	Pipes connected to pump 1
p2	Pipes connected to pump 2

Table 2.9: Codes for the different elements in the project

This task can be utilized prior to or after the calculations. It will, however, not display the right results if the identifier is not properly placed.

Restoring

Usually, there are specific requirements for the color and patterns of the lines for a P&ID. Since the add-in alters the colors and the thickness of the lines, it causes inconsistencies if the P&ID is used for print. For this reason, a "Restore" function is added. This function restores all the visual settings to default. The way this works is that it loops through each element in the view, and restores it. The restore function only alters the "SetElementOverrides" method in Revit, so there is nothing else that is affected by this function. However, it does not delete the parameters or the values written to the parameters, such as temperature or volumetric flowrate.

Chapter 3

Results

3.1 Verification

3.1.1 Isobaric heat capacity for water

By doing calculations for the isobaric heat capacity for water, it was found that the most significant offset was about 0.5% from the traditional static value and the calculated value. This is a deficient error, and it might be questionable if there is a need for such precise calculations. As shown in figure 3.1, there is a little difference in the variation. If 4.2 kJ/kgK is used as the heat capacity for water, there will rarely be any significant error since the calculations will overestimate by a meager percentage.

However, the added calculation adds higher precision to the calculations by including pressure. This means that the program will be able to calculate precisely within any applicable range in a conventional plumbing system. As shown in figure 3.2 the difference is minimal. In fact, the largest difference is 0.11%, and this is within the temperature range that is not often used in plumbing applications.

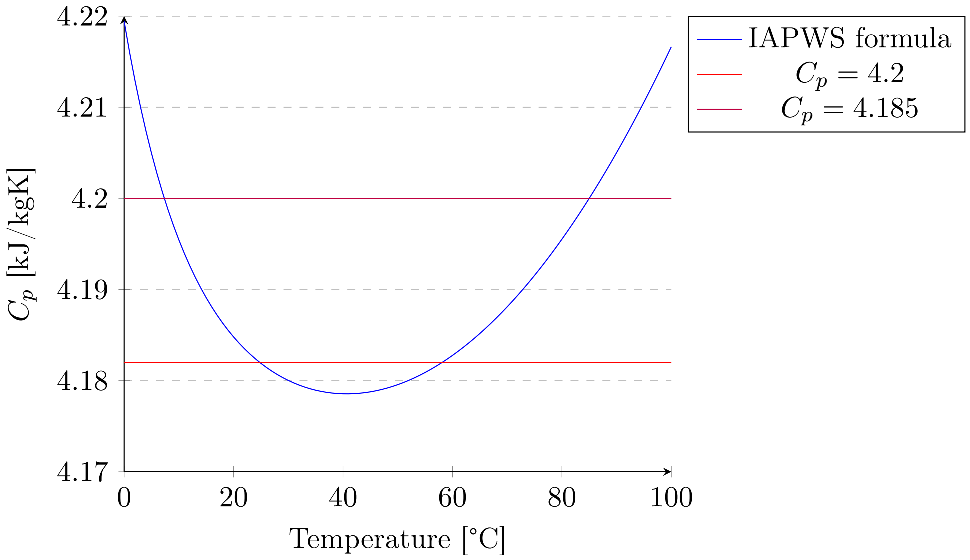


Figure 3.1: Comparison of values for H_2O with pressure at 1 atm and temperature ranging from 0°C to 100°C

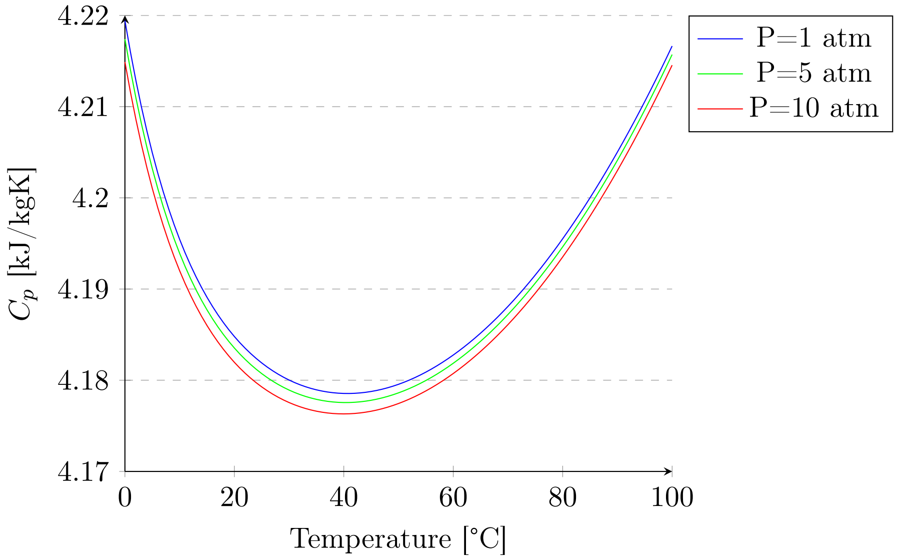


Figure 3.2: Comparison of heat capacity for H_2O with pressure at 1 atm, 5 atm and 10 atm with temperature ranging from 0°C to 100°C

3.1.2 Density for water

The results from the calculations of water density are presented in figure. As displayed, there is almost no difference between 1 and 10 atm. This means that using Kells equation gives almost no error compared to the IAPWS equation. Nevertheless, density should be calculated, since the difference between the density is significant, especially at higher temperatures.

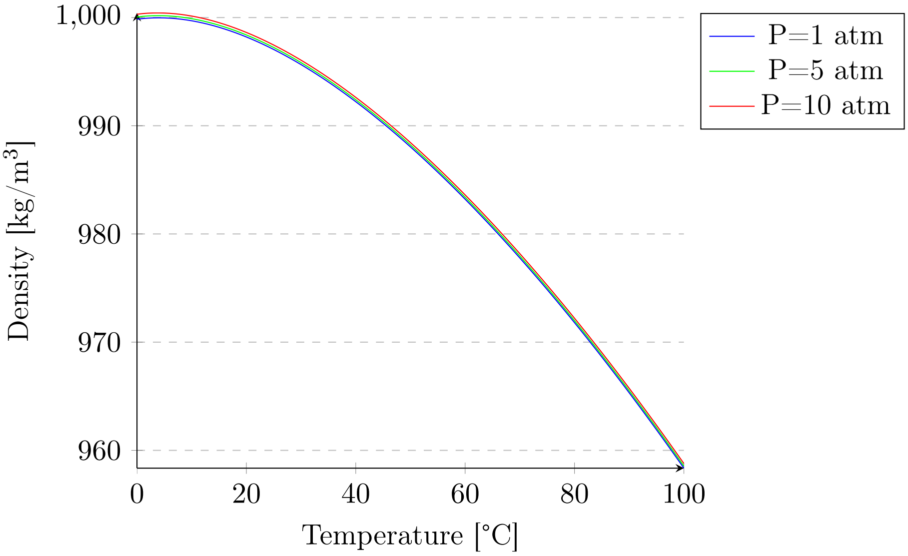
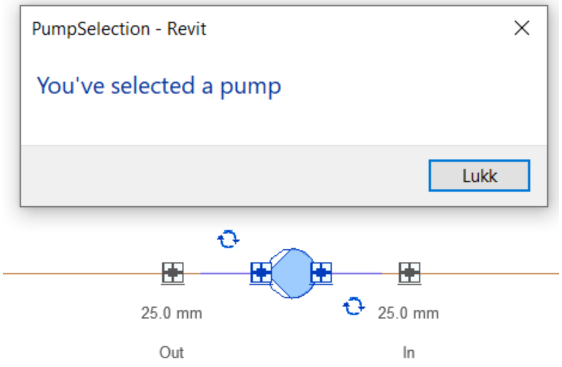


Figure 3.3: Comparison of density for H_2O with pressure at 1 atm, 5 atm and 10 atm with temperature ranging from 0°C to 100°C

3.2 Results from the program

3.2.1 First test

At the beginning of the process, a simple test was constructed to check if the script would behave as expected. The test is relatively simple. An object representing a pump in Revit was examined, and it showed that the object had “Element ID: 2550488”. The designed test would then give feedback in the form of a message box if the user had selected this specific object, something else, or nothing was selected. The result of the test is presented in figure 3.4.



PumpSelection - Revit

You've selected a pump

Figure 3.4: Result from the first test

The test-script performed exactly as expected with the message shown in figure 3.4.

3.2.2 Calculations

To evaluate the results from the program, a real energy simulation calculation has been used as input. The results used for this purpose are the calculations from the new Life science building at the University of Oslo. The input file contains results previously calculated in IDA-ICE. These calculations represent a real case, which is indeed very good for evaluating the add-in.

As presented in figure 3.5, the power demand for the building fluctuates through the year. The highest power demand is approximately 3300kW, which is quite high for one heat pump. However, it is not too much for a building with a size of 66,700 m^2 [24].

By sorting the power demand from highest to lowest, it becomes clear that there is only a short amount of time when the power demand is above 1500kW. Zijdemans states that “as a rule of thumb, the heat pump should be able to cover 50-60% of the maximum power demand for the building” [16]. For the evaluation of the program, the power supply is set to 1650kW, even though the building initially has four heat pumps. According to SINTEF, the winter design temperature for Oslo is -19.8℃, which is used for input in the evaluation. The average outdoor air temperature for Oslo throughout

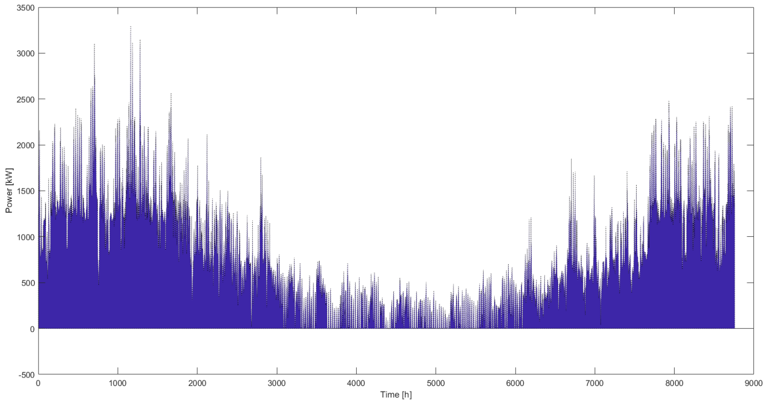


Figure 3.5: Power demand for the Life science building in one year

a year is 6.3°C [25]. This means that the borehole should use the specifications for 5°C when calculating the source temperature [15]. Besides, two different supply and return temperatures have been evaluated in order to investigate the outcome of such changes.

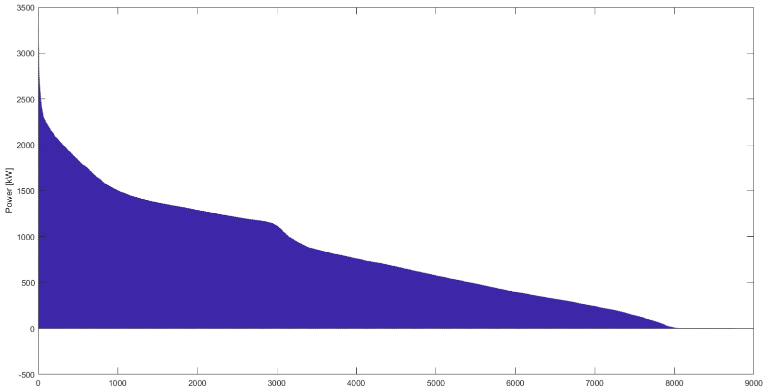


Figure 3.6: Power demand for the Life science building in one year sorted by demand

In the following sections, each case will be discussed. The inputs from table 3.1 were used as input to the program. The first case will have a more in-depth discussion than the following cases since some of the results are relatively similar. All results presented are results produced by the add-in.

	Unit	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Power	kW	1650	1650	1650	1650
COP	-	3	3	3	3
$\mathbf{T_{su}}$	℃	60	40	60	40
$\mathbf{T_{r}}$	℃	40	30	40	30
Supply source	-	Borehole	Borehole	Air	Air
WDT	℃	-19.8	-19.8	-19.8	-19.8

Table 3.1: The four different test cases and its properties

Case 1

In the first case, the supply temperature is set to 60°C, and the return temperature is set to 40°C. A supply temperature of 60°C is the highest pre-accepted supply temperature by "Regulations on technical requirements for construction works" (TEK17), which is why this temperature has been chosen[10].

With a supply temperature above 55°C at winter design temperature, it becomes clear that the heat pump never can deliver such high temperature. However, from the data, there are only 106 hours when the supply temperature is supposed to have a temperature above 55°C. The inner diameter for each circuit has been calculated to be 215.1mm. With this diameter, the maximum pressure drop per meter is 55.7Pa/m for supply, 57.2Pa/m for return, and 55.1Pa/m for the peak. The variations are due to the change of density and viscosity due to temperature. Since the maximum threshold value for pressure drop is 100Pa/m, these numbers confirm that the sizing algorithm works as supposed.

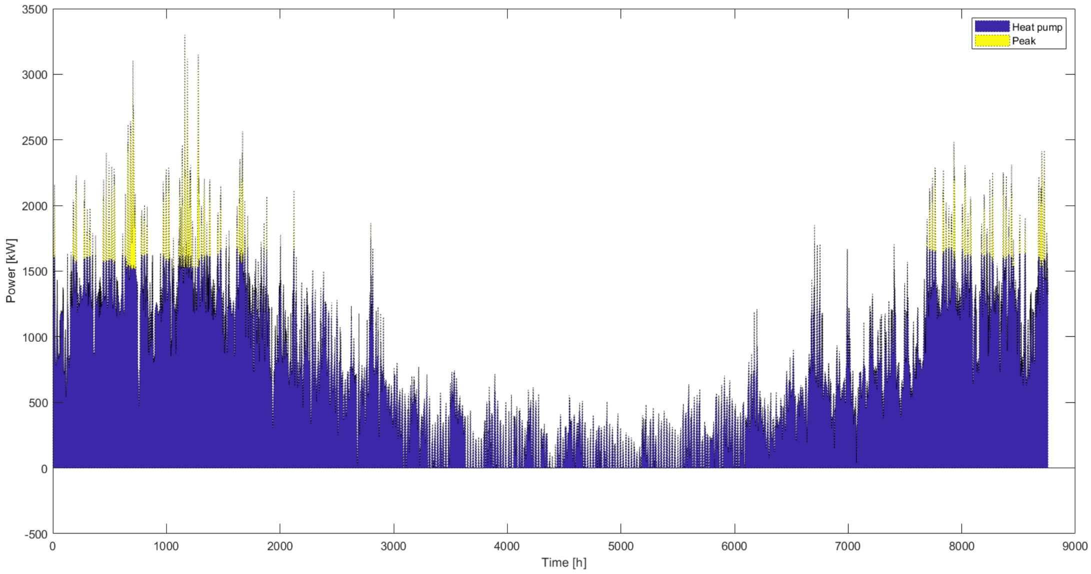


Figure 3.7: Heat pump and peak power for case 1

As seen in figure 2.8, it is clear that the borehole temperature never drops below -5°C, which is the point where the heat pump should be switched off. This means that the heat pump can raise the temperature of the fluid the whole year. However, as seen in figure 3.8, the lines are jagged. This is probably due to the fact that there are other factors than the outside temperature that dictates the power demand.

Figure 3.9 displays the COP for the heat pump during a year. The results clearly show that the COP increases during the summertime. From equation 2.58, the COP drops to zero when there is no power demand for the building.

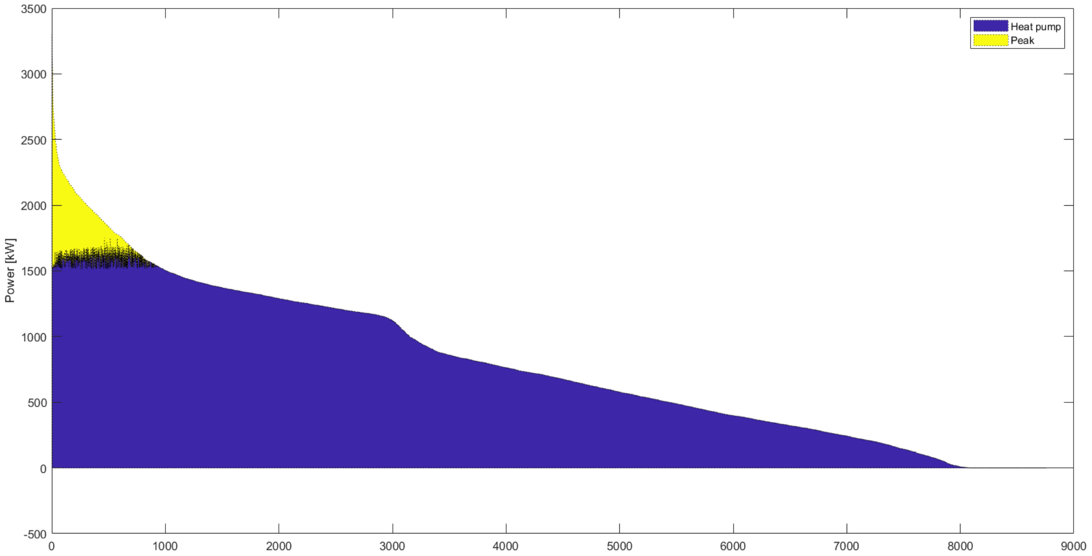


Figure 3.8: Heat pump and peak power for case 1 sorted by demand

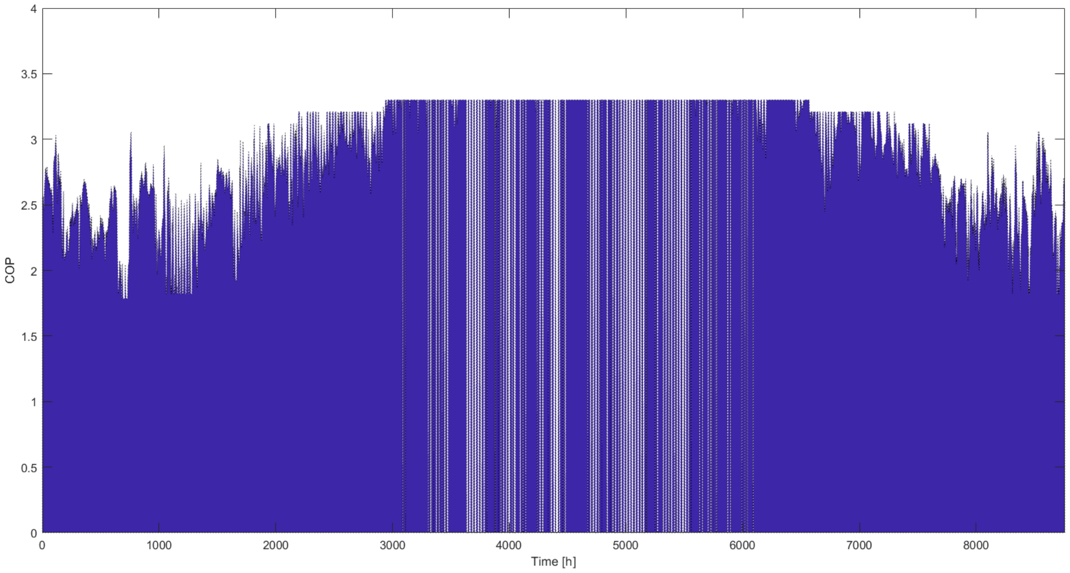


Figure 3.9: COP for case 1

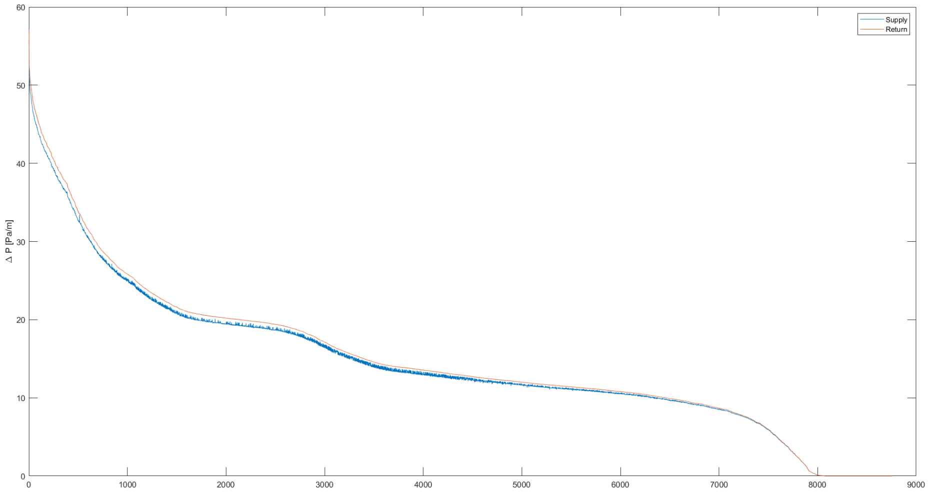


Figure 3.10: Pressure drop per meter for supply and return pipes for case 1 sorted by the highest peak values

Not surprisingly, the results for the volumetric flow rate looks similar to the figure for pressure drop. When analyzing the results, the volumetric flow rate for the return pipe is a bit lower than the volumetric flow rate for the supply pipe, which is due to the density difference.

As seen in figure 3.12, the heat loss for the supply pipe fluctuates at the start. This is due to the lower supply temperature before the heat exchanger because the heat pump is not able to raise the temperature of the water to the desired temperature. From equation 2.22, the heat loss is dependent on the fluid temperature, which means that a lower temperature in the supply pipe will result in a lower heat loss. At the end of the graph, both values drop to zero, which is when the power demand for the building is zero and the heating plant is not in operation.

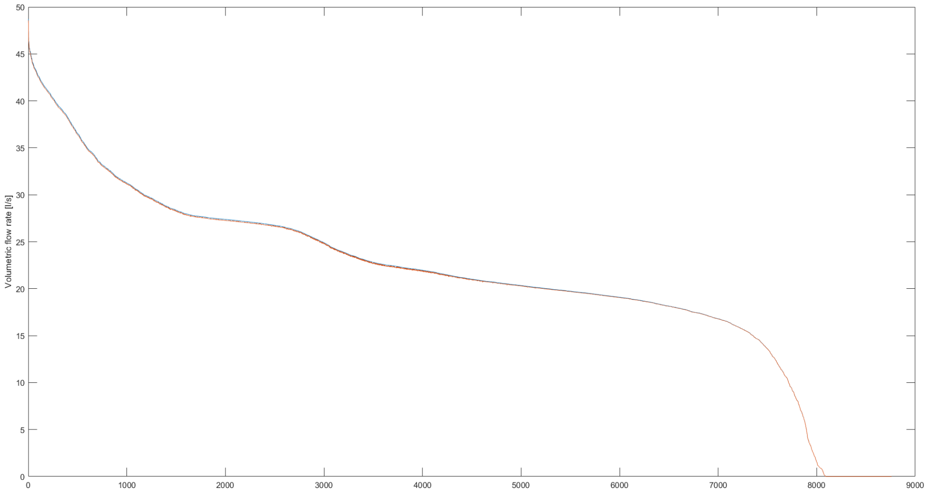


Figure 3.11: Volumetric flow rate for supply and return pipes for case 1 sorted by the highest peak values

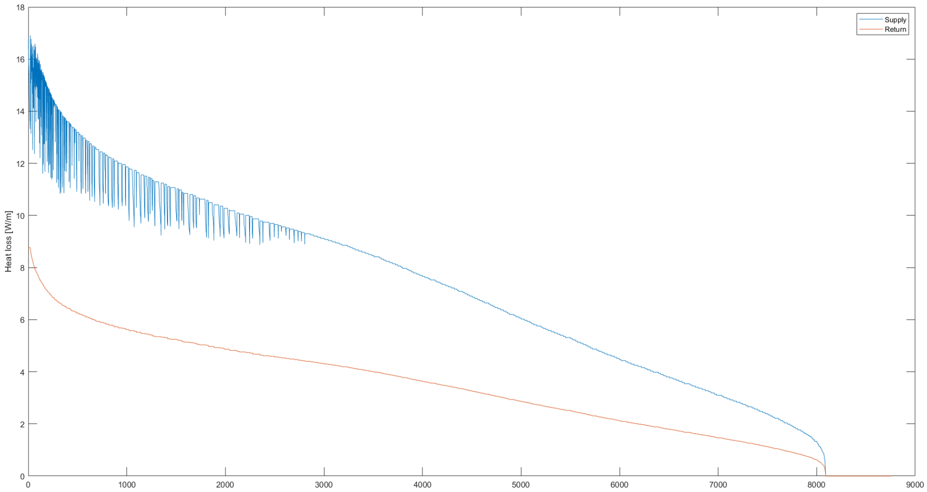


Figure 3.12: Heat loss for the supply and return pipe sorted after highest value for the return pipe

Case 2

In the second case, the supply temperature have been slightly lowered. The temperature difference is also changed from 20K to 10K.

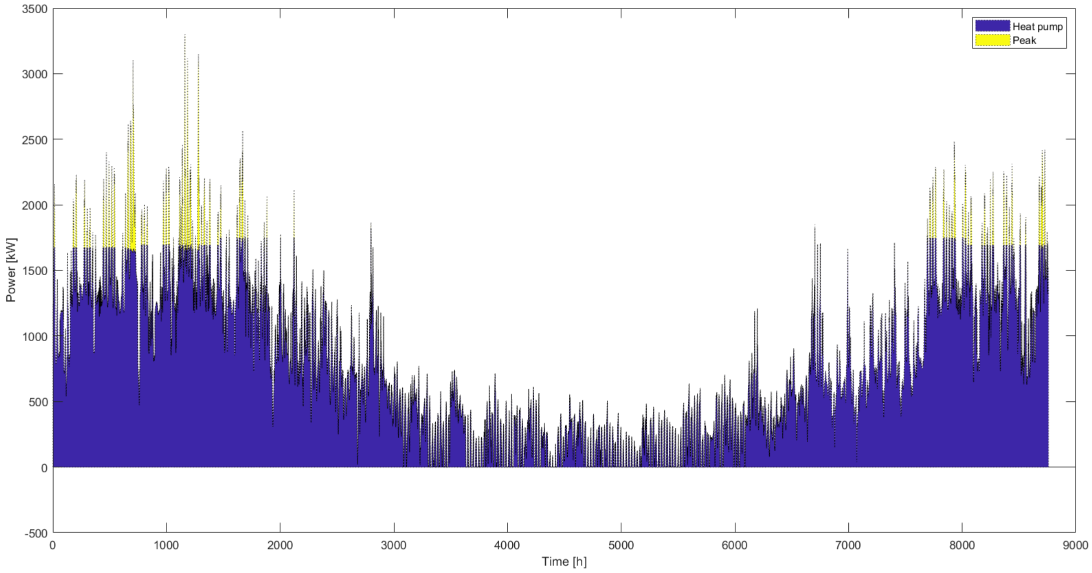


Figure 3.13: Heat pump and peak power for case 2

As seen in figure 3.14, the maximum peak power is slightly lower than for case 1. However, to compensate for the lower temperature, the volumetric flowrate has to increase. The maximum volumetric flow rate for the second case is now 97.1 l/s for supply, 96.9l/s for return, and 97.1 l/s for the peak. The pipe diameter has increased to 269mm, and the highest pressure drop is, 69.1 for supply, 70.2 for return, and 68.7 for the peak. The COP, on the other hand, is drastically better. It looks like the COP is stepping up and down. This is due to the outdoor compensation curve, which lowers the supply temperature to below 35°C most of the time. Again, since the borehole temperature is stepwise, the result will also be stepwise. As in case 1, a COP of zero means that the heat pump is not in operation which is due to no heating power demand from the building.

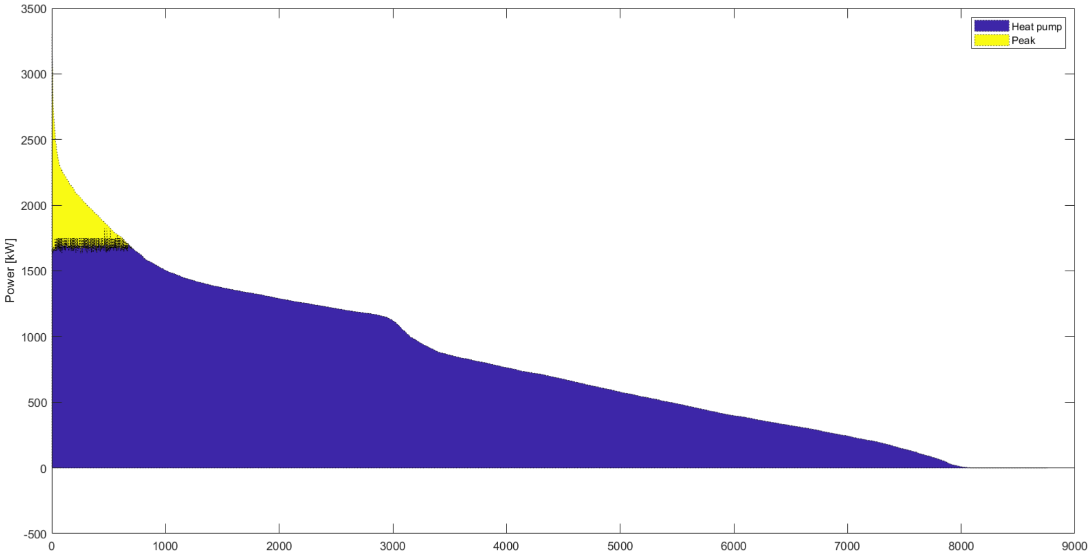


Figure 3.14: Heat pump and peak power for case 2 sorted by demand

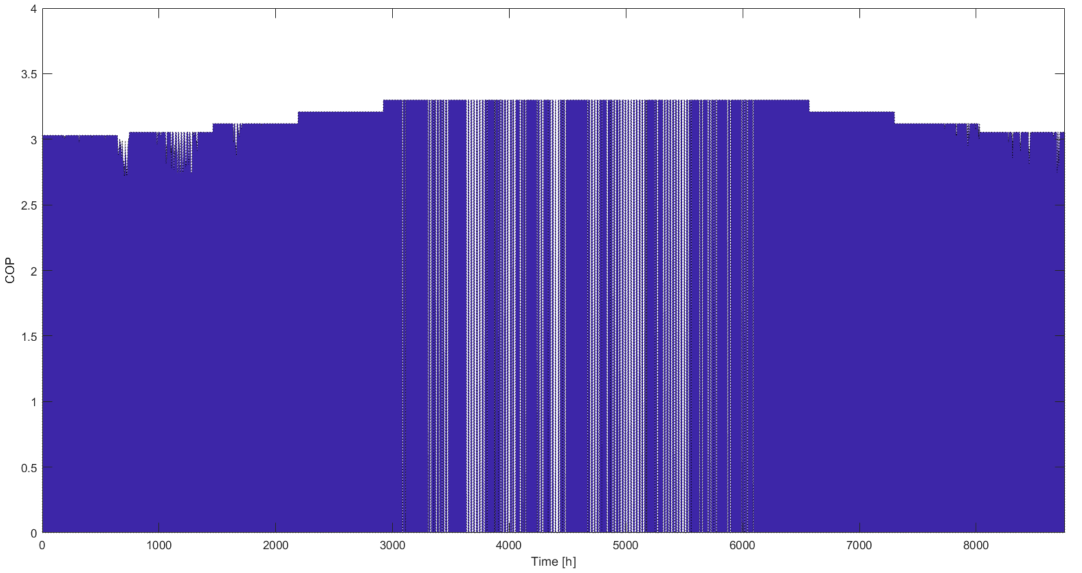


Figure 3.15: COP for case 2

Case 3

In this case, the input values are the same as for Case 1, except the source has been changed from borehole to outdoor air, which results in a much higher peak load.

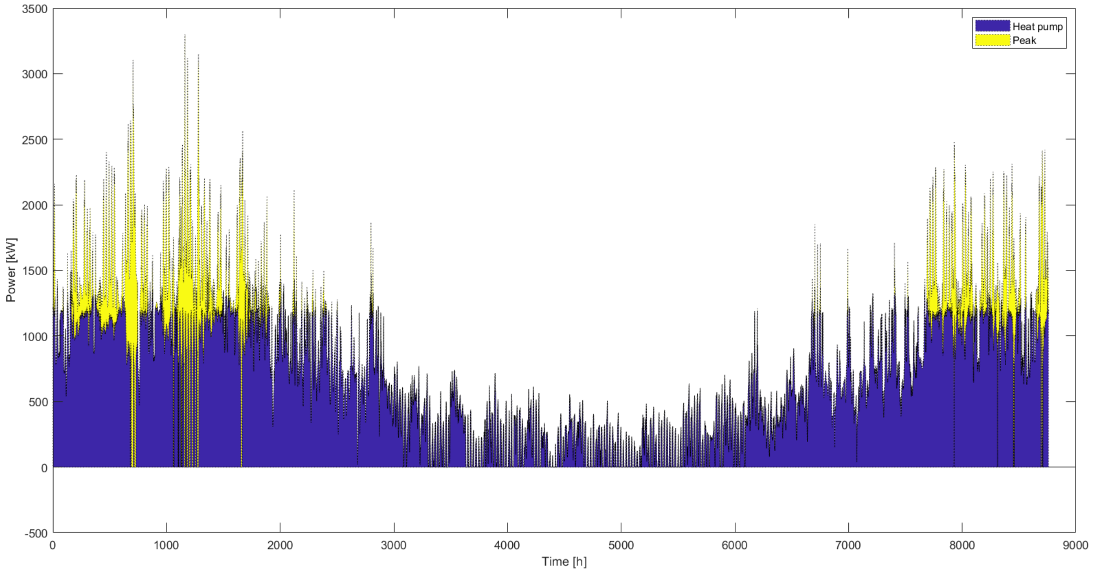


Figure 3.16: Heat pump and peak power for case 3

As seen in figure 3.18, the COP for this case is lower than the COP for case 1. Also, according to the standard, the rated COP for air to water heat pumps is the maximum COP output in contrast to water to brine heat pumps, where the COP can peak at 1.1 times the rated COP.

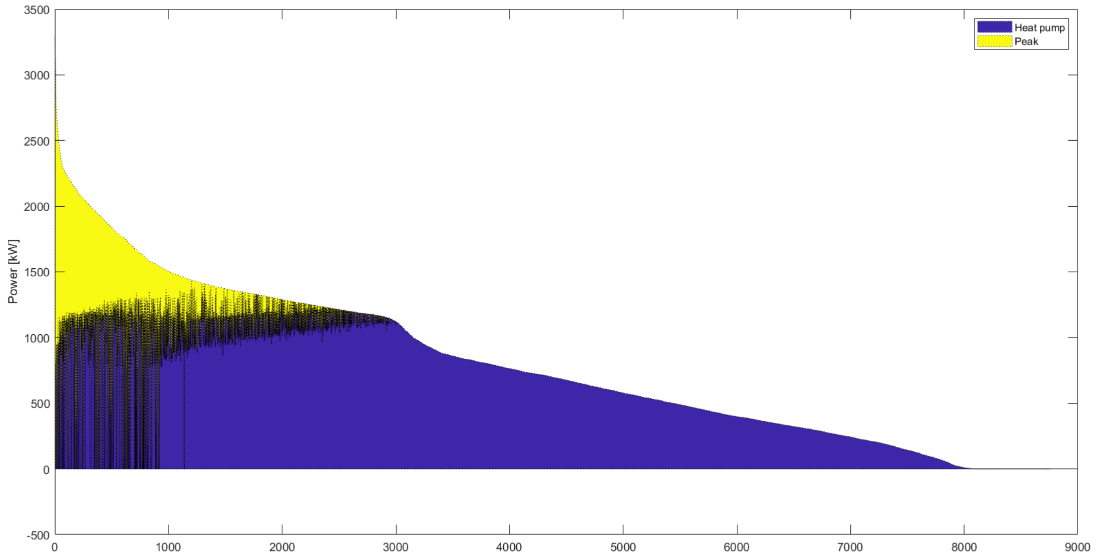


Figure 3.17: Heat pump and peak power for case 3 sorted by demand

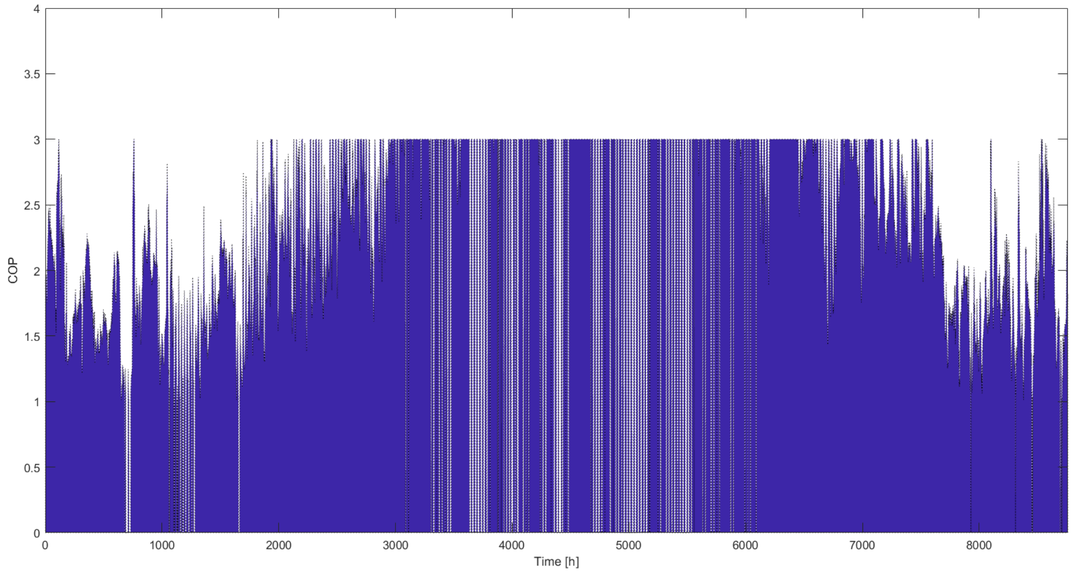


Figure 3.18: COP for case 3

Case 4

In this case, the input values are the same as for Case 2, except the source has been changed from borehole to outdoor air.

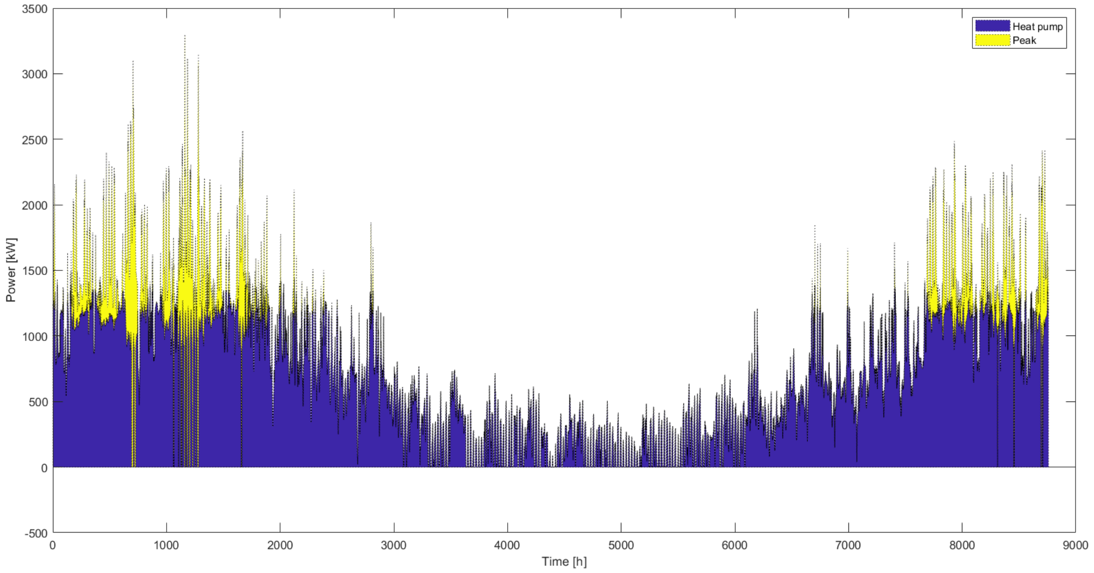


Figure 3.19: Heat pump and peak power for case 4

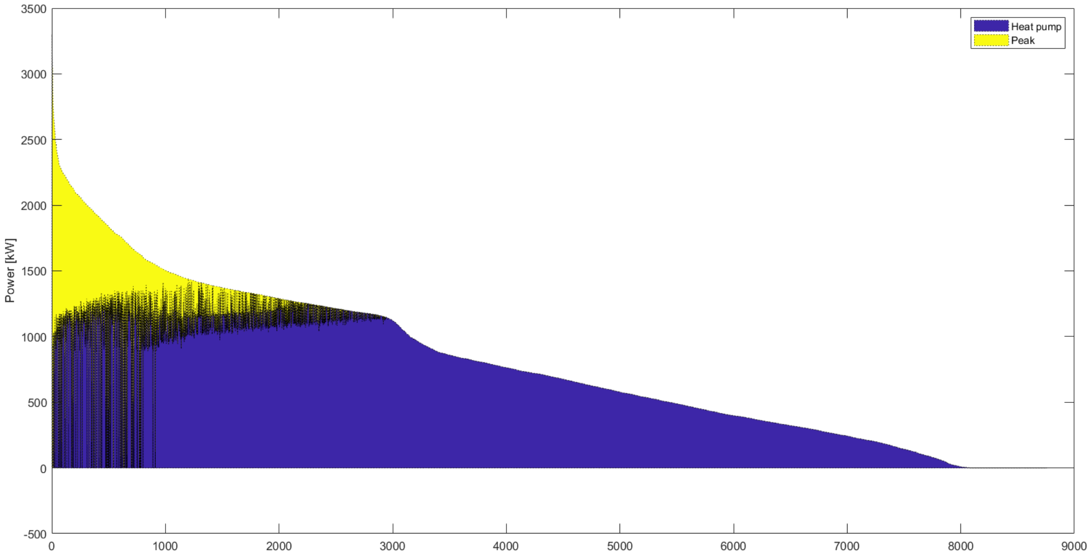


Figure 3.20: Heat pump and peak power for case 4 sorted by demand

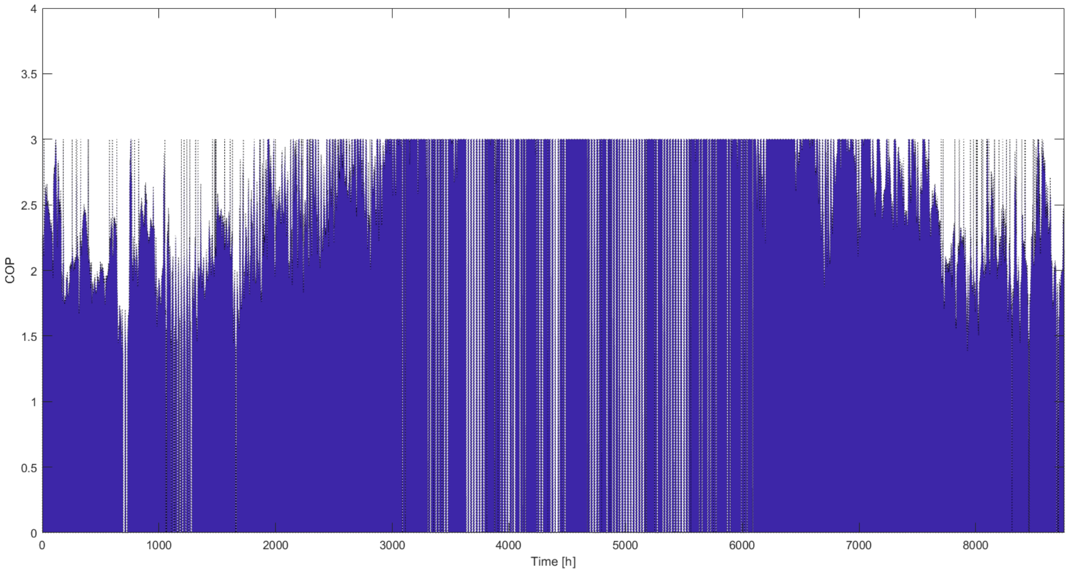


Figure 3.21: COP for case 4

Summary from the cases

In table 3.2, some of the important observations are summarized.

	Unit	**Case 1**	**Case 2**	**Case 3**	**Case 4**
Peak high	kW	1771	1638	3300	3300
Diameter	mm	215.1	269	215.1	269
$\dot{V}_{\text{su,max}}$	l/s	48.7	97.1	48.7	97
$\Delta P_{\text{su,max}}$	pa/m	55.7	69.1	56.1	69.4
Insulation thickness$_{\text{su}}$	mm	60	80	60	80
U'$_{\text{su}}$	W/mK	0.49	0.47	0.49	0.47
Heatloss$_{\text{su,max}}$	W/m	16.9	8	12.2	5.75

Table 3.2: Some results from each case

As seen, with a lower temperature difference, the volumetric flowrate increases. However, a higher flowrate also requires a larger diameter for the pipe in order to keep the pressure drop low. The peak load difference between borehole and air is very high. This is explained by fact that the outdoor temperature is below -15℃ when the highest power demand occurs, and that the source temperature for boreholes are much more stable than the outdoor air temperature.

3.2.3 Visualization

Case 2 has been chosen to present the visual output for the P&ID. Since there are 8760 different results, some hours with values that are interesting have been chosen for the presentation. The chosen values are presented in table 3.3.

Hour NO.	Explenation
1162	Highest power demand during the year
3452	Lowest power demand during the year
8136	Average power demand during the year
1279	Lowest outside temperature for the simulation

Table 3.3: Hours of interest for visualization

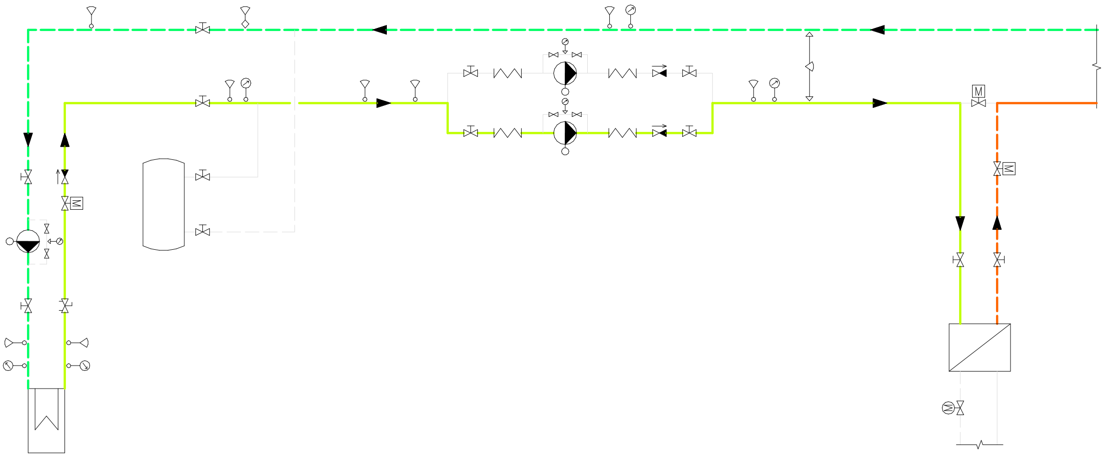


Figure 3.22: Result for hour NO. 1162

As seen in figure 3.22. the heat pump cannot raise the supply temperature to the desired temperature. This means that the flow needs to be further raised by the heat exchanger. After leaving the exchanger, the temperature is higher, which corresponds to the color on the pipe. Note, however, that the color is not entirely red. This is due to the outdoor temperature compensation, which dictates the supply temperature. In this case, the outdoor temperature is higher than -19.8. Note also the thickness of the lines.

In figure 3.23, everything is gray, which complies with the expected value that the heating plant should cease operation when there is not a demand.

In figure 3.24, there is a lower power demand than the maximum. Since the colors are

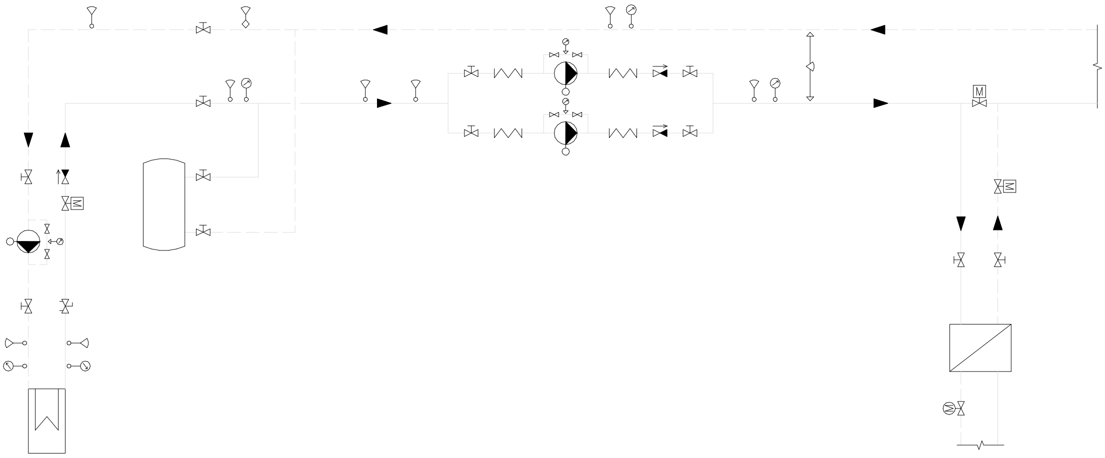


Figure 3.23: Result for hour NO. 3452

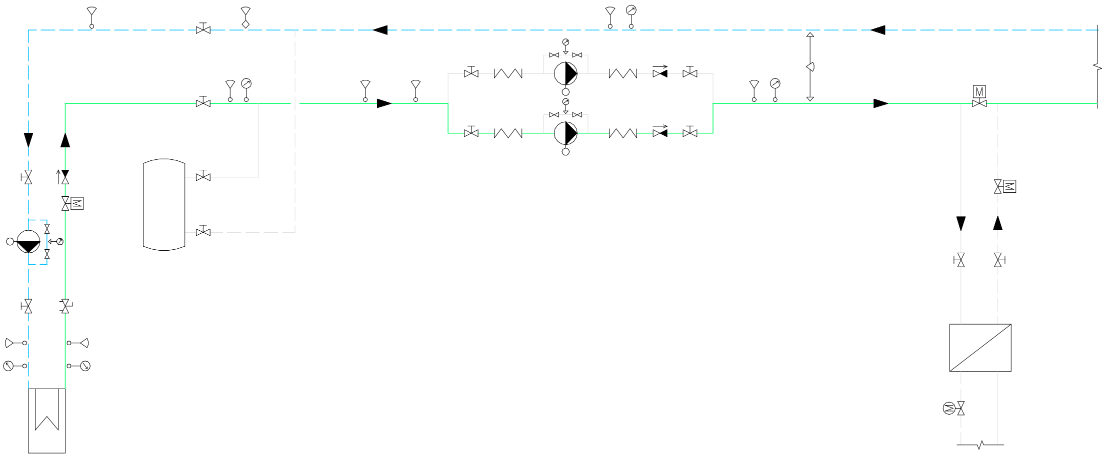


Figure 3.24: Result for hour NO. 8136

relative to minimum and maximum, the supply is green, and the return is bright blue. The lines are also thinner than in figure 3.22, which implies a lower volumetric flow rate.

As in figure 3.22, the heat pump is not able to raise the supply temperature to the

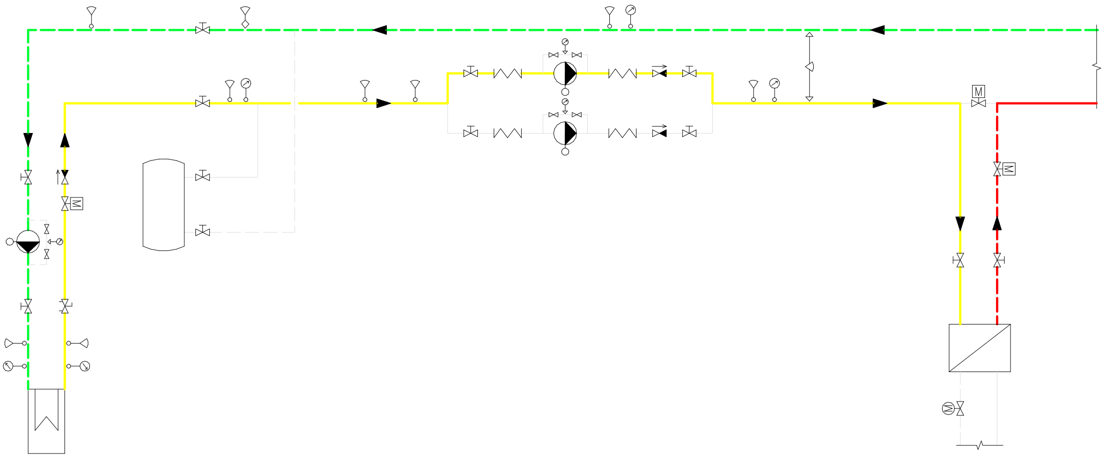


Figure 3.25: Result for hour NO. 1279

desired temperature. However, the supply temperature after the heat exchanger in figure 3.25 is 40°C, which can be seen in the color of the pipe. Note also that now, the upper pump is in operation.

3.2.4 Evaluation of the results

In order to evaluate the method and results, the results can be analyzed by examining the correlation. The correlation of the results is described as

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2}\sqrt{\sum(y-\bar{y})^2}} \tag{3.1}$$

Where an $|r|$ closer to 1 may represent stronger correlation [26].

Considering that the regulation of the temperature is based on the assumption that power-demand for the building is linear to the outdoor temperature, it is interesting to examine if this is the case for the input file.

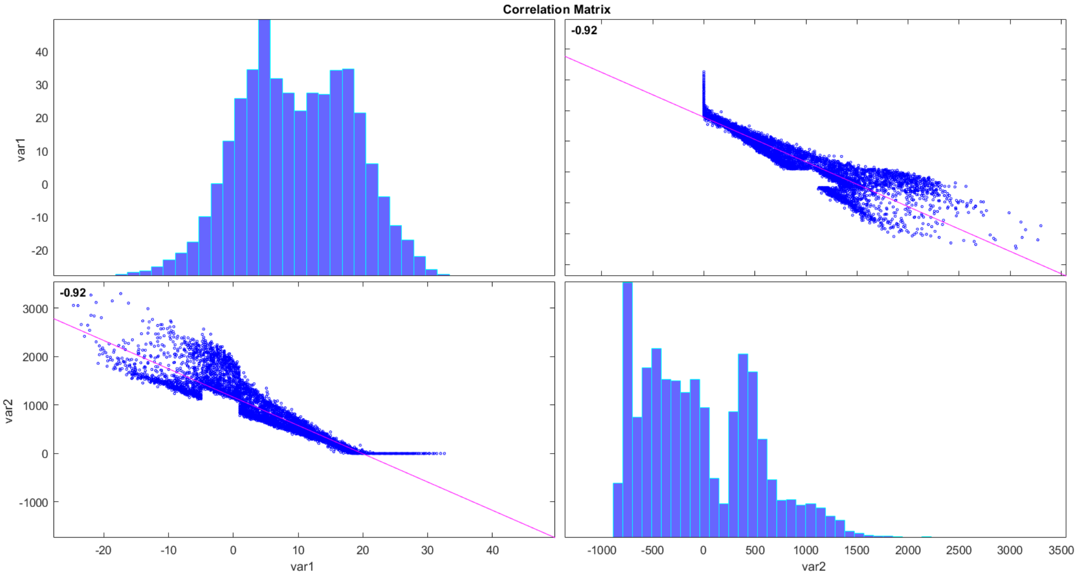


Figure 3.26: Correlation for outdoor temperature and power-demand. $r = -0.92$

As figure 3.26 shows, there is a strong correlation between the outdoor temperature and the power-demand. It is, however, not perfectly correlated, which means that the power demand is not only dependent on the outdoor temperature. This also means that since the supply temperature is dependent on the outdoor temperature, there will be days where the power demand is high even though the temperature is higher than the winter design temperature. In contrast, the supply temperature will be lower than the maximum since there is not a perfect correlation.

By plotting the correlation between the COP and the outdoor temperature, it shows a rather interesting result. As figure 3.27, the COP has steps where it goes up or down. This is related to the way the borehole temperature is calculated, where it is indeed stepped. This is probably not representable for the temperature variation in a real borehole. There are also times when the outdoor temperature is low, and the COP

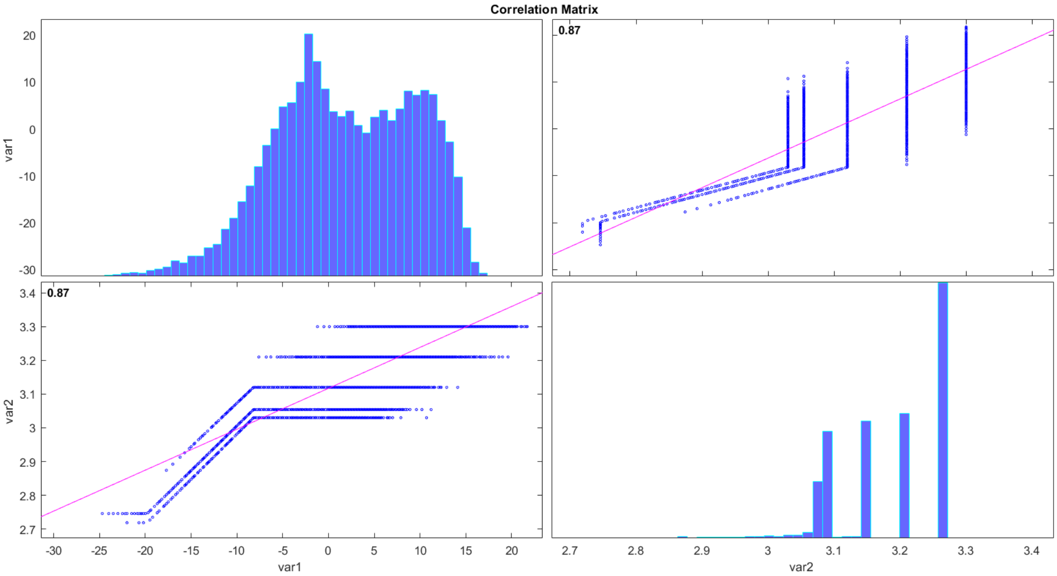


Figure 3.27: Correlation for outdoor temperature and COP for case 2. $r = -0.87$

is high. This is probably due to how the adjusted supply temperature is calculated. The consequence of a power deficiency from the heat pump results in a lower supply temperature than expected. From subsection 2.2.1, the COP is dependent on only the supply temperature and the source temperature. Since the source temperature never drops below the threshold temperature of -5℃, this will result in a high COP, even though the heat pump is not able to raise the temperature of the water sufficiently.

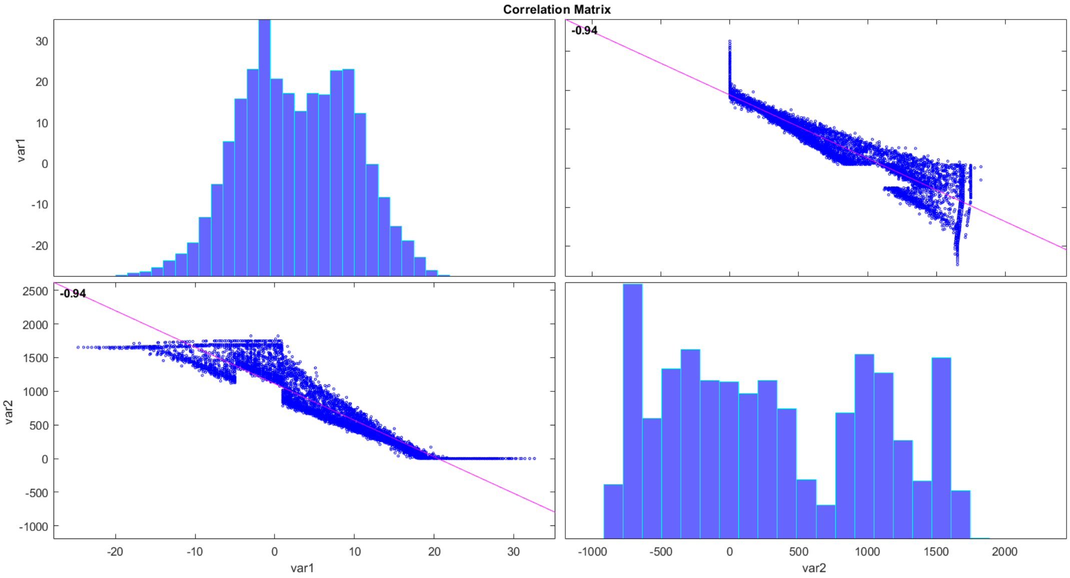


Figure 3.28: Correlation for outdoor temperature and power for case 2. $r = -0.94$

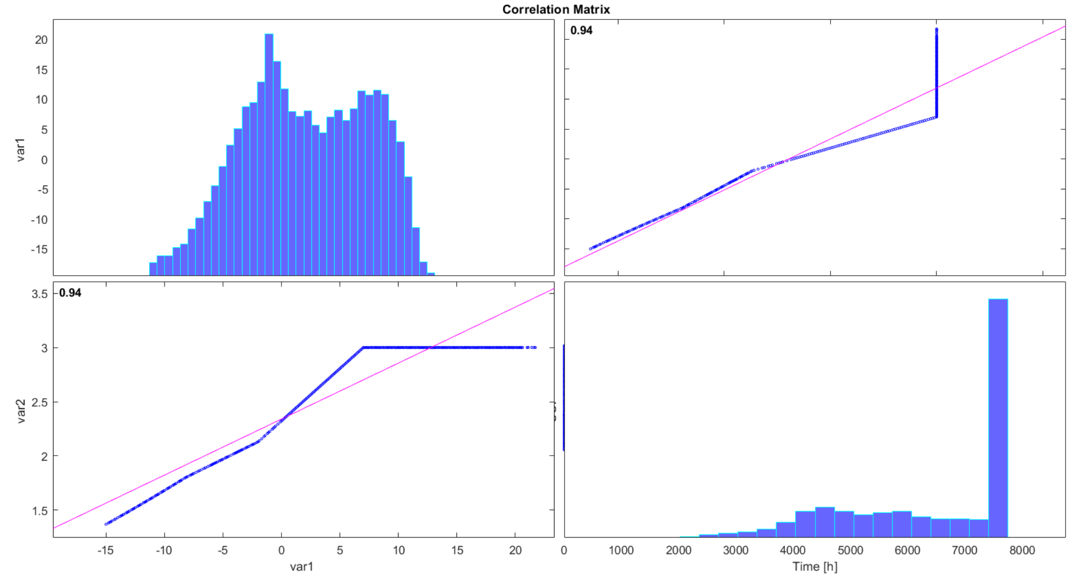


Figure 3.29: Correlation for outdoor temperature and COP for case 4. $r = 0.94$

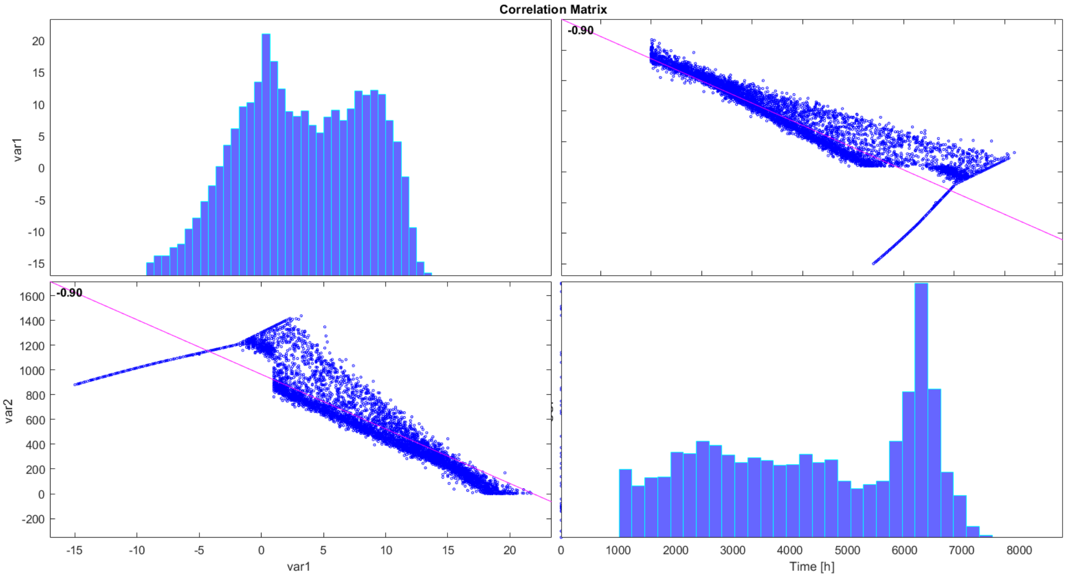


Figure 3.30: Correlation for outdoor temperature and power for case 4. $r = -0.9$

Chapter 4

Discussion

In the thesis, the possibility for creating an Add-in for Revit, which simulates and visualizes the energy flow for a heating plant, has been investigated. The result shows that this is indeed possible. However, the ease of the user is slightly impaired since the user has to input specific identifiers to the elements in Revit so the program can identify the elements. Such a way to construct a program is possibly bad and might lead to wrong results if the input is incorrect. This might be improved if there is a way of importing or connecting the elements in the "drafting view" to the actual 3D model. While this possibility has not been investigated in this thesis, this is absolutely an approach that should be considered.

The performed calculations in the program seem to be of high quality. The difference between a static and dynamically calculated heat capacity is so small that the calculations can be executed with a static heat capacity for water. As previously stated, the difference is minimal, which means that the calculations should either be done according to equation 2.13 or with a static capacity. Furthermore, the pressure has a negligible impact on the results, that it is unnecessary to include pressure in the calculations. Nevertheless, the method is applicable if other systems need higher precision or operate at pressures higher than a conventional heating plant for buildings, which leads us to the next point.

By the looks of it, it seems like this add-in is a bit underdeveloped for the task. There are too many custom dependencies for the test system that it might not work as planned on more complicated systems with another setup. However, it works in simple systems. On the other hand, there are many upsides with the program. The precision is high. It calculates the pressure drop for pipes and the properties for water. By doing so, parts of the program can be extracted to perform their supposed task. The method on how to implement the add-in also seems to be of very high quality. It is suggested that a person with top skills in the field of computer programming should have a look at the code to develop this add-in further.

The results from the calculations show that the simplification that the power demand is proportional to the outdoor temperature is not perfectly correlated. If this was the case, the sorted diagrams should not be jagged when sorted for outdoor temperature, which leads to another question: "is outdoor temperature compensation the best way to regulate the supply temperature for the plant?" Looking at just one calculation without any numbers from an actual heating plant will not answer the question. It does, however, imply that this kind of regulation should be further examined since both the COP and the output power for the heat pump drops with higher supply temperatures and lower source temperatures.

The presentation of the results in the P&ID is solid. The idea of a temperature scale relative to the highest and lowest temperature seems quite appealing and informational. It should, however, contain some information about the scaling. This issue was tried to be dealt with by using the "color scheme" in Revit. However, this is a function that is not available in "drafting view". The workaround to this issue is, of course, that it is possible to click at each element and obtain information about the volumetric flow. The line thickness, on the other hand, is presenting the results poorly. The main issue, in this case, is that the thickness of the line gives rather bad freedom in terms of scaling. Since lines above the thickness of eight give too thick lines, and lines with a thickness of one give almost non-seeable lines, especially when printing. Given the arguments, the idea of scaling lines relative to volumetric flow should be abolished.

Another flaw in the model is that it does not account for thermal energy storage. The reason for this is that the filling and emptying of the storage tanks take place within one hour. In order to consider such behavior, the calculations should be divided into the number of minutes or seconds within a year. By implementing this, the performance of the add-in would drop, which is why thermal energy storage is not implemented. While the pressure drop per meter is precisely calculated, the total pressure drop for the system is not, due to lack of information about the length of the pipes, and other losses such as minor losses due to bends and valves. While valves could be considered, and the length of the pipeline could be solved in the same way as identifying elements, bends cannot be accounted for, because of how a P&ID is presented. Since one of the criteria for this tool was high precision, this is also the reason for not accounting for minor losses or total pressure drop.

In contrast, there are several positive sides to the created program. Given that one of the requirements was the ease of use, and to be able to perform calculations within Revit, the program meets the given conditions. What seems to be one of the most substantial pros with the program is that the user can test different conditions, such as supply and return temperatures, and various power ratings for the heat pump. By doing so, one can consider different possible combinations to find the most optimized combination. As previously stated, there are also several positive parts of the code that can be separated from the main code to perform its task on its own.

Since one of the main tasks with this thesis was to investigate the possibility of creating an Add-in for Revit with C#, the results show that this is av very powerful way to do so. It seems like the possibilities are enormous, and most of the time, it is the developer's limitations that restrict the add-in.

4.1 Conclusion

In this thesis, an add-in for Revit that analyzes a heating plant P&ID has successfully been developed. The add-in utilizes several complex calculations to analyze the COP, power output, water flow, and heat loss of the system. The results are visually presented for the user in Revit. With high precision, this tool makes it possible for engineers to evaluate different combinations of how to design a heating plant, which might reduce CO_2 emissions.

4.2 Future work

Since this add-in has been developed by a person that is not a programmer, improvements can be applied for further development. One of the issues is, for example, that the calculations use about 2.5 minutes to be executed. This can definitely be improved. Since this is somewhat a first attempt to create such a tool, there are possibilities of improvements when it comes to the presentation of the results. Even though the results are written to a .CSV file, it would have been nice to have the graphs directly presented in Revit.

The calculations could also need some further evaluation. While the heat loss and the pressure drop seems to be good, the COP and power calculations could be improved. This is mainly due to the simplified method from the standard. By implementing a more precise method, the results would possibly improve.

In addition, cooling could be added. Since this thesis only investigates a heating plant, a combined heating and cooling plant would be interesting to investigate. The possibility of adding more circuits are also very appealing.

Lastly, the inclusion of thermal energy storage tanks would also improve the add-in and the accuracy of the calculations.

Bibliography

[1] United Nations, “About the sustainable development goals,” [Online] https://www.un.org/sustainabledevelopment/sustainable-development-goals/, 2020.

[2] Statistisk sentralbyrå, “Utslipp til luft,” [Online] https://www.ssb.no/klimagassn, 2019.

[3] C. Markusson, “Efficiency of building related pump and fan operation application and system solutions,” Ph.D. dissertation, CHALMERS UNIVERSITY OF TECHNOLOGY, 2011.

[4] K. S. Hansen, “Dimensjonerende samlet effekt på byggnivå for varme og kjøling i yrkesbygg,” Master’s thesis, Høgskolen i Oslo og Akershus, 2016.

[5] Autodesk, “Lesson 1: The basic plug-in, year=2020, howpublished=[Online], https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/simplecontent/content/lesson-1-the-basic-plug.html .”

[6] The International Association for the Properties of Water and Steam, “Revised Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam,” [Online], http://www.iapws.org/relguide/IF97-Rev.pdf, 2007.

[7] NEMITEK, “Sikkerhetsventiler for vannbårne anlegg - krav,” [Online], URL:http://kompetansebiblioteket.no/Prenok/Distribusjon_Systemer_og_komponeneter_rornett_ventiler_pumper_medier/5_17_Sikkerhetsventiler_for_vannbarne_anlegg_krav.aspx, 2003.

[8] Y. A. Çengel and J. M. Cimbala, *Fluid mechanics third edition in SI units.* Singapore: Mc Graw Hill Education, 2014.

[9] G. S. Kell, “Density, thermal expansivity, and compressibility of liquid water from 0.deg. to 150.deg.. correlations and tables for atmospheric pressure and saturation reviewed and expressed on 1968 temperature scale,” *Journal of Chemical & Engineering Data*, vol. 20, no. 1, pp. 97–105, 1975. [Online]. Available: https://doi.org/10.1021/je60064a005

[10] Direktoratet for byggkvalitet, “Byggteknisk forskrift (TEK17) med veiledning,” [Online], URL:https://dibk.no/byggereglene/byggteknisk-forskrift-tek17/, 2017.

[11] "NS-EN 12828:2012+a1:2014 varmesystemer i bygninger - utforming av vannbaserte varmesystemer," Standard Norge, Standard, 2014.

[12] X.-S. Yang, *Mathematics for Civil Engineers - An Introduction.* Dunedin Academic Press, 2018. [Online]. Available: https://app.knovel.com/hotlink/khtml/id:kt011F8B7A/mathematics-civil-engineers/root-finding-algorithms

[13] R. G. Rice and D. D. Do, *Applied Mathematics and Modeling for Chemical Engineers (2nd Edition).* John Wiley & Sons, 2012. [Online]. Available: https://app.knovel.com/hotlink/khtml/id:kt011C6EI2/applied-mathematics-modeling/successive-substitution

[14] Y. A. Çengel, *Thermodynamics : an engineering approach*, ninth edition. ed. New York: McGraw-Hill, 2018.

[15] "SN-NSPEK 3031:2020 energy performance of buildings calculation of energy needs and energy supply," Standard Norge, Specification, 2020.

[16] D. Zijdemans, *Vannbaserte oppvarmings- og kjølesystemer.* Skarland Press AS: Oslo, 2014.

[17] Norsk VVS Energi -og Miljøteknisk Forening, *Varmenormen.* Norsk VVS Energi -og Miljøteknisk Forening, 2017.

[18] O. Hellevik, *Forskningsmetode i sosiologi og statsvitenskap*, 7th ed. Oslo, Norway: Universitetsforlaget, 2016.

[19] J. Ky, *C# : A Beginner's Tutorial.* USA: Brainy Software, 2015.

[20] JHE, *General information om Revit.* Oslo: NTI, 2018.

[21] Autodesk, "Create a drafting view," [Online] https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/Revit-DocumentPresent/files/GUID-31AB33A6-9403-4DC4-88E5-1C83FFD3E942-htm.html, 2020.

[22] Y. A. Çengel, "Heat and mass transfer : fundamentals and applications," New York, NY, 2015.

[23] Isover, "Climpipe section alu2," [Online] https://www.isover.co.uk/sites/isover.uk/files/assets/documents/datasheet_isover_climpipe_section_alu2.pdf, 2019.

[24] Universitetet i Oslo, "Livsvitenskapsbygget," [Online] https://www.uio.no/tjenester/eiendom/vare-byggeprosjekter/livsvitenskapsbygget/, 2020.

[25] "451.021 Klimadata for termisk dimensjonering og frostsikring," Sintef, Byggdetljblad, 2018.

[26] G. G. Løvås, *Statistikk for universiteter og høgskoler*, 3rd ed. Oslo: Universitetsforl, 2013.

Appendix A

Numerical values of the coefficients and exponents of the dimensionless Gibbs free energy

i	I_i	J_i	n_i	i	I_i	J_i	n_i
1	0	-2	0.146 329 712 131 67	18	2	3	-0.441 418 453 308 46E-5
2	0	-1	-0.845 481 871 691 14	19	2	17	-0.726 949 962 975 94E-15
3	0	0	-0.375 636 036 720 40E+1	20	3	-4	-0.316 796 448 450 54E-4
4	0	1	0.338 551 691 683 85E+1	21	3	0	-0.282 707 979 853 12E-5
5	0	2	-0.957 919 633 878 72	22	3	6	-0.852 051 281 201 03E-9
6	0	3	0.157 720 385 132 28	23	4	-5	-0.224 252 819 080 00E-5
7	0	4	-0.166 164 171 995 01E-1	24	4	-2	-0.651 712 228 956 01E-6
8	0	5	0.812 146 299 835 68E-3	25	4	10	-0.143 417 299 379 24E-12
9	1	-9	0.283 190 801 238 04E-3	26	5	-8	-0.405 169 968 601 17E-6
10	1	-7	-0.607 063 015 658 74E-3	27	8	-11	-0.127 343 017 416 41E-8
11	1	-1	-0.189 900 682 184 19E-1	28	8	-6	-0.174 248 712 306 34E-9
12	1	0	-0.325 297 487 705 05E-1	29	21	-29	-0.687 621 312 955 31E-18
13	1	1	-0.218 417 171 754 14E-1	30	23	-31	0.144 783 078 285 21E-19
14	1	3	-0.528 383 579 699 30E-4	31	29	-38	0.263 357 816 627 95E-22
15	2	-3	-0.471 843 210 732 67E-3	32	30	-39	-0.119 476 226 400 71E-22
16	2	0	-0.300 017 807 930 26E-3	33	31	-40	0.182 280 945 814 04E-23
17	2	1	0.476 613 939 069 87E-4	34	32	-41	-0.935 370 872 924 58E-25

Numerical values of the coefficients and exponents of the dimensionless Gibbs free energy [6]

Appendix B

Factors for heat sources and rejectors

Type	n
Radiator	1.33
Radiator	1.1-1.33
Heating coil	1
Convector	1.5
Heating pipes	1.33
Rib pipes	1.25-1.3

Factors for heat sources and rejectors [16]

Appendix C

Sizes for steel pipes

Inner diameter mm	Outer diameter mm
13.2	17.2
18.1	21.3
23.3	16.9
30.5	33.7
38.4	42.4
44.3	48.3
56.3	60.3
72.1	76.1
84.9	88.9
110.3	114.3
164.3	168.3
215.1	219.1
269	273
319.9	323.9
451	457
504	508

Sizes for steel pipes in Revit

Appendix D

Guide for how to use "EneCalc"

1

Start by downloading the folder, and store it in an appropriate path (It is important to remember the file path).

2

Locate the folder "`\FilMedBilde\FilMedBilde\bin\Debug`", and insert the excel file containing the calculations. Make sure to rename the excel-file to "EneCalcInput".

3

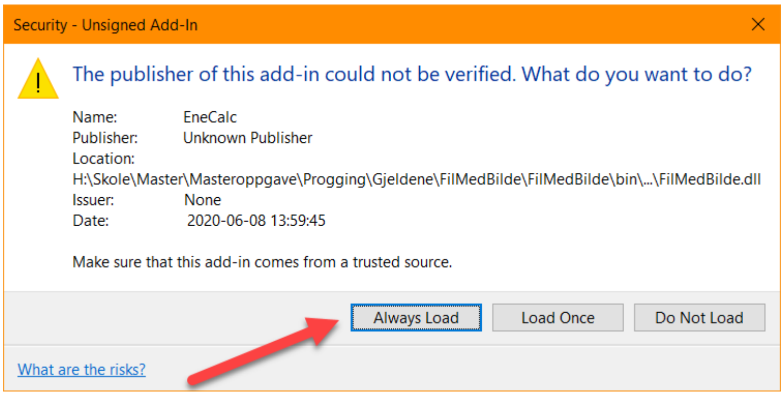
Copy the file "EneCalc.addin", locate the folder "`C:\ProgramData\Autodesk\Revit\Addins\20xx`". (The two last digits should comply with the Revit version you're using). Then paste the file. If you cannot locate the folder "ProgramData", make sure that you have turned on "Show hidden folders" in the file explorer

4

Open the add-in file in notebook, and replace the string "(REPLACE THIS)" with the file path where you stored the folder with the add-in.

5

Start Revit. When prompted with this window, make sure to press "Always Load".



Security - Unsigned Add-In

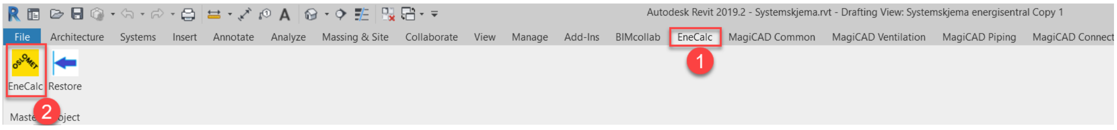
The publisher of this add-in could not be verified. What do you want to do?

Name: EneCalc
Publisher: Unknown Publisher
Location:
H:\Skole\Master\Masteroppgave\Progging\Gjeldene\FilMedBilde\FilMedBilde\bin\...\FilMedBilde.dll
Issuer: None
Date: 2020-06-08 13:59:45

Make sure that this add-in comes from a trusted source.

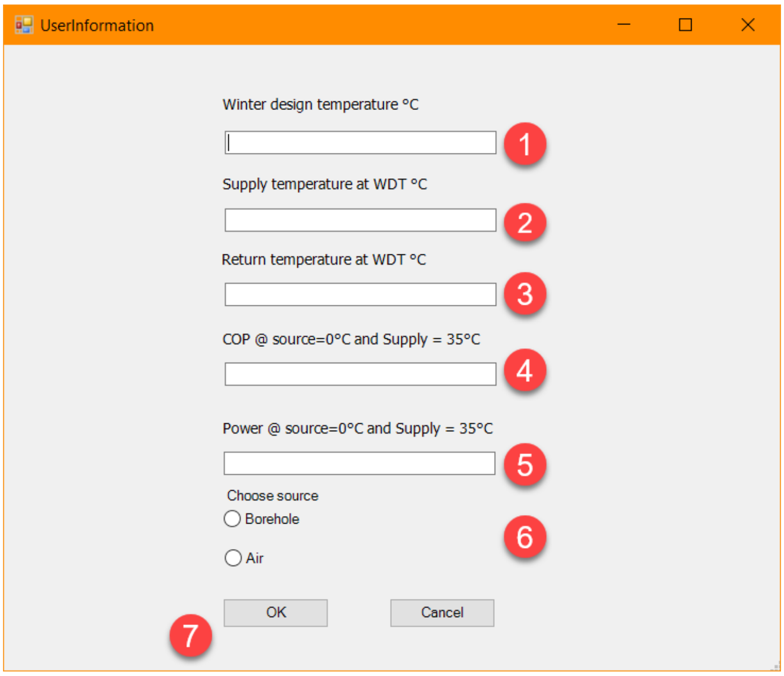
6

Locate the “EneCalc” tab, then click on the“Enecalc” button.



7

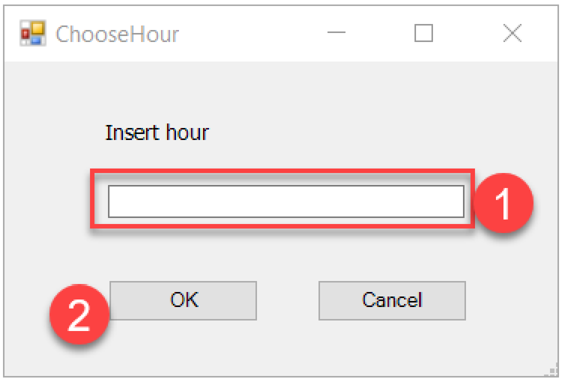
You will now be prompted with an input window.



1. Insert the winter design temperature for the place
2. Insert the designed supply temperature at winter design temperature
3. Insert the designed return temperature at winter design temperature
4. Insert the COP from the manufacturer
5. Insert the power from the manufacturer
6. Choose between either air or borehole as source
7. Press "OK"

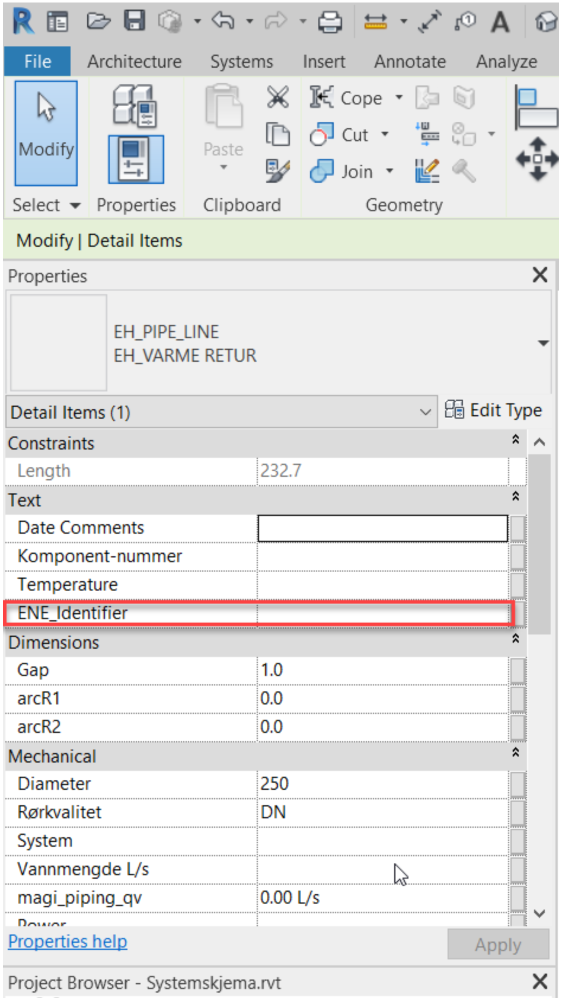
8

After the calculations are finished, you will be prompted to choose a specified hour. Insert the hour you want to investigate, then and press "OK".


ChooseHour
Insert hour
1
2
OK
Cancel

9

If this is the first time you’re doing these calculations on the model, you need to place identifiers to the elements. Click on each element of interest, and locate the parameter “Ene_Identifier”.



Constraints	
Length	232.7
Text	
Date Comments	
Komponent-nummer	
Temperature	
ENE_Identifier	
Dimensions	
Gap	1.0
arcR1	0.0
arcR2	0.0
Mechanical	
Diameter	250
Rørkvalitet	DN
System	
Vannmengde L/s	
magi_piping_qv	0.00 L/s

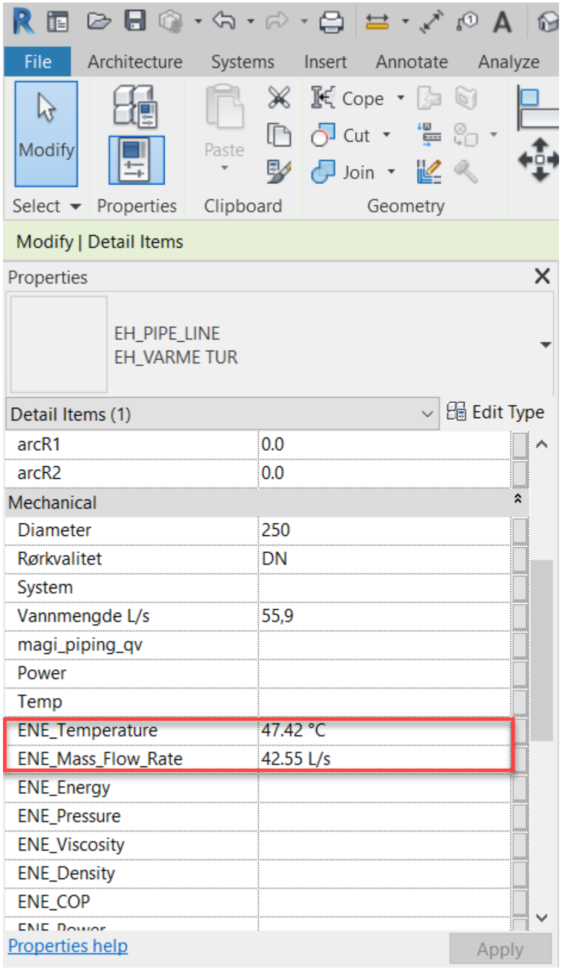
- Insert “x” for pipes that should not have any flow
- Insert “p1” on all the pipes connected to the first pump
- Insert “p2” on all the pipes connected to the second pump
- Insert “peakSupply” on the pipe that is supplying the peak load

- Insert “peakReturn” on the pipe that is returning from the peak load
- Insert “peakSupply” on the pipe that is supplying the peak load
- Insert “peakReturn” on all the supply pipes after the peak return pipe is connected to the supply circuit
- Insert “peakVale” on the pipe where the magnetic solenoid is connected

Press EneCalc again and fill inn the same hour value to update the result.

10

You can click on each element to read information about the element.



EH_PIPE_LINE
EH_VARME TUR

Detail Items (1)

arcR1	0.0
arcR2	0.0
Mechanical	
Diameter	250
Rørkvalitet	DN
System	
Vannmengde L/s	55,9
magi_piping_qv	
Power	
Temp	
ENE_Temperature	47.42 °C
ENE_Mass_Flow_Rate	42.55 L/s
ENE_Energy	
ENE_Pressure	
ENE_Viscosity	
ENE_Density	
ENE_COP	
ENE_Power	

If you want to see more hours, just press the "EneCalc" again, and fill inn a new hour.

11

To restore the colors to its original state, press the "Restore" button.

12

A .CSV file named "EneResults" with all the results will be stored in the folder you saved the file in. Open it to analyze the results. If you want to perform several simulations with different inputs, please move the result file to another folder.

Appendix E

Add-in manifest

```
<?xml version="1.0" encoding="utf-8"?>
<RevitAddIns>
    <AddIn Type="Application">
        <Name>EneCalc</Name>
        <Assembly>(REPLACE THIS)\FilMedBilde\FilMedBilde\bin\
            Debug\FilMedBilde.dll</Assembly>
        <FullClassName>FilMedBilde.App</FullClassName>
        <VisibilityMode>AlwaysVisible</VisibilityMode>
        <ClientId>07FA8EB9-0A2B-4C9E-9799-2B054C57D120</ClientId
            >
        <VendorId>ADSK</VendorId>
        <VendorDescription>Autodesk, Inc, www.autodesk.com</
            VendorDescription>
    </AddIn>
</RevitAddIns>
```