

# DISTRICT ENERGY NETWORK SIMULATION FOR THE EVALUATION OF SOLAR THERMAL POTENTIAL

Master Thesis  
handed in on August 7th 2020  
Physics Section

École Polytechnique Fédérale de Lausanne  
to obtain a Master in Applied Physics  
by

Pierre Cognet

Under the supervision of :

Jérôme Kämpf, Founder and Managing Partner at kaemco  
Prof Paolo Ricci, Swiss Plasma Center Theory group, EPFL

Martigny, 2020





# Acknowledgements

I thank Jérôme Kämpf, my "boss" at kaemco during this thesis in the industry. He has always taken the time to explain things and to guide me. I thank Paolo Ricci, my EPFL thesis supervisor, for help on technical subjects and for the support. I thank Thierry Ursenbacher, Giona Galizia and Loïc Puthod with whom we had fun modelling all kinds of hydraulic systems. I thank RWB Groupe SA, Altis and Satom SA, the companies that have provided data on district networks, which are crucial to help validate the models. I thank the CREM and all its members, for being excellent workplace neighbours and for their cooperation. And last but not least, I thank all of my family and friends who are always there to support me.

*Martigny, August 7th 2020*

P. C.



# Abstract

More than ever, it is desired to decrease our dependence and use of non-renewable, greenhouse gas emitting, imported energy sources. Most of the heating in Switzerland is currently based on fossil gas and fuel oil. A solution explored to become more independent from these sources is the development of low-fossil-carbon district energy networks, powered by biomass and solar thermal energy. However, no tool has been found that can simulate in detail the consumer heat demand in district networks with complex topologies that have solar injection. This was identified in the present thesis as a crucial need to build a well-sized network.

This thesis aims to implement such a simulation tool, free and open source. It is based on the CitySim software, which can already make detailed urban physics simulations. This thesis describes the implementations and models that have been added to the software to simulate heating or cooling district networks. Efforts are made to make it possible to simulate complicated networks, such as looped topologies. There is the possibility to add multiple thermal stations to produce heat or cold. Also the integration of solar power injection in centralised or decentralised ways is explained and implemented.

The simulation tool has been tested against real data from district networks in Switzerland. The results of the simulations are satisfying, and underline the importance of having good complete data of buildings and networks to simulate. The potential of the CitySim tool to simulate solar district networks is then shown through different tests.

Key words: energy, district energy network, district heating network, district cooling network, renewable energy, low-carbon energy, biomass, solar, solar thermal, solar thermal potential, looped topology, hydraulics, thermal, simulation, CitySim



# Contents

|  |            |
|--|------------|
| <b>Acknowledgements</b>                              | <b>i</b>   |
| <b>Abstract</b>                                      | <b>iii</b> |
| <b>1 Introduction</b>                                | <b>1</b>   |
| 1.1 General context . . . . .                        | 1          |
| 1.2 State of the art . . . . .                       | 6          |
| 1.3 Problematic . . . . .                            | 7          |
| 1.4 Thesis content . . . . .                         | 7          |
| <b>2 Original status of CitySim</b>                  | <b>9</b>   |
| 2.1 Building thermal simulation . . . . .            | 9          |
| 2.2 District heating simulation . . . . .            | 10         |
| 2.2.1 Substation . . . . .                           | 11         |
| 2.2.2 Network . . . . .                              | 14         |
| 2.2.3 Central thermal station . . . . .              | 16         |
| <b>3 Methodology</b>                                 | <b>19</b>  |
| 3.1 Branched code . . . . .                          | 19         |
| 3.1.1 Informing effectively provided power . . . . . | 19         |
| 3.1.2 Substations . . . . .                          | 19         |
| 3.1.3 Thermal station . . . . .                      | 20         |
| 3.1.4 Network . . . . .                              | 20         |
| 3.2 Looped code . . . . .                            | 23         |
| 3.2.1 Looped topology . . . . .                      | 23         |
| 3.2.2 Pressure . . . . .                             | 24         |
| 3.2.3 Graph . . . . .                                | 25         |
| 3.2.4 Loops . . . . .                                | 26         |
| 3.2.5 Kirchhoff's nodal rule . . . . .               | 28         |
| 3.2.6 Kirchhoff's loop rule . . . . .                | 29         |
| 3.2.7 Pipes . . . . .                                | 30         |
| 3.2.8 Valves . . . . .                               | 33         |
| 3.2.9 Pumps . . . . .                                | 34         |
| 3.2.10 Hydraulic convergence . . . . .               | 35         |

|   |           |
|---|-----------|
| 3.2.11 Thermal convergence . . . . .  | 41        |
| 3.2.12 Overall convergence of valve positions and pump rotations . . . . .  | 42        |
| 3.2.13 Robbins-Monroe annealing . . . . .                                   | 43        |
| 3.2.14 PID controller . . . . .   | 45        |
| 3.2.15 PID Continuous Action Reinforcement Learning Automata [22] . . . . . | 46        |
| 3.2.16 Combination of Ideal and PID . . . . .                               | 49        |
| 3.3 Solar thermal feed-in . . . . .   | 50        |
| 3.3.1 Solar thermal collectors . . . . .                                    | 50        |
| 3.3.2 Integration in district heating networks . . . . .                    | 51        |
| 3.3.3 Prosumer substation . . . . .   | 55        |
| 3.3.4 Thermal stations with seasonal storage . . . . .                      | 60        |
| 3.3.5 Overall Control . . . . .   | 63        |
| 3.4 Miscellaneous contributions . . . . .                                   | 63        |
| <b>4 Results and discussions</b>  | <b>65</b> |
| 4.1 Branched network Broc case study . . . . .                              | 66        |
| 4.1.1 Network geographic data . . . . .                                     | 66        |
| 4.1.2 Exterior temperature . . . . .  | 70        |
| 4.1.3 Substation consumption . . . . .                                      | 71        |
| 4.1.4 Thermal station measurements . . . . .                                | 74        |
| 4.2 Looped network . . . . .  | 81        |
| 4.2.1 Control strategy . . . . .  | 81        |
| 4.3 Solar thermal . . . . .   | 81        |
| 4.3.1 Control strategy . . . . .  | 81        |
| 4.3.2 Decentralisation . . . . .  | 82        |
| 4.3.3 Setpoint supply temperature . . . . .                                 | 87        |
| 4.3.4 Seasonal storage . . . . .  | 88        |
| <b>5 Conclusion</b>   | <b>91</b> |
| <b>Bibliography</b>   | <b>98</b> |

# 1 Introduction

## 1.1 General context

Our modern society is deeply dependent on energy. Energy is consumed in many different forms to improve our quality of life. It is used to heat us in winter, to make us move in many different types of vehicles, in the industry to manufacture goods that we use, to preserve our health, and so on. Figure 1.1 shows a detailed overall view of the energy extraction and use in Switzerland.

For most of humanities history, the energy supply was limited and sparsely used. It consisted of the plants we ate to get calories, the wood we burned, the sun that shined to dry or heat, the wind that blows in the sails of our boats and mills, etc. During the industrial revolution of the 19<sup>th</sup> century, machines were discovered that can consume fossil fuels and extract their huge energy content, and convert it to useful mechanical energy. Ever since, the fossil energy consumption in the work has not stopped increasing.

However, it is desirable to move away from this dependence. Firstly because fossil fuels will eventually run out. When they do, it will be more desirable to be prepared and not to suffer this reduction in the fossil energy consumption. A fast uncontrolled reduction in energy consumption will likely lead to a reduction in comfort of life, which may pose political stability problems.

Secondly, it is desirable to move away from this dependence as the combustion of fossil fuels emits large quantities of greenhouse gases, among which carbon dioxide. The carbon in fossil fuels was stuck in the Earth's crust for a long time. In only a couple of decades, mankind is releasing large amounts into the atmosphere in the form of carbon dioxide. This pollution is causing climate change, which will be the cause of many problems. Populations in the equatorial countries will need to migrate to avoid lethal heat waves [35]. These large migrations will likely cause political and social instability possibly leading to war. Also the overall global agricultural yields will decrease. Since the current demographic models predict an increase in the world population, there will likely be challenges posed to feed everyone.

Figure 1.2 describes the greenhouse gas emissions of Switzerland. First one can observe the increase in emissions over the last century. Almost all these sectors emit, mainly transport, industry, households and agriculture. It means that reduction efforts will spare none of these sectors. As can be seen in Figure 1.1, a large part of the industry energy supply in fossil gas, which is mainly used for industrial process heating [2]. In households, 64% of the energy

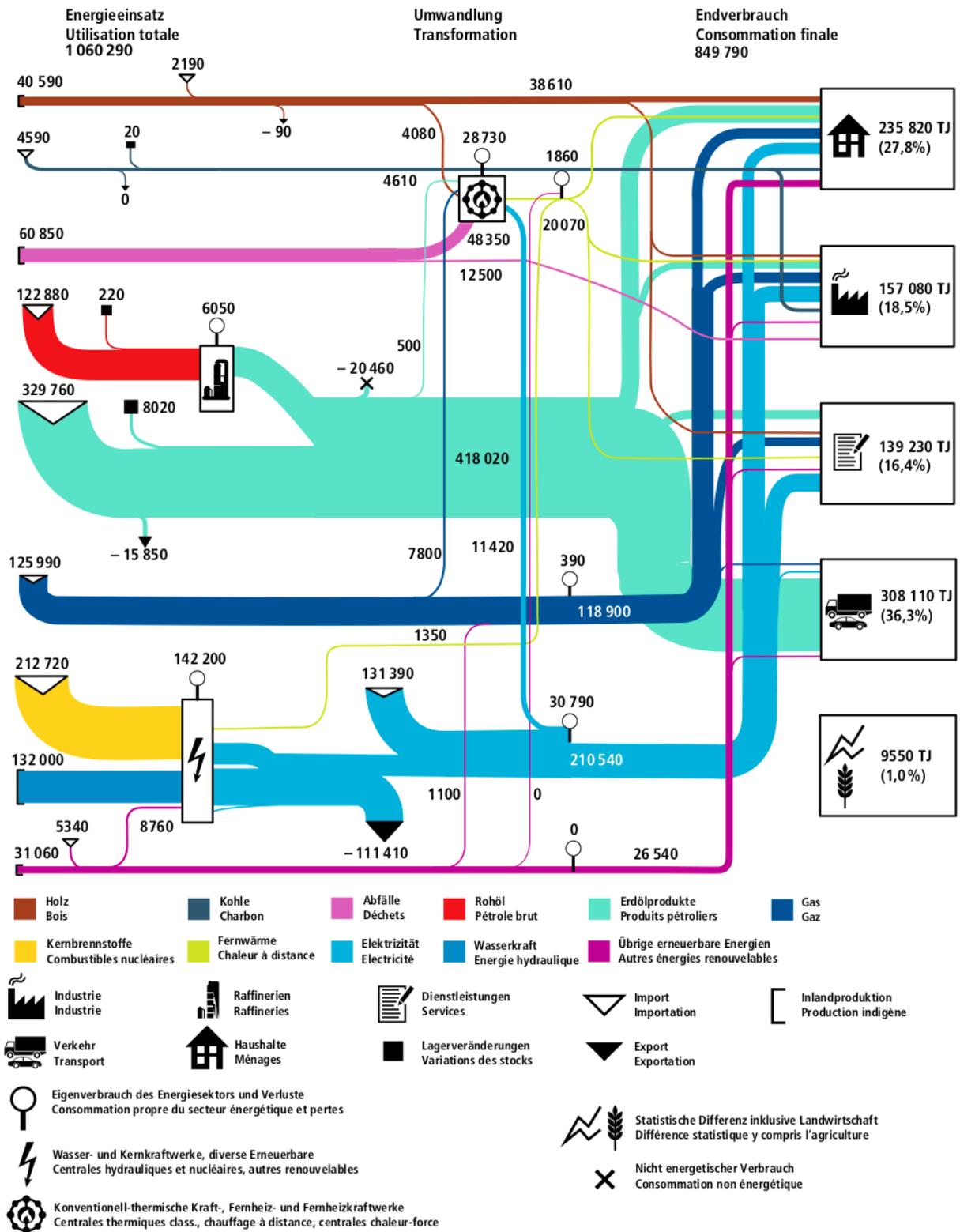


Figure 1.1 – Detailed energetic flux diagram in Switzerland 2017 (in TJ) [48].

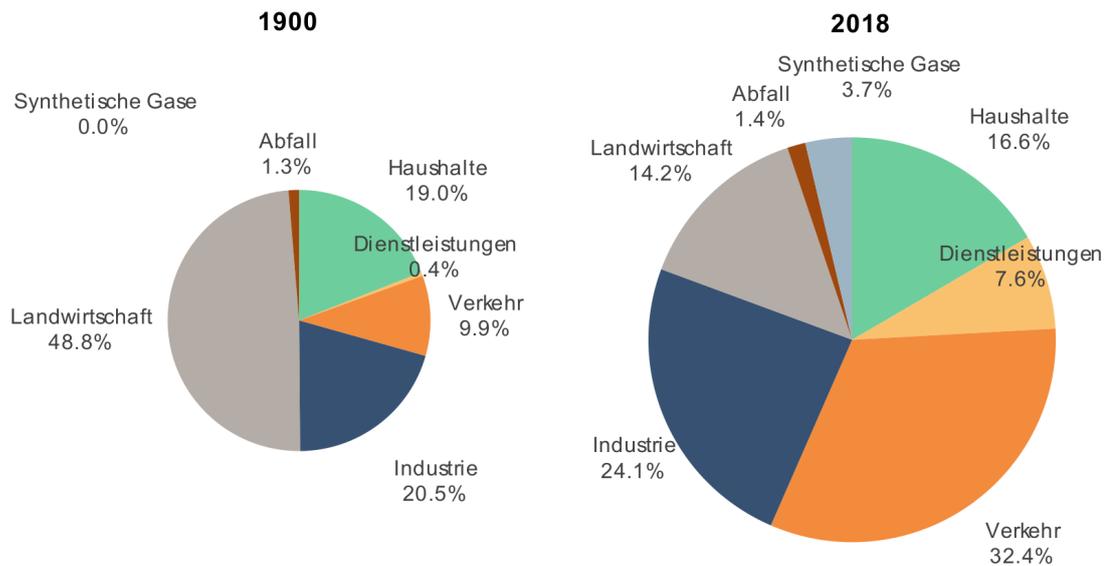


Figure 1.2 – Greenhouse gas emissions of agriculture (Landwirtschaft), industry (Industrie), transport (Verkehr), services (Dienstleistungen), households (Haushalte) and synthetic gases (Synthetische Gase) in Switzerland [54].

consumed is for space heating and 15% for domestic hot water [15], of which 56% comes from fossil gas and fuel oil.

It can be noted that thermal energy has a lower exergy compared to electricity. This can be understood as thermal energy being of lower quality than mechanical energy. For example, with electricity, one can heat an object with the Joule effect, theoretically with no loss of energy. But with that hot object, one cannot get back the same amount of electricity. One must combine that hot object with a cold source, through a Carnot cycle, and the theoretical yield is limited by the Carnot efficiency, as some heat has to be lost, given to the environment. It is therefore sometimes interesting to directly use heat instead of converting it into electricity, then converting it back to heat. One kWh of electricity can be used to harness more than one kWh of heat, by taking it from the environment, if a heat pump is used.

Therefore, there is a challenge in reducing the emissions linked to heat production. A solution would be of course to consume less. For example heating less the houses, at lower temperature during the winter, or not the whole house. Some office building have timers to stop heating during the night and weekends. But there is still a need for domestic hot water for showers and baths, to wash dishes and clothes. In the chemical industries such as pharmaceutical industry, heat is also necessary to get certain reactions. There are many efforts to be done in reducing consumptions, but it will be difficult to get completely rid of it.

There is a need to decarbonise the heat production. In the current technologies, it means replacing fossil gas and fuel oil to sources which emit less fossil carbon dioxide such as heat pumps running on decarbonised electricity, solar thermal, geothermal, renewable wood extraction, etc. There is also a need in improving efficiencies of the heat production systems.

This can be done by centralising the heat production unit, to heat a fluid, then sent that hot fluid via insulated pipes to the heat consumers. This is called a district heating network, where the thermal stations produce heat and distribute it to the substations of each consumer. Similarly to electricity networks, they are used to share to transport heat between producers and consumers. This can for example be used at an industrial site that must evacuate heat from its process, to make it available for another consumer, which can be source to heat recycling and improved efficiencies.

Wood as a means of heat production has the advantage that it doesn't put fossil carbon into the atmosphere. Biomass in general, when growing, absorbs almost all its carbon from the CO<sub>2</sub> in the atmosphere during the photosynthesis reaction. Forests are therefore carbon stocks. When the biomass is burned, the CO<sub>2</sub> from the combustion is restored to the atmosphere. If the biomass wood doesn't come from deforestation, the combustion doesn't add new CO<sub>2</sub> to the atmosphere. Since chainsaws to cut down tree, trucks to transport logs and sawmills to cut are powered by fossil, biomass with no carbon-stock depletion still has a non-zero carbon footprint and emits overall around 17.6 gCO<sub>2</sub>eq/kWh [24], which can vary depending on the cases of course. In the future these operations could become less fossil dependant of course. Wood is also a local source of heating. So it creates less dependence on external countries, which may not always supply Switzerland in the future if they need the energy sources for themselves. It also creates local jobs, and has to be transported smaller distances. Wood has the disadvantage of creating local air pollution which causes health problems to local populations. Wood combustion emits notably NO<sub>2</sub> and CO. Carbon monoxide is toxic to humans, it attaches to hemoglobine restricting oxygenation. Nitrogen dioxide causes lung irritation, acid rain, formation of aerosols and smog. Its extraction is also requires dedicated forest surfaces, which means it is taking surfaces away from other activities such as agriculture, housing, etc. It is estimated that more than half of the renewable biomass resources in Switzerland are already exploited [8].

Solar thermal includes a range of technologies which directly use the sunlight to get heat, unlike the more famous photovoltaic panels which produce electricity. It is by nature a renewable local source of energy, but is intermittent. It is currently a very little exploited resource, there is a potential to use building rooftops to install solar thermal collectors. This is changing, it is now mandatory in Switzerland that new buildings produce at least 10% of the heat through renewable sources. Solar thermal is an easy way to reach that objective by heating domestic hot water in summer. Characteristics of this technology are described in further detail in section 3.3.1.

For these reasons, these are interesting technologies that are being pushed by the Swiss Federal Office of Energy (SFOE) in plans for the future. The current projections are that in the following years, the number of low-carbon district heating networks, especially those using biomass and solar should continue to increase. In this goal, the SFOE has launched a call [13]. We have prepared a proposal to answer the call named SolCAD. SolCAD stands for "Solar CAD", where the second term is the French abbreviation for "Chauffage à distance", meaning "district heating network". This thesis takes part in the project. The goals of SolCAD are to :

- Extend the evaluation of the potential of solar thermal energy coupled with heat storage

in district heating networks in Switzerland, while taking into account the technological and socio-economical constraints.

- Validate and optimize the energetic and economic performances of solar district heating networks in Switzerland, including different combinations of technologies, such as solar thermal with other renewable energy sources, co-generation and storage.
- Analyse the success factors and limiting factors for the integration of solar thermal in district heating networks (DHN) using a DHN archetype approach.

Among the participants of this research project are :

- CREM : a non-profit association founded in 1986 by the City of Martigny and the EPFL. The CREM is a centre for R&D, services and information, active in the field of urban energy, particularly with regard to its sober use as well as its production, distribution and storage.
- HEIG-VD : the School of management and engineering of canton Vaud.
- kaemco : a spin-off company of the Solar Energy and Building Physics Laboratory (LESO-PB) of EPFL. Founded in 2013 by Dr Jérôme Kämpf, senior researcher at the Idiap Research Institute. kaemco aims at adding the dimensions of energy and comfort to 3D geometrical data. It provides consulting in energy and urban physics. It is the company employing me during this thesis.
- Planair, a consulting engineering firm for sustainable development.

The role of this thesis in the project is to validate and optimize the potential of archetypes by modelling case studies based on concrete cases.

Biomass district heating networks face challenges. In summer, the heat consumption is much less than in winter, there is no more space heating needed, but only domestic hot water. This means that the yield of the over-sized boiler decreases. This usually leads to the boiler being turned on and off repeatedly, leading to inefficiencies and damages to the network components which are subject to thermal shocks. A solution that needs to be studied is integrating solar thermal energy, which produces more during the summer. However, this solar of energy isn't a dispatchable means of production, meaning that its operation and integration is more difficult.

Building such a district heating network poses challenges, especially in estimating the correct size of the components. A difficulty is to have good models of everything at once : predict the heat needs of the consumer buildings, the thermal solar panels, the thermal energy storage, the heating network pipes with their losses. This leads to over-sizing of many components of the network, which leads to energy inefficiencies, along with high operation fees for the network providers. This makes prices higher than they could be, sometimes making it more expensive than if consumers simply used a fossil boiler. This is not desired for the economical development of these low fossil carbon technologies.

## 1.2 State of the art

The current scientific literature has been searched, to look for answers to the challenges presented in section 1.1, on the sizing of renewable district heating networks.

There are some studies and papers on this subject. Some have very complete simulations of district heating networks of complex topologies. Others have very good models for the building heat demand. Others model well the solar thermal collectors and its storage. However it is difficult to find studies that model all these aspects well. It is also difficult to find a tool that are free and open source.

Wang [56] has a simplified model, which only works for branched networks. The study performed is on a toy district heating network, made for the occasion. The goal is to analyse the effects of decentralized solar of the network performances. The heating demand for buildings is based on a simplified model from [32] that doesn't take into account detailed geometry of the building. Only two types of demand profiles are available : residential and office. It does not seem that this simulation model is freely and openly available. The substation's demand is imposed as true. And is not described how the network behaves when the boilers, pumps or substations are under-sized. The model is used to compute thermal losses and pumping power, and concludes that adding distributed solar thermal injection, slightly decreases thermal losses but can quite significantly increase the pumping power required.

Tian [50] implements a detailed model for large fields of solar thermal collectors. He validates the model using measured data from a real plant. The study doesn't concentrate of the actual district pipeline network, which is considered as a simple heat load.

Renaldi [40] simulates a district network on TRNSYS. Weather and solar irradiance are taken from real meteorological measurements. The thermal station is modelled with more detail, but the rest of the district network is simplified to a simple heat load. It only works for centralised solar production, not decentralised. There is a quite complete and customisable borehole heat storage, to study thermo-economical aspects of the network. Synthetic heat demand profile models are used [18], which gives only a yearly value of consumption. Then to get hourly values, it is set to be proportional to the temperature difference between the outdoor air and desired indoor temperature. The focus is on economical feasibility.

Abokersh [1] uses a TRNSYS model from [52]. Similarly, only the thermal station is detailed, and the rest of the district network is just a heat load. It simulates only central solar heating plants coupled with seasonal storage (CSHPSS). Results of simulations with different scenarios are used to train machine learning algorithm. The goal is to optimize the sizing and other thermo-economical parameters.

Oppelt [36] is an example of a very complete simulation of district energy system (particularly for district cooling). It can make many complex topologies. However the cold demand is not modelled in detail. No implementation of decentralized injections of cold or hot are considered, there is only one central producer.

It has also been found that operating decentralized solar injections is still ongoing research, because it is hydraulically a quite complex and unstable system to converge [30].

### 1.3 Problematic

When building a new district heating network, it is important to get the correct sizing to improve performances. Indeed, boilers, substations, pipes, solar collectors have a range of operation under which the efficiency is higher. This is especially true for solar collectors and its storage.

There is therefore a need for a software tool that can estimate the energy consumption of buildings, simulate a district heating network of complex topologies and the solar thermal panels. The solar power injections play an important role, and the power produced during summer can be used in winter using seasonal storage. This means simulations over long periods of time, for a whole year, are important to integrate properly the seasonal storage. For the solar, it is also important to have sufficiently small time steps, of the order of one hour, to capture the dynamics. This will help new projects to take place, with as precise information as possible on the energy that will be consumed, the amount of losses, the solar power produced and when that power is produced. This helps to make the desired thermo-economical optimisations, crucial for the network constructors.

This is the tool that this thesis aims to develop, to meet the described demand. To do so, the CitySim software has been identified as a basis to work on. It is already quite complete with respect to building thermal simulations, and on solar thermal panel modelling. It also already has a simplified district heating network model. This thesis aims to build on this software, to implement a global tool that can simulate efficiently district heating networks with complex topologies, powered by low-fossil-carbon energy sources.

The software will be free and open source, in a hope to promote this kind of technology.

### 1.4 Thesis content

Chapter 2 presents the general features of the CitySim code as it was originally, before the work from this Thesis has added modifications. This includes thermal models of buildings, solar thermal panel models and a code with simplified hydraulics to simulate district heating networks by a former PhD candidate Dapeng Li. However, his work was never published as the data was missing for the validation of the dynamics of the model.

Chapter 3 presents all the contributions of this Thesis. It describes the new models and algorithms implemented. It starts with the incorporation and improvements of a district heating simulator to CitySim. This first simulation only works for networks with a "branched" topology and one central thermal station. Then a new formalism is introduced to simulate "looped" topologies with multiple possible thermal stations. Finally is introduced the model to inject solar thermal power into the network.

Chapter 4 presents results and discussions of simulations using the tools developed. To validate the model and verify the code, simulations are compared to real data. A comparison with the "branched" Broc network is made. However due to time restrictions the comparison with a looped network has not been finished for this written thesis, but will be presented in the oral examination. The solar thermal injection has no data to be compared to, but have been tested a toy network. Remarks are made on the control strategies and integration of solar to district networks.

Chapter 5 summarizes the thesis, its conclusions and proposes further works to explore.

## 2 Original status of CitySim

The CitySim software is a numerical tool to simulate urban physics. It aims to help urban energy planners and stakeholders make decisions to minimise greenhouse gas emissions and consumption of non renewable energy sources. The software was originally developed by the EPFL Home Solar Energy and Building Physics Laboratory LESO-PB [14]. A graphical user interface is distributed for free by kaemco [25].

The program takes as input a file that contains information on a whole district or city. It accepts 3D shapes of the buildings, information on the different zones inside ; the insulation materials of walls and roofs ; the types of occupants and their, possibly stochastic, activity types such as electric and water consumption ; ground properties ; trees ; and more. The 3D shapes can be imported from multiple sources, and can be transformed into the dedicated XML format.

The program can then simulate the district. It computes thermal exchanges, heating systems of buildings, solar irradiance, shadows, window reflections, etc, to get properties such as heat demand, temperatures and solar power production, at each time step. The default time step is one hour, and simulations can last a whole year.

The results can be exported in standard text files (tab separated values), to be analysed.

The C++ code behind this software can be approximate to over 40'000 lines of instructions dispatched over 40 files.

### 2.1 Building thermal simulation

Each building's thermal properties can be simulated by CitySim. The software takes into account wide range of affects. Based on the climate files, it can model the solar irradiance heating the building, the interaction and consumptions, window opening of the occupants, the thermal losses through the building envelop to the environment, and more. Each building can have parametered to specify the desired minimum and maximum temperature accepted by the occupants. The evolution of the system is evaluated as well as the demand for space heating and space cooling. This demand is then communicated to the building's energy system. A range of standard energy systems are available such as boilers, heat pumps, etc.

The energy system has the possibility to include thermal storage tanks. In real life, such tanks are used as buffers. They make it possible to store energy when the demand is low, to restore that power in addition to the heat producer when the demand is very high. They are also

useful to avoid turning the energy system on and off repeatedly at low power. They are usually tanks of water that can be heated or cooled. In CitySim, a building can have a tank for the space heating (hot storage HS), one for domestic hot water (DHW) and one for space cooling (cold storage CS).

The volume and operating temperatures of these tanks can be specified. For example, the HS tank can have lower limit of 25°C and an upper limit of 35°C. When the thermal energy is required to heat the building, it takes the energy from that tank. As the tank is used, its temperature decreases, for example by going through the radiators of the building. When its temperature goes below the lower limit, of 25°C in the present case, the algorithm asks the heating unit the power to heat it back up the upper limit, of 35°C in our case.

Solar thermal panels can be linked to the building. To simulate their behaviour, the solar irradiance reaching it is computed. The efficiency of the collector then depends on the outside air temperature  $T_a$  [°C] and the mean temperature of the fluid being heated in the collector  $T_{fl}$  [°C]. Indeed the temperature difference between the fluid in the collector and the outside air is too big, heat is exchanged and lost to the environment, as the insulation of the collector is imperfect. Usually if the heat transfer fluid enters the collector at  $T_{in}$  [°C] is heated, and exits at  $T_{out}$  [°C], then the average fluid temperature is taken as  $T_{fl} = \frac{1}{2}(T_{in} + T_{out})$ . The efficiency  $\eta$  [-] is therefore

$$\eta(T_{fl}, T_a, G) = \eta_0 - a_1 \frac{T_{fl} - T_a}{G} - a_2 \frac{(T_{fl} - T_a)^2}{G} \quad (2.1)$$

with  $G$  the irradiance on the surface [W/m<sup>2</sup>]. The parameters  $\eta_0$  [-],  $a_1 > 0$  [W/(K·m<sup>2</sup>)] and  $a_2 > 0$  [W/(K·m<sup>2</sup>)] are usually given by the panel constructor [26], and have to be given in the input file. The parameters  $a_1$  and  $a_2$  are positive as the solar collector's efficiency decreases as  $T_{fl} - T_a$  increases. The total solar power available  $\dot{Q}_{sol}$  [W] is

$$\dot{Q}_{sol} = \eta AG \quad (2.2)$$

where  $A$  [m<sup>2</sup>] is the surface area of the collector.

Once the solar thermal power is computed, it is distributed among the storage tanks. For each tank, power that is still needed is asked to the heating and cooling units. After the heating and cooling units compute how much power they can effectively provide. With this the final temperature is computed inside the building's zones. If the units could not provide enough power, the temperature in the building goes below the temperature required by the user. At each time step the consumptions and temperatures of tanks are recorded. At the end of the simulation, these values can be exported to an output file.

## 2.2 District heating simulation

CitySim originally had a simplified model for district heating and cooling simulation. This work was performed by the former PhD candidate Dapeng Li. Because no data on real district

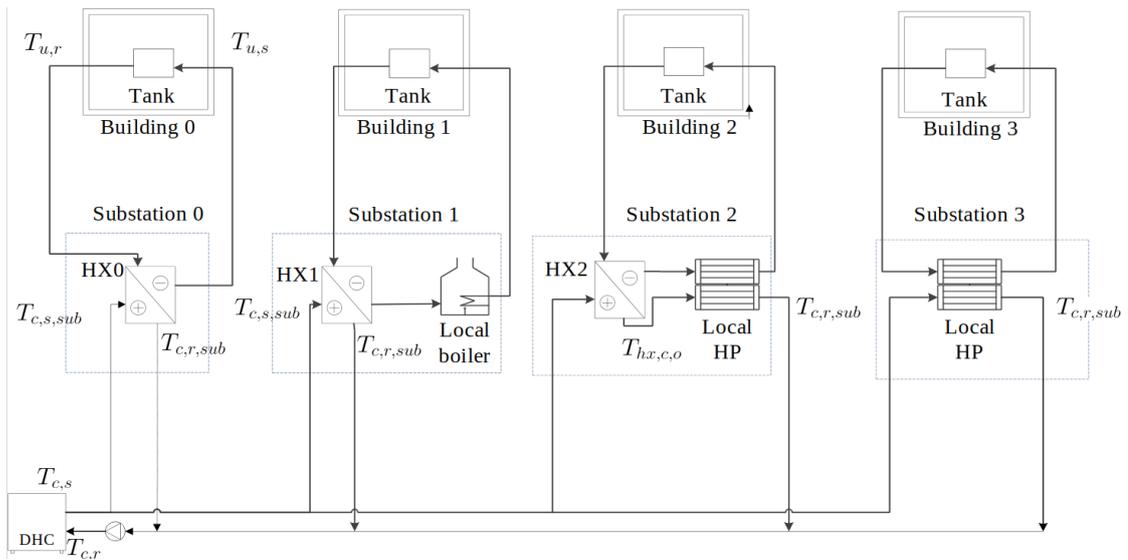


Figure 2.1 – District network schematics by Dapeng Li with different types of substations and a thermal station of his original model.

heating plants were available, the model and code had not been validated on real data. It must be noted that the rest of the CitySim code had evolved, making this part of the code not compatible any more. Some work was necessary to make the code function again.

The code could simulate networks with one thermal station, and as many substations as desired. The network had to be tree-shaped, and the hydraulics were very simplified. The central thermal station produces the heat or cold. The substations are each associated to a building that consumes the heat or cold. The primary circuit is the set of pipes that connects the central station to the substations. The secondary circuits connect the substations to their respective building's water tanks. At the substation heat is exchanged between the primary and secondary circuits.

### 2.2.1 Substation

As shown in Figure 2.1, 4 types of substations could be simulated :

- simple substations with only a heat exchanger.
- substations with a heat exchanger and an additional local boiler to complement the heating.
- substations with a heat exchanger and an heat pump exchanging heat between the primary and secondary networks to complement the heating.
- substations with only heat pumps between the primary and secondary networks

The substation manages the heat exchange between a district energy center's pipe network, called primary network, and a building's pipe network, called secondary network. The sec-

ondary network then exchanges heat with tanks (e.g. heat storage, cold storage or domestic hot water tanks). The parameters needed to create a substation are:

- Design thermal power  $\dot{Q}_d$  [W]. It is the maximal power that can be exchanged between the primary and secondary network. It is defined as positive for both heating and cooling substations.
- Design temperature difference  $\Delta T_d$  [K]. It is the temperature difference, in "design condition", between the fluid entering and leaving the substation on the primary side. It is defined as positive for both heating and cooling substations. The substation will try to produce the  $\Delta T_d$  temperature difference even when working at less than design power.
- Design effectiveness  $\epsilon_d$  [-]. Its range is  $0 < \epsilon_d \leq 1$ . As formulas will clarify, it quantifies the effectiveness of the heat exchanger at design condition.

The following explains the computations for a heating substation. The cooling substation case is very similar.

Let  $\dot{Q}_{needs}$  [W] be the power needs from the building that the substation will try to supply. Let  $T_{s,in}$  be the temperature of the fluid entering the heat exchanger on the secondary network side. The model takes  $T_{s,in}$  as the temperature at the end of the previous time step, of the hot storage tank to be heated. Let  $T_{p,in}$  be the temperature of the fluid entering the heat exchanger on the primary network side. The model takes  $T_{p,in}$  as the converged temperature in the pipe that the substation is connected to. The design mass flow is  $\dot{m}_d = \frac{\dot{Q}_d}{c_p \Delta T_d}$  [kg/s], where  $c_p$  [ $\frac{J}{kg \cdot K}$ ] is the specific heat capacity of the primary network fluid.

The mass flow on the secondary side is computed as

$$\dot{m}_s = \dot{m}_d \cdot \max \left[ 0.1, \min \left[ 1, \frac{\dot{Q}_{needs}}{\dot{Q}_d} \right] \right] \quad (2.3)$$

More simply explained, it grows linearly with the needs, but cannot exceed the design mass flow  $\dot{m}_d$ , nor be smaller than  $0.1 \dot{m}_d$ . The mass flow on the primary side is determined by formula

$$\dot{m}_p = \begin{cases} \dot{m}_d & \text{if } 0.75 \leq \dot{Q}_{needs} / \dot{Q}_d \\ 0.75 \dot{m}_d & \text{if } 0.5 \leq \dot{Q}_{needs} / \dot{Q}_d < 0.75 \\ 0.5 \dot{m}_d & \text{if } 0.25 \leq \dot{Q}_{needs} / \dot{Q}_d < 0.5 \\ 0.25 \dot{m}_d & \text{if } \dot{Q}_{needs} / \dot{Q}_d < 0.25 \end{cases} \quad (2.4)$$

The granularity in these values, with only 4 possible settings can help make the network more

stable, since less variations in the mass flow demand occurs. Then define

$$C_{min} = \min[c_p \dot{m}_s, c_p \dot{m}_p] \quad (2.5)$$

$$C_{max} = \max[c_p \dot{m}_s, c_p \dot{m}_p] \quad (2.6)$$

$$C^* = C_{min}/C_{max} \quad (2.7)$$

$$(UA)_d = \frac{\dot{Q}_d}{\Delta T_d} \frac{\epsilon_d}{1 - \epsilon_d} \quad (2.8)$$

The heat transfer equation between the primary  $p$  and secondary  $s$  sides of a heat exchanger is

$$UA = A \left[ \frac{1}{h_p} + \frac{1}{h_s} \right]^{-1}, \quad (2.9)$$

with  $A$  [m<sup>2</sup>] the exchange area,  $U$  [W/(m<sup>2</sup>·K)] the overall heat transfer coefficient and the convective heat transfer coefficients  $h$  [W/(m<sup>2</sup>·K)]. It is approximated that the effect of the finite heat exchange wall is negligible. Under the design condition,  $h_{p,d} = h_{s,d} = h_d$  since the flows on the primary and secondary side are the same. This means that  $(UA)_d = \left[ \frac{2}{Ah_d} \right]^{-1}$ . In the counter-flow heat exchanger, the convective heat transfer coefficient [46] can be simplified to

$$h = Z_1 \dot{m}^{0.583}, \quad (2.10)$$

where  $Z_1$  is a constant. This therefore leads to

$$UA = (UA)_d \frac{2}{\left(\frac{\dot{m}_d}{\dot{m}_p}\right)^{0.583} + \left(\frac{\dot{m}_d}{\dot{m}_s}\right)^{0.583}} \quad (2.11)$$

From this, one can compute

$$NTU = \frac{UA}{C_{min}} \quad (2.12)$$

Finally, this leads to the computation of the desired variable, the effectiveness

$$\epsilon = \begin{cases} \frac{1 - \exp[-NTU(1 - C^*)]}{1 - C^* \exp[-NTU(1 - C^*)]}, & \text{if } C^* \neq 1, \\ \frac{NTU}{1 + NTU}, & \text{if } C^* = 1. \end{cases} \quad (2.13)$$

This effectiveness describes what portion of the maximal theoretical heat exchange  $\dot{Q}_{max,theo}$  is in fact exchanged. In a counter-current heat exchanger where the primary flow enters hot at

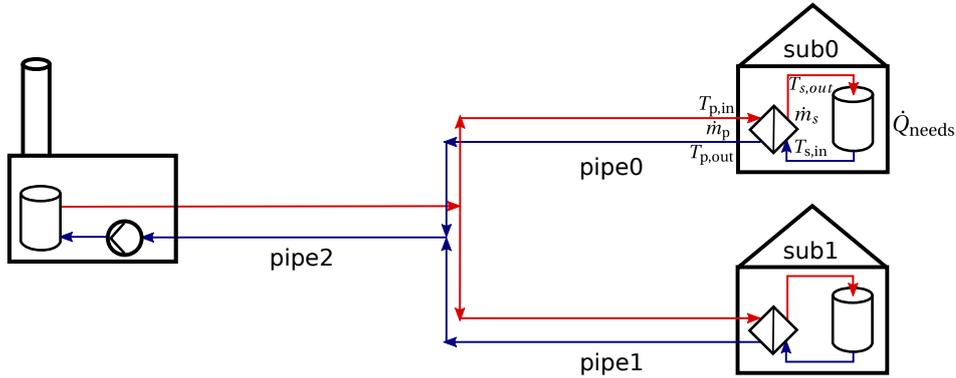


Figure 2.2 – District network schematics.

$T_{p,in}$  and the secondary flow to be heated enters the heat exchanger at cold  $T_{s,in}$ , then :

$$\dot{Q}_{\max, \text{theo}} = C_{\min}(T_{p,in} - T_{s,in}). \quad (2.14)$$

This can be shown because otherwise either the secondary output temperature will be higher than the primary input  $T_{s,out} > T_{p,in}$ , or the vice versa  $T_{p,out} < T_{s,in}$ . Either of these is not possible otherwise it would mean that heat has flown from a cold fluid to a warm fluid, which is forbidden by the second law of thermodynamics. Therefore, for substation types with a heat exchanger, the heat exchanged in that component is

$$\dot{Q}_{\text{HX}} = \varepsilon \dot{Q}_{\max, \text{theo}} = \varepsilon C_{\min}(T_{p,in} - T_{p,out}) \quad (2.15)$$

For substations with an additional boiler or heat pump, if this heat exchanged is not enough to meet the heating needs of the building  $\dot{Q}_{\text{needs}}$ , the CitySim models for the boiler or heat pump is called to get the additional heat left  $\dot{Q}_{\text{needs}} - \dot{Q}_{\text{HX}}$ .

For substation types that only have a heat pump, the heat pump model of CitySim is called using temperature  $T_{p,in}$  for the cold source of the heat exchanger.

However the substations had not yet implemented a way to inform the building whether or not sufficient heat had been extracted from the network, meaning that the heat demands of buildings were always satisfied.

### 2.2.2 Network

The network had one central thermal station and a pipeline network that could connect as many substations as desired. As shown in Figure 2.3, the pipelines represent as a graph which must be tree-like. No loops are allowed. Each pipe segment is oriented, the inward direction leads towards the central thermal station. The outward direction leads to a substation or multiple outward pipes if it is a junction that splits the pipe.

To compute the mass flows and temperatures through the network, it is supposed that each

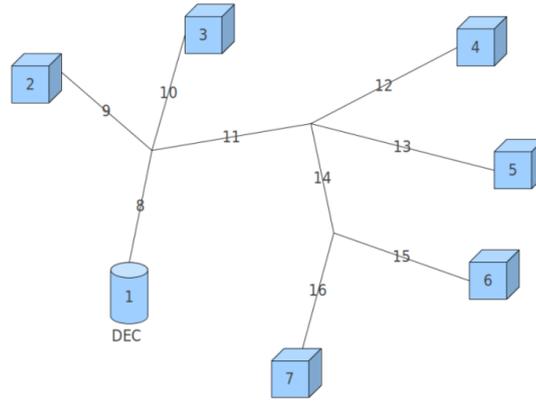


Figure 2.3 – Pipeline network schematics by Dapeng Li with central thermal station (DEC) as a cylinder and substations as cubes. In this original model, the network had to be a tree shape. Each pipe had an inward direction, which points towards the central thermal station and connects to the previous pipe. The outward direction leads to either a substation or multiple outward pipes, if the junction splits the pipe.

substation manages to impose its desired mass flow  $\dot{m}_{des}$ , and the temperature exiting the substation  $T_{p,out}$  is the input temperature of the previous step  $T_{p,in,prev}$ , minus the temperature drop due to the heat exchanged.

$$\dot{m}_{\text{substation pipe}} = \dot{m}_{des} \quad (2.16)$$

$$T_{p,out} = T_{p,in,prev} - \frac{\dot{Q}_{needs}}{\dot{m}_{des} c_p} \quad (2.17)$$

Then the mass flows are propagated inwards towards the central thermal station. At each junction, the flow through the inward pipe is computed as the sum of the outward pipes, in order to conserve mass flow. Similarly by conservation of energy, the temperature of the inward pipe is the weighted average of temperatures in the outward pipes.

$$\dot{m}_{\text{inward pipe}} = \sum_{i \in \text{outward pipes}} \dot{m}_i \quad (2.18)$$

$$\dot{T}_{\text{inward pipe}} = \frac{\sum_{i \in \text{outward pipes}} T_i \dot{m}_i}{\sum_{i \in \text{outward pipes}} \dot{m}_i} \quad (2.19)$$

For example in Figure 2.3, if all substations want to impose 1 kg/s, then pipes 9, 10, 12, 13, 15 and 16 are computed first with a mass flow of 1 kg/s, then pipe 14 is computed with 2 kg/s,

then pipe 11 with 4 kg/s then pipe 8 with 6 kg/s. Computations are done in a similar way for the temperatures.

The temperature is supposed uniform in the pipe, without thermal losses. The temperature drop due to thermal losses in return pipes are only added at the end, just before the fluid flows into the central thermal station. The total loss  $\dot{Q}_{\text{loss}}$  [W] is

$$\dot{Q}_{\text{loss}} = \sum_{i \in \text{return pipes}} \frac{T_i - T_{\text{ext}}}{R_i} L_i \quad (2.20)$$

with  $T_i$  [°C] the temperature in the pipe,  $T_{\text{ext}}$  [°C] the exterior soil temperature,  $L_i$  [m] the length of the pipe and  $R_i$  [K·m/W] the sectional resistance of the pipes. When transmitting the temperature through the supply pipes, there are no thermal losses, and flows only divide, they do not combine, so the temperature in all supply pipes is the same as the temperature supplied by the central thermal station. The temperature change due to the thermal losses in supply pipes is only added at the end, just before the fluid enters the substation. The thermal loss for the fluid entering a given substation  $s$  is

$$\dot{Q}_{\text{loss,subs}} = \sum_{i \in \text{pipes in path from central station to subs}} \frac{\dot{m}_{\text{subs}}}{\dot{m}_{\text{pipe}i}} \frac{T_i - T_{\text{ext}}}{R_i} L_i. \quad (2.21)$$

The idea is that for each pipe in the path from the central station to the substation, only add the fraction of the thermal loss that is the fraction of the mass flow from that pipe that will go to the substation. To converge to the real losses, a loop is implemented, propagating through the return pipes, then the central thermal station, then the supply pipes, then the substations and so on, until the thermal losses are change by less then 10W from the previous iteration.

### 2.2.3 Central thermal station

The central thermal station heats up the fluid to a fixed setpoint supply temperature. However, the heating unit has a nominal power, and may not be powerful enough to heat the fluid all the way to the setpoint temperature. In which case the fluid is only partially heated.

To compute the electric power consumed by the pump, the pressure difference around the thermal station is estimated. To do so the "furthest" substation  $s^*$  is identified. To do so, for each substation count the number of pipes in the path from the substation to the central thermal station. The "furthest" substation is the one whose path has the greatest number of pipes. (It would probably have been better to add the lengths of the pipes). Then pressure drop around the pump is

$$\Delta p = \sum_{i \in \text{pipes in path from central station to } s^*} 6.819 \left( \frac{V_i}{C} \right)^{1.852} \left( \frac{1}{D_i} \right)^{1.167} \rho g L_i, \quad (2.22)$$

with  $C$  the roughness factor of the pipes,  $D_i$  [m] the diameter of the pipe,  $V_i$  [m/s] the average velocity of fluid through the pipe,  $\rho$  [kg/m<sup>3</sup>] the density of the fluid,  $g = 9.81$  m/s<sup>2</sup> the gravita-

tional acceleration. This formula comes from [4]. The required electric pump power  $P_{\text{pump}}$  [W] is then defined as

$$P_{\text{pump}} = \frac{\dot{m}_{\text{ts}} \Delta p (1 + R)}{\rho \eta_{\text{pump}}}, \quad (2.23)$$

with  $\dot{m}_{\text{ts}}$  [kg/s] the mass flow through the thermal station,  $R$  [-] is described as "the ratio of local loss to frictional one" (a parameter of the pump which is chosen by the user) and  $\eta_{\text{pump}}$  [-] the overall pump efficiency.



## 3 Methodology

This section describes the additions and modifications that have been implemented during this thesis. It starts with the implementation of the code to simulate district networks with branched topologies. Then it explains the implementations for more complex looped topologies with multiple heat sources. Finally it describes how centralized or decentralized solar thermal power is handled.

### 3.1 Branched code

The basis used for this first step is the original code from Dapeng Li described in section 2.2.

#### 3.1.1 Informing effectively provided power

The original code of CitySim has been built for heating and cooling systems in buildings that work independently of other buildings. Therefore there was a global loop over all buildings. One at a time, each building would determine its thermal needs, ask the heating and cooling units the required power. The stand-alone unit could respond the power it could provided. Then the new temperature of the building could be computed. After that, the algorithm proceeded to do the same computations for the next building in the array.

However, a substation is part of a district heating network, linked to the other buildings. It is not a stand-alone thermal system. One must simulate the network to determine how much heat is available for each substation. But to simulate the network, one first needs to know all consumption needs of all substations.

This led to a reordering of the computations. The code has been modified to first determine and store all thermal needs of buildings. Then all the heating and cooling units of the buildings are simulated. This includes the district heating networks. This determines the power that substations can provide to the buildings. Only after that the buildings determine the evolution of their temperature.

#### 3.1.2 Substations

The basics of the substation model has stayed the same as the original code from Dapeng Li, discussed in section 2.2.1.

As mentioned, there was an effort to make that code compatible with the rest of the CitySim program. This required a strong reorganisation of the code. In that process, only the most

common type of substation has been kept, the one that has a heat exchanger. The others should be implementable quite easily if one wants to add them back.

There has been added verifications to make the code more robust. Including a verification that the thermal power extracted is no more than the thermal power needed by the building. Also there are verifications that the temperatures are compatible for the heat exchange, so that power is not transmitted the wrong direction by the substation.

### 3.1.3 Thermal station

The basics of the thermal station model has stayed the same as the original code from Dapeng Li, discussed in section 2.2.3.

There has been the addition of multiple temperature setpoint possibilities. There is still the constant temperature setpoint, which stays the same all year long. There are also "affine" setpoints, in which the temperature supplied to the network depends on the exterior temperature average over the last 24 hours  $T_{\text{ext},24\text{h}}$ . Any linear by parts function can be implemented, so to some extent any function by adding sufficiently many points. This is commonly used in district networks. There is also the possibility to switch between a constant temperature mode in summer and an affine mode in winter. Many additional setpoint strategies can be easily implemented.

### 3.1.4 Network

#### Hydraulics

The hydraulics of the pipeline system of the district network stays similar to the original described in section 2.2.2. The substations have a desired mass flow based on the thermal power needed. This mass flow is imposed at the substation. The flows are then propagated to inward pipes, closer to the central thermal station. At each junction, the outward flows are summed to get the inward flow. The pump at the thermal station is supposed powerful enough to provide the hydraulic power necessary.

Overall there has been a reorganisation of the data structures in the code. More efficient algorithms based on graph theory have been implemented and a better use of C++ class hierarchy. Class entities such as `Pipes` and `network Nodes` have been created.

#### Pipe thermal model

A new model for the thermal behaviour in pipes has been implemented. The pipe has a flow through it, coming from an upstream node. This node can be a pipe junction or a substation or a thermal station. The upstream end of the pipe is considered as the  $x = 0$  location, and has the temperature  $T(x = 0)$  [°C] given by the upstream node. As the fluid travels through the pipes, the model considers three phenomena that can add or remove heat :

- Heat exchange with the exterior soil surface  $\dot{Q}_{\text{ext}}$  [W]
- Heat exchange with the parallel twin pipe  $\dot{Q}_{\text{twin}}$  [W]. The twin pipe of a supply pipe is its return pipe neighbour, and vice versa.

- The dissipation of hydraulic energy into heat as the flow loses pressure through friction with the pipe walls  $\dot{Q}_{\text{diss}}$ . In the branched-code, no friction is considered so this term can be set to zero, but it will be used in the looped-code.

When heat is exchanged through conduction between an object at  $T_1$  [°C] and an object at  $T_2$  [°C], the heat flow  $\dot{Q}$  [W] from object 1 to object 2, can usually be written in the form  $R\dot{Q} = (T_1 - T_2)$ , where  $R$  [K/W] is the overall thermal resistance corresponding to the geometric configuration between the two objects.

Figure 3.1 shows the case as the water inside pipe exchanging heat across the insulation layer, with resistance  $R_{\text{insul}}$ . Figure 3.2 shows the case of the outside of the insulation layer of the pipe exchanging heat with the soil surface  $R_{\text{surf}}$ . Figure 3.3 shows the case of two parallel pipes exchanging heat across the soil, with resistance  $R_{\text{parallel}}$ . In general, when multiple thermal resistances are put one after the other, in series, then the resistances add up. In the case of  $\dot{Q}_{\text{ext}}$ , the heat has to go through the insulation layer, and then to the surface, so  $R_{\text{ext}} = R_{\text{insul}} + R_{\text{surf}}$ . In the case of  $\dot{Q}_{\text{twin}}$ , the heat has to go through the insulation layer of the first pipe then through the soil to the other pipe, then through the insulation layer of the second pipe, so  $R_{\text{twin}} = R_{\text{insul},1} + R_{\text{parallel}} + R_{\text{insul},2}$ .

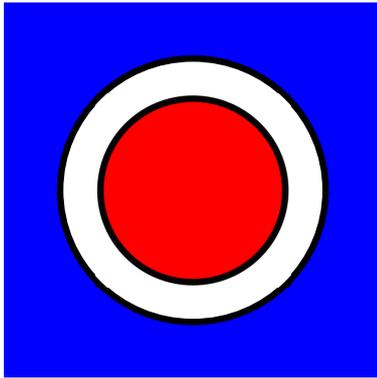


Figure 3.1 – Section view of a red inner cylinder at temperature  $T_1$  [°C] of length  $L$  [m] and radius  $r$  [m] which is inside a blue exterior holed object which is at temperature  $T_2$  [°C]. The white material between the two objects has conductivity  $\lambda$  [W/(K·m)] and a thickness  $t$  [m]. Under the condition that  $L \gg d$ , this geometry corresponds to an equivalent resistance of  $R = \frac{\ln[(r+t)/r]}{2\pi L\lambda}$ .

This total resistance of a heat exchange can be turned into a total conductance  $K = 1/R$  [W/K], which is then turned into a linear conductivity  $\lambda = K/L = 1/(L \cdot R)$  [W/(K·m)] by dividing by the pipe length  $L$  [m]. This linear conductivity is useful to describe the heat exchange per unit length of pipe. For the term due dissipation, a pressure loss of  $\Delta p$  [Pa] through the pipe of length  $L$  at a flow of  $\dot{V} = \frac{\dot{m}}{\rho}$  [m<sup>3</sup>/s] is a loss of mechanical power  $\dot{W} = \dot{V}\Delta p$  [W], which is transformed into heating power  $\dot{Q}_{\text{diss}} = \dot{W} = \dot{V}\Delta p$  [W]. This heat source per unit length of pipe is therefore  $\frac{\dot{Q}_{\text{diss}}}{L} = \frac{\dot{m}\Delta p}{\rho L}$  [W/m].

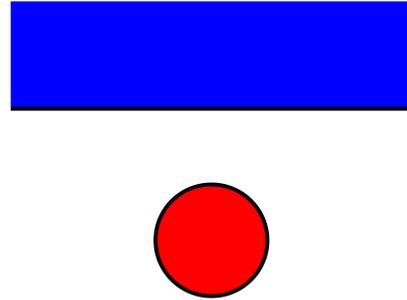


Figure 3.2 – Section view of a red cylinder at temperature  $T_1$  [°C] of length  $L$  [m] and diameter  $d$  [m] whose centre is at depth  $z$  [m] beneath the infinite plate delimiting the blue object which is at temperature  $T_2$  [°C]. The white material between the two objects has conductivity  $\lambda$  [W/(K·m)]. Under the condition that  $L \gg d$ , this geometry corresponds to an equivalent resistance of  $R = \frac{\text{arcosh}(2z/d)}{2\pi L\lambda}$  if  $z \leq 1.5d$ , and  $R = \frac{\ln(4z/d)}{2\pi L\lambda}$  otherwise.



Figure 3.3 – Section view of a red cylinder at temperature  $T_1$  [°C] and a blue cylinder at temperature  $T_2$  [°C] of length  $L$  [m] and diameter respectively  $d_1$  [m] and  $d_2$  [m]. Their centres are separated by  $B$  [m]. The white material between the two objects has conductivity  $\lambda$  [W/(K·m)]. Under the condition that  $L \gg d$ , this geometry corresponds to an equivalent resistance of  $R = \frac{\text{arcosh}[(4B^2 - d_1^2 - d_2^2)/(2d_1d_2)]}{2\pi L\lambda}$ .

During the simulation, the temperature of fluid in the twin pipe at  $x = 0$  and at  $x = L$  are known, so in the following computation, the twin pipe temperature is approximated to an affine function  $T_{\text{twin}}(x) = T_{\text{twin},0} + (T_{\text{twin},L} - T_{\text{twin},0})\frac{x}{L} = Ax + B$ .

At a given position  $x$  [m] of the pipe, over an infinitesimal distance  $dx$  [m], the sum of the heat sources and sinks make the temperature evolve :

$$\begin{aligned} \dot{m}c_p[T(x) - T(x + dx)] &= d\dot{Q}_{\text{ext}} + d\dot{Q}_{\text{twin}} + d\dot{Q}_{\text{diss}} \\ -\dot{m}c_p dT(x) &= \lambda_{\text{ext}}[T(x) - T_{\text{ext}}]dx + \lambda_{\text{twin}}[T(x) - T_{\text{twin}}(x)]dx - \frac{\dot{m}\Delta p}{\rho L}dx \\ \frac{dT(x)}{dx} &= \frac{-1}{\dot{m}c_p} \left\{ \lambda_{\text{ext}}[T(x) - T_{\text{ext}}] + \lambda_{\text{twin}}[T(x) - (Ax + B)] - \frac{\dot{m}\Delta p}{\rho L} \right\}. \end{aligned} \quad (3.1)$$

This gives rise to a differential equation of the form  $T'(x) = aT(x) + b + cx$ . The coefficients are  $a = -\frac{\lambda_{\text{ext}} + \lambda_{\text{twin}}}{\dot{m}c_p}$ ,  $b = \frac{\lambda_{\text{ext}}T_{\text{ext}} + \lambda_{\text{twin}}T_{\text{twin},0}}{\dot{m}c_p} + \frac{\Delta p}{c_p\rho L}$  and  $c = \frac{\lambda_{\text{twin}}(T_{\text{twin},L} - T_{\text{twin},0})}{\dot{m}c_p L}$ . This has an explicit solution  $T(x) = -\frac{c+a(b+cx)}{a^2} + Ke^{ax}$ . The integration constant  $K$  can be determined by imposing the initial condition  $T(x=0) = T_0$  which gives  $K = T_0 + \frac{c+ab}{a^2}$ .

We are interested in the downstream temperature of the fluid at  $T(x=L)$ , right before it enters the next node. The thermal loss of the fluid over the pipe is therefore  $\dot{Q}_{\text{loss}} = \dot{m}c_p[T_0 - T(L)]$ . One can notice that this thermal loss can be negative, if the fluid actually gains heat, either because it is colder than the exterior, or the twin pipe, or the pressure loss is sufficiently high.

### Thermal convergence

The thermal convergence is used to determine the temperatures throughout the network components.

As explained in section 3.1.4, for each pipe, when if the upstream temperature is given as input  $T(x=0) = T_0$ , the downstream temperature  $T(x=L)$  can be computed. This is the temperature of the fluid, right before reaching the next node. Similar can be done for the substations, if the temperature entering it is known, and the mass flow through it, the substation can compute the heat exchanged and therefore the downstream temperature. Similarly for the thermal

station.

To compute the temperature at a node, identify all pipes which flow to that node  $n$  and do a weighted average of their downstream temperatures

$$T_{\text{noden}} = \frac{\sum_{p \in \text{pipes with flow towards node } n} T_{\text{downstream}, p} \dot{m}_p}{\sum_{p \in \text{pipes with flow towards node } n} \dot{m}_p} \quad (3.2)$$

This node temperature becomes the "upstream" temperature of components (pipes, substations of thermal station) connected to that node, where the flow goes from the node to that component.

To get the converged temperatures in all components of the network :

- Suppose that the upstream temperatures of the substations are the same as those from the previous time step.
- Make the substations compute the downstream temperature.
- Propagate the temperature from the substation return pipes through the return network, to the thermal station return pipe.
- Make the thermal station compute the downstream temperature that it injects at the supply pipe.
- Propagate the temperature from the thermal station supply pipe through the supply network, to the substation supply pipes.
- The convergence criteria is that all upstream substation temperature have changed by less than  $\epsilon$  [ $^{\circ}\text{C}$ ]. If it is not met, repeat from the second bullet point.

Unless mentioned otherwise, this parameter is set at  $\epsilon = 0.2^{\circ}\text{C}$ .

## 3.2 Looped code

The next task in this thesis is to improve the simulation, by adding new features and possibilities in the networks that can be represented and simulated. Notably among these new specifications is the possibility to model looped (or meshed) topologies, with multiple thermal stations.

### 3.2.1 Looped topology

Historically district heating or cooling networks have been built in tree shapes [55, 58]. The tree shape can be mathematically described as so. Consider either the network of supply pipes, or of return pipes. Both are usually the same in most district networks since the supply and return pipes are built parallel to one another. Make a mathematical graph to describe it, where edges are the pipes, and nodes are either pipe intersections, the thermal station, or the substations. Then the graph is a tree, and substations are leaves in that tree as shown in Figure 3.4.

The complexity of networks can then increase if multiple thermal power plants are connected in a tree-like graph. This can for example happen if two independent tree-shaped networks are joined together. The city network goes from multiple islands to one large coherent tree, containing multiple thermal power plants.

Another possibility is that a network's graph is not mathematically a tree, but has cycles, which we'll call loops or meshes, as shown in Figure 3.5. This can happen by starting from a branched network. Over time, the network increases as more substations connect sequentially to the network. Each time a new substation joins the network, it connects to the nearest pipe of the network, but after a while, the path to a substation may be much longer than the direct distance to the thermal station, as the crow flies. This may be inefficient as the hydraulic pump needs to operate at high power and consume a lot of electricity, but the substation still has low differential pressure between the supply and return. A way to fix this is to add a pipe that short cuts the long snaking path, by creating a loop in the process.

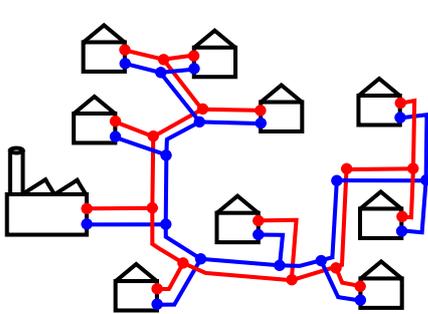


Figure 3.4 – Schematics of a district network with a branched topology.

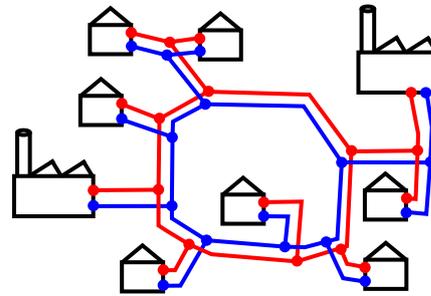


Figure 3.5 – Schematics of a district network with a looped topology and two thermal stations.

These networks have advantages. They can have multiple paths between a given substation and any thermal power plant. This redundancy means that if one of the power plant or one of the pipes breaks, or undergoes maintenance, the substations can still be supplied in heat. This makes the network more resilient, by permitting regular inspections and repairs on different network components.

These networks also have disadvantages. The complexity brought by multiple paths between thermal power plants and substations means that sometimes mass flow can reverse or stagnate in some pipes, depending on how much flow certain consumers are requesting and how much pressure each thermal power station is producing. Computations are more complicated because there are non-linearities of thermal and hydraulic coupling, and more degrees of freedom. Therefore more thorough simulations to see how the network responds to stimuli. These simulations are needed to analyse and compare operation strategies, in order to find the optimal. They are also important for the sizing of the network and its different components.

### 3.2.2 Pressure

To simulate networks with loops and with multiple thermal power plants, a whole new mathematical formalism is necessary. The major novelty is that mass flow direction is not known within a pipe. In the branched simulations, the substations at the leaves of the tree-shaped

graph imposed there mass flow. Then as we moved towards the thermal power plant, at each intersection, one could simply add the mass flows of the pipes converging into one "inward" pipe. Now at each intersection, we do not know in which pipes are flowing to the intersection, and which ones are flowing away from the intersection.

The new element that must come into play is hydraulic pressure in the water, measured in Pascals. The overall idea is that the thermal power plants use pumps, to impose a pressure difference between the supply and return pipes. In the pipes, the water flows from high to low pressure. In each pipe there is some pressure loss due to the friction at the pipe walls. Substations can control the mass flow that they receive by choosing the opening position of a valve, which obstructs more or less the flow, to get the pressure drop corresponding to the desired flow.

It is very similar to an electric network, the pressure corresponds to the electric voltage and the mass flow to the electric current. The equations that apply to determine the system are the Kirchhoff laws and the specific laws governing the different components.

### 3.2.3 Graph

There are now many more degrees of freedom in the possibilities to build the district thermal network, so it will be represented in a much more flexible way. To do so this work uses graph theory, in a similar way to Oppelt [36]. The general idea of making each edge correspond to network components with a defined characteristic relation  $\Delta p = f(\dot{m})$  is taken. However the components differ and are implemented in a different manner. This formalism will be useful to set up the system of equations to solve in order to find the pressure and mass flow in all components of the district network.

A directed graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  is the pair containing the set of nodes (also called vertices or points)  $\mathcal{N}$ , and the set of edges (also called links or lines)  $\mathcal{E}$ . An edge is an unordered pair of nodes, so  $\mathcal{E} \subseteq \{\{x, y\} | (x, y) \in \mathcal{N}^2 \wedge x \neq y\}$ .

In the present case, edges are pipes, substations or thermal power plants. We note that a supply pipe and the corresponding return pipe correspond to two distinct edges. In the present case, nodes are intersections between edges, for example a pipe tee fitting, which joins 3 pipes, or the fitting between a thermal power plant and the first pipe of the network.

Note that the edges are defined as unordered, in the sense that the heat transfer fluid can flow through them in both directions. It will be useful for computations to define a direction of the mass flow and pressure drop in each edge. The edge links two nodes, define one as the tail, the other as the head. The convention used is to consider a mass flow from tail to head of edge  $i$  as positive  $\dot{m}_i > 0$  and a pressure drop from tail to head of edge  $i$  as positive  $\Delta p_i = p_{\text{tail},i} - p_{\text{head},i} > 0$ .

To define the structure of the graph, how the  $E$  edges and  $N$  nodes are connected, the edge-node index matrix  $\mathbf{A} \equiv [a_{ij}]_{N \times E}$  is introduced. It has  $N$  rows and  $E$  columns. Its elements are

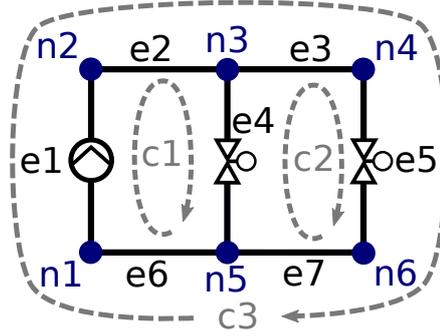


Figure 3.6 – Schematics of the graph formalism used to describe a linear district network with one thermal station, represented by the pump of edge  $e_1$ , and 2 substations, represented with valves on edges  $e_4$  and  $e_5$ . The cycles are  $c_1 = \{n_1, n_2, n_3, n_5\}$ ,  $c_2 = \{n_5, n_3, n_4, n_6\}$  and  $c_3 = \{n_1, n_2, n_3, n_4, n_6, n_5\}$ . Note that this is not a looped topology. The supply pipes are  $e_2$  and  $e_3$ , so it is just a linear topology of district network.

defined by equation (3.3).

$$a_{ij} = \begin{cases} 1, & \text{if edge } j \text{ points to node } i, \\ -1, & \text{if edge } j \text{ points away from node } i, \\ 0, & \text{if edge } j \text{ and node } i \text{ are not connected.} \end{cases} \quad (3.3)$$

To define the graph structure of the graph, how the  $E$  edges and  $N$  nodes are connected, the edge-node index matrix  $\mathbf{A} \equiv [a_{ij}]_{N \times E}$  is introduced. It has  $N$  rows and  $E$  columns. Its elements are defined by equation (3.3).

To characterize the  $L$  loops (also called cycle or mesh) of the graph, the edge-loop index matrix  $\mathbf{B} \equiv [b_{ij}]_{N \times L}$  is introduced. It has  $L$  rows and  $E$  columns. Its elements are defined by equation (3.4).

$$b_{ij} = \begin{cases} 1, & \text{if edge } j \text{ is in loop } i \text{ in the same direction,} \\ -1, & \text{if edge } j \text{ is in loop } i \text{ in the opposite direction,} \\ 0, & \text{if edge } j \text{ is not in loop } i. \end{cases} \quad (3.4)$$

### 3.2.4 Loops

The goal of this section is to develop the algorithm to find the edge-loop index matrix  $\mathbf{B}$ . To do so, one needs to define and understand cycles and their properties in a graph.

In a graph, a cycle  $c = \{n_0, n_1, \dots, n_k\}$  is a sequence of nodes  $n_i \in \mathcal{N}$ , which are all distinct, except the first and last one which are the same. So  $n_i = n_j$  if  $i = 0$  and  $j = k$ , or vice versa, otherwise  $n_i \neq n_j$ . Also any two consecutive nodes must be linked by an edge,  $\forall 1 \leq i \leq k$ ,  $\{n_i, n_{i+1}\} \in \mathcal{E}$ .

A graph with no cycles is called a tree. The graphs that will be used to represent the district

networks will have numerous cycles, for example, the ones starting at a thermal power plant supply pipe, going through the supply pipes to a substation, going through the substation to the return pipes, and going through the return pipes back to the thermal power plant. If the district network is what has been called a "tree-shaped", it means that there exists no cycle that only goes through supply pipes (similarly, no cycle only goes through return pipes).

The goal to make the  $\mathbf{B}$  matrix is to find a cycle basis. A cycle basis is a minimal set of cycles that allows every Eulerian subgraph to be expressed as a symmetric difference of basis cycles. This means that any cycle of the graph can be created by doing the symmetric difference of cycles in the cycle basis. The symmetric difference (also known as disjunctive union) between two cycles is the cycle made up of edges that are present in one of the cycles, but not both. For example in Figure 3.6, the symmetric difference of cycles  $c_1$  and  $c_3$  is cycle  $c_2$ . For a set of cycles to be a basis, it must be minimal. This means that if any cycle is removed from the basis, then not all cycles of the graph can be obtained through symmetric difference operations between the others.

To illustrate these definitions, some examples can be taken based on Figure 3.6. This graph contains 3 cycles,  $c_1$ ,  $c_2$  and  $c_3$ . There are 3 possible cycle bases:  $\mathcal{B}_1 = \{c_1, c_2\}$ ,  $\mathcal{B}_2 = \{c_2, c_3\}$  and  $\mathcal{B}_3 = \{c_1, c_3\}$ . The basis are complete since all cycles of the graph can be made from cycles in the basis. Indeed  $c_1 \oplus c_2 = c_3$ ,  $c_2 \oplus c_3 = c_1$  and  $c_1 \oplus c_3 = c_2$ . The basis are minimal since one cannot make the other two cycles with only one cycle.

To find a cycle basis, spanning trees are used. A subgraph  $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$  of the graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , is a graph whose nodes and edges are part of the original graph, so  $\mathcal{N}_s \subseteq \mathcal{N}$  and  $\mathcal{E}_s \subseteq \mathcal{E}$ . A spanning tree of  $\mathcal{G}$  is a subgraph  $\mathcal{G}_s$  which spans all nodes, so  $\mathcal{N}_s = \mathcal{N}$ , and that is a tree, meaning all nodes can be connected by a path, but there are no cycles. For example of Figure 3.6, the subgraph containing all nodes and edges except for  $e_6$  and  $e_7$  is a spanning tree.

From a given spanning tree  $\mathcal{G}_s$ , if one adds any edge  $e$  such that  $e \in \mathcal{E}$  but  $e \notin \mathcal{E}_s$ , it creates a cycle, called a fundamental cycle. A short proof is that since  $\mathcal{G}_s$  is a tree, there exists a path connecting the two nodes linked by  $e$ , so adding  $e$  creates a cycle. In fact all edges that are in  $\mathcal{G}$  but not in  $\mathcal{G}_s$  correspond to a fundamental cycle, and together they form a cycle basis  $\mathcal{B}$ . That set is minimal since: for any cycle  $c \in \mathcal{B}$ , the set  $\mathcal{B}' = \mathcal{B} \setminus c$  is not a basis. Indeed, take edge  $e$  of  $c$  such that  $e \in \mathcal{E}$  but  $e \notin \mathcal{E}_s$  (the edge that wasn't in the spanning tree), then  $e$  is not part of any of the other cycle  $c' \in \mathcal{B}'$ , meaning cycle  $c$  cannot be made by symmetric difference of the other cycles in  $\mathcal{B}'$ , which therefore isn't complete, and cannot be a basis. One could also prove that  $\mathcal{B}$  is complete.

One can check that for connected graphs that are trees, the number of nodes is one more than the number of edges,  $N_{\text{tree}} = E_{\text{tree}} + 1$ . So for general connected graphs, the spanning tree has a number of edges  $E_s = N_s - 1 = N - 1$ . Since each edge that is in  $\mathcal{E}$  but not in  $\mathcal{E}_s$  corresponds to a fundamental loop, number of fundamental loops is  $L = E - E_s$ , therefore  $L = E - N + 1$ .

To find a cycle basis for  $\mathbf{B}$ , we must find a spanning tree. It can be done through standard algorithms, in this thesis a breadth first search variant is implemented. The basic idea is to start from a node  $n_0$ , then explore the graph from node to node by following edges. When visiting a node  $n$ , for all of its adjacent edges  $e \in \text{Adj}(n)$ , follow edge  $e$  to the opposing node  $n'$ .

- If node  $n'$  has not yet been discovered, mark edge  $e$  as part of the spanning tree and

enqueue node  $n'$  for later visit.

- If node  $n'$  has already been discovered, do nothing and don't visit  $n'$ .

This algorithm finishes when all nodes have been discovered. The spanning tree is made up of marked edges, making up the set  $\mathcal{E}_s$ . All unmarked edges represent a loop. To find the loop associated to  $e \in \mathcal{E} \setminus \mathcal{E}_s$ , find the path  $p$  in the spanning tree  $\mathcal{G}'$  that connects the head and tail nodes of  $e$ . The loop is then the concatenation of edges in  $p$  and edge  $e$ .

### 3.2.5 Kirchhoff's nodal rule

For the solution to make physical sense, it must respect some laws. A common physical hypothesis for macroscopic non-relativistic situations such as here, is conservation of mass. More precisely in our case conservation of mass flow. For any pipe or every node, since there is no accumulation or source or sink of heat transfer fluid, the sum of mass flows coming in must equal the mass flows that come out. In other contexts this equation is called the continuity equation, or Kirchhoff's current law.

For an edge (e.g. pipe, substation) it simply means that the mass flow entering at one end equals the mass flow leaving at the other, so they can be fully described by only one scalar  $\dot{m}_{\text{edge},i}$  [kg/s].

For a node  $n_i$ , it means that

$$\sum_{\text{edge } j \text{ pointing to } n_i} \dot{m}_{\text{edge},j} - \sum_{\text{edge } j \text{ pointing away from } n_i} \dot{m}_{\text{edge},j} = 0. \quad (3.5)$$

Since equation (3.5) must be respected for all nodes, there are  $N$  simultaneous equations that can be written in matrix using matrix  $\mathbf{A}$ , defined in equation (3.3):

$$\mathbf{A} \cdot \vec{m}_{\text{edge}} = \vec{0}, \quad (3.6)$$

where  $\vec{m}_{\text{edge}}$  is the vector of size  $E$  containing the mass flow of each edge.

A way to impose that this condition is true, is to realize that any flow through a graph from a district thermal network can be expressed as the sum of elementary loop flows. More concretely, what is called here an elementary loop flow (ELF) corresponding the loop in the  $i^{\text{th}}$  row of matrix  $\mathbf{B}$ , is the flow with  $\dot{m}_{\text{edge}} = 0$  in every edge, except those in the  $i^{\text{th}}$  loop. It can then be seen that that this elementary loop flow has a simple matrix notation, which is  $\vec{m}_{\text{edge}, \text{ELFi}} = \mathbf{B}^T \cdot \vec{m}_{\text{loop}, \text{ELFi}}$ , where  $\vec{m}_{\text{loop}, \text{ELFi}}$  is a vector of size  $L$ , with zeros everywhere except in the  $i^{\text{th}}$  location,  $[\vec{m}_{\text{loop}, \text{ELFi}}]_j = \dot{m} \delta_{ij}$ . The symbol  $\delta_{ij}$  is the Kronecker delta defined as  $\delta_{ij} = 1$  if  $i = j$  but  $\delta_{ij} = 0$  if  $i \neq j$ . These elementary loop flows can then be added to make up any mass conserving flow. Therefore

$$\vec{m}_{\text{edge}} = \mathbf{B}^T \cdot \vec{m}_{\text{loop}}. \quad (3.7)$$

These mass flows intuitively respect mass conservation, this can be proven by showing that

$\mathbf{A} \cdot \mathbf{B}^T = \mathbf{0}$  meaning that indeed it respects mass conservation

$$\mathbf{A} \cdot \vec{m}_{\text{edge}} = \mathbf{A} \cdot \mathbf{B}^T \cdot \vec{m}_{\text{loop}} = \vec{0} \quad (3.8)$$

To prove that  $\mathbf{A} \cdot \mathbf{B}^T = \mathbf{0}$ , one can focus on element  $[\mathbf{A} \cdot \mathbf{B}^T]_{ij}$ , which is the scalar product of  $\mathbf{A}$ 's  $i^{\text{th}}$  row, corresponding to the  $i^{\text{th}}$  node, and  $\mathbf{B}$ 's  $j^{\text{th}}$  row, corresponding to the  $j^{\text{th}}$  loop. So  $\mathbf{A}$ 's  $i^{\text{th}}$  row has mostly zeros, a plus one each time an edge points to the  $i^{\text{th}}$  node and a minus one each time an edge points away from the  $i^{\text{th}}$  node. Since  $\mathbf{B}$ 's  $j^{\text{th}}$  row corresponds to the fundamental loop, so it is zero on edges that are not in the loop, plus one on edges that are in the loop in the same direction, and minus one on edges that are in the loop in the opposite direction. So the scalar product is the number of edges in the loop that enter the node, minus the number of edges in the loop that exit the node. Since the loop is fundamental, it either never enters a node, or enters it once and exits it once, meaning that the scalar product is zero. This is a reduction of the number of free variables to solve for, instead of having to solve and find the mass flow for each of the  $E$  edges, we only need to solve and find the mass flow for each of the  $L$  loops. Indeed the  $N$  constraints of equation (3.5) has reduced the number of unknowns from  $E$  to  $L$ .

Intuitively one can see that every edge must be part of at least one loop. Otherwise it is in a path of the graph that leads to a dead end, meaning no flow can flow through. The software verifies that such cases do not occur, and throws an error to the user explaining the topology problem of the input.

### 3.2.6 Kirchhoff's loop rule

The next laws that the solution must respect in order to make physical sense is "loop" or "mesh" law. It comes from the fact that the pressure [Pa] at each node is a well defined scalar function. So if to express this function we do not take the pressure at each node, but instead the pressure loss at each edge  $\Delta p_{\text{edge},i} = p_{\text{tail},i} - p_{\text{head},i}$  [Pa], then to ensure that the pressure is well defined, the sum of pressure losses around a loop must equal zero. This means that if there is a pressure loss in some part of the loop (e.g. a pipe whose rough walls are causing dissipation through friction) then some other part of the loop must compensate by re-increasing the pressure (e.g. with a pump).

In the analogy with electrical circuits, this is Kirchhoff's loop rule. It applies not to pressure, but to the voltage. Indeed the pressure is the hydraulic potential energy density, Pascals have dimensions of Joules per cubic meter. Similarly the voltage is the electrical potential energy, it has dimensions of Joules per Coulomb (energy per unit electric charge).

For the  $i^{\text{th}}$  loop, this condition imposes  $\sum_{\text{edge } j \text{ in loop } i} \Delta p_{\text{edge},j} = 0$ . The matrix form of these  $L$  equations can be expressed using  $\mathbf{B}$ , defined in equation (3.4):

$$\mathbf{B} \cdot \vec{\Delta p}_{\text{edge}} = \vec{0}, \quad (3.9)$$

where  $\vec{\Delta p}_{\text{edge}}$  is the vector of size  $E$  containing the pressure loss of each edge.

This will be the main set of equations to solve to get the mass flows in the district network.

Table 3.1 – Absolute roughness  $\epsilon$  of some common materials [29].

| Material                   | roughness [mm] |
|----------------------------|----------------|
| Concrete                   | 0.30-3.0       |
| Cast Iron                  | 0.26           |
| Commercial or Welded Steel | 0.045          |
| PVC, Drawn Tubing, Glass   | 0.0015         |

This will be possible because in each edge, there is a known relationship between the mass flow and pressure  $\Delta p_{\text{edge},i} = \varphi_i(\dot{m}_{\text{edge},i})$ . The more general vector form is

$$\vec{\Delta p}_{\text{edge}} = \vec{\varphi}(\vec{m}_{\text{edge}}). \quad (3.10)$$

The final system of  $L$  equations to solve in order to determine the  $L$  unknown loop mass flows  $\vec{m}_{\text{loop}}$ , which determine the pressure and mass flows throughout the network is obtained by combining equations (3.7), (3.9) and (3.10) to give:

$$\mathbf{B} \cdot \vec{\varphi}(\mathbf{B}^T \cdot \vec{m}_{\text{loop}}) = \vec{0} \quad (3.11)$$

The resolution algorithm is a variant of the Newton-Raphson method, explained in section 3.2.10.

### 3.2.7 Pipes

As the equation (3.10) shows, there is a need to determine how the pressure evolves through a pipe, especially its link to the mass flow.

The first essential characteristic of a flow in a pipe, is that there is friction between the fluid and the pipe walls. This friction happens when the fluid has finite viscosity and encounters defects on the surface of the pipe. It creates a pressure drop and increases the temperature because of the dissipation.

The roughness of the pipe surface  $\epsilon$  [m] is the characteristic length of the deviations from the perfect ideal plane. It is the characteristic size of hills and valleys. Having smooth pipe surfaces can sometimes be more expensive. Some values from common materials are shown in Table 3.1. Unless mentioned otherwise, the default value used is the roughness of commercial steel,  $\epsilon = 0.045$  mm.

The viscosity of the fluid is its resistance to deformation. It quantifies the strength of the resisting friction force between neighbouring volume elements of fluid which have relative velocity. It can be described by the dynamic viscosity  $\mu$  [Pa·s]. The viscosity of water depends on temperature  $T$  [K]. The semi-empirical Vogel-Fulcher-Tammann relation is used,

$$\mu(T) = A \exp\left(\frac{B}{T - C}\right), \quad A = 0.02939 \text{ mPa} \cdot \text{s}, \quad B = 507.88 \text{ K}, \quad C = 149.3 \text{ K}. \quad (3.12)$$

The Reynolds number in the pipe is  $Re = \frac{2\dot{m}}{\pi R \mu}$  [-]. This number is important in fluid mechanics as it helps predict the pattern of the flow. Small Reynolds number tend to give smooth laminar flows, and big Reynolds number tend to give turbulent chaotic flows.

The loss of pressure due to friction in a pipe can be approximated by the Darcy-Weisbach formula

$$\Delta p_{fr} = \frac{L f_D}{4\pi^2 \rho R^5} \dot{m}^2, \quad (3.13)$$

with the length of the pipe  $L$  [m], the inner radius of the pipe  $R$  [m], the mass density  $\rho$  [kg/m<sup>3</sup>], the mass flow  $\dot{m}$  [kg/s] and the Darcy friction factor  $f_D$  [-]. The Darcy friction factor actually depends on Reynold number, and therefore on the mass flow  $f_D = f_D(\dot{m})$ . It's value is

- if  $Re \leq 2320$ , the flow is laminar and there is an explicit formula

$$f_D = \frac{64}{Re}. \quad (3.14)$$

- if  $4000 \leq Re$ , the flow is turbulent and so the Darcy coefficient is defined by the implicit Colebrook-White equation

$$\frac{1}{\sqrt{f_D}} = -2 \log \left( \frac{\epsilon}{7.42R} + \frac{2.51}{Re \sqrt{f_D}} \right) \quad (3.15)$$

There exists ways to solve it, among which the Haaland explicit formula [31] which has been implemented in the code

$$f_D = \left[ 1.8 \log_{10} \left( \left( \frac{\epsilon}{7.4R} \right)^{1.11} + \frac{6.9}{Re} \right) \right]^{-2} \quad (3.16)$$

- if  $2320 < Re < 4000$ , this is the transition flow, which is not fully laminar nor turbulent. The uncertainties are large in this regime, the approximation taken to compute the Darcy coefficient is an affine function, such that the function is continuous throughout the regimes

$$f_D = f_D(2320) + \frac{f_D(4000) - f_D(2320)}{4000 - 2320} (Re - 2320). \quad (3.17)$$

This can be visualised with the Moody diagram in Figure 3.7. The turbulent regime's function will change depending on the relative roughness  $\frac{\epsilon}{2R}$ .

There is no analytical derivative of the Darcy coefficient with respect to the mass flow  $\frac{df_D}{d\dot{m}}$  in the turbulent regime. But this derivative is usually quite small compared to the derivative of the total  $\frac{d\Delta p_{fr}}{d\dot{m}}$ , so it will be considered constant during the derivative computations.

The heat capacity  $c_p$  [J/(K·kg)] and the mass density  $\rho$  [kg/m<sup>3</sup>] of water don't vary significantly over the temperature range used for district energy centres, that is, between zero and one hundred degrees Celsius. Therefore, these quantities are considered constant during the

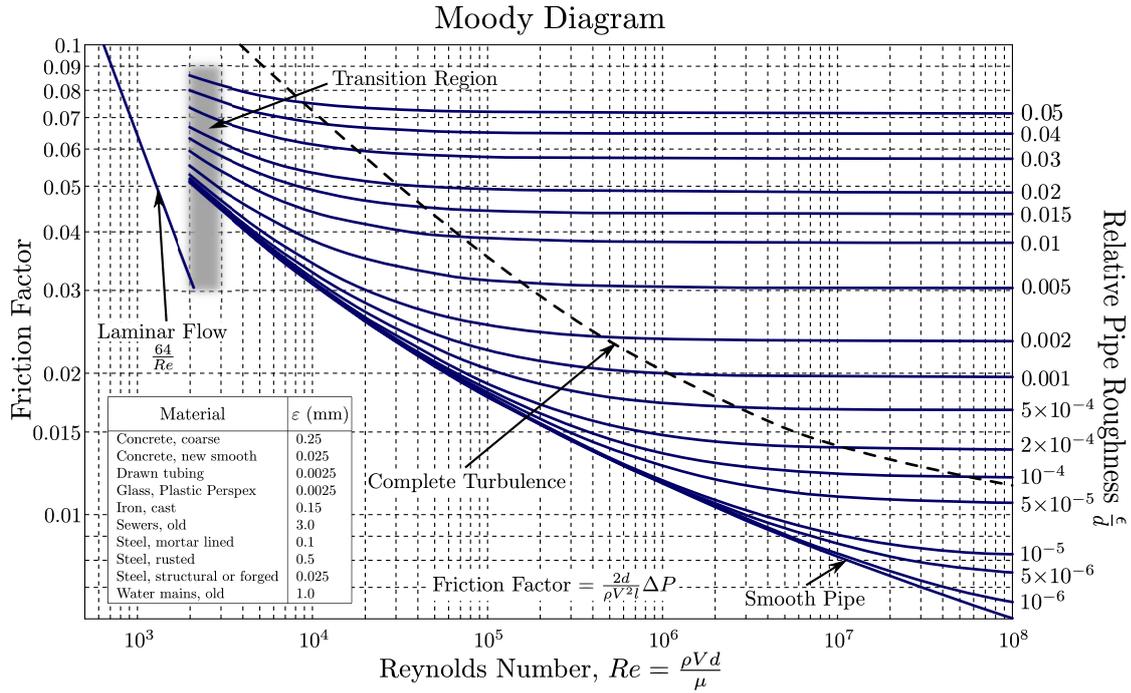


Figure 3.7 – Moody diagram explaining the relation between friction pressure drop and the mass flow through the pipe [11].

simulation. Figure 3.8 shows this for the mass density.

This frictional pressure drop is essential for the convergence to work, especially in cases where there are multiple thermal power plants imposing differential pressure with their pumps. Otherwise there would be no energy dissipation and the algorithm could converge to non-physical results. This is also important as this effect can become significant for substations that are far away from the power plants, especially with pipes of small radius. Therefore CitySim could be used for dimensioning a network. Also some thermal power plants control their pumps using the pressure difference around the "furthest" substation, and without taking into account frictional losses, this control strategy would be pointless.

Of course conservation of energy applies, so the high quality hydraulic pressure energy is therefore dissipated by the friction into lower quality thermal energy. The heat dissipated  $\dot{Q}_{fr}$  [W] is

$$\dot{Q}_{fr} = \frac{\dot{m} \Delta p_{friction}}{\rho} \tag{3.18}$$

Another source of pressure change in the pipes comes from the famous Bernoulli equation, which comes from an altitude change,

$$\Delta p_{alt} = \rho g \Delta z, \tag{3.19}$$

with  $\Delta z = z_{head} - z_{tail}$  [m] the altitude difference and  $g = 9.81 \text{ m/s}^2$  the gravitational accelera-

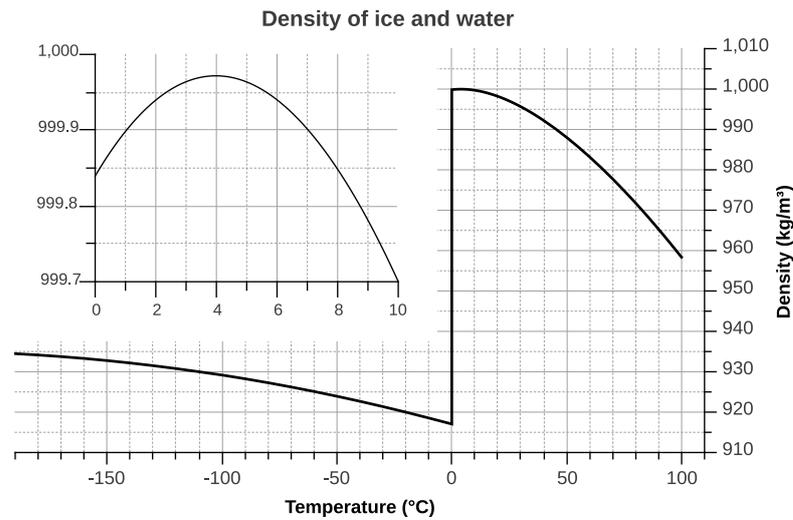


Figure 3.8 – Density of water as a function of temperature. Between zero and one hundred degrees, the change is not significant, only 4% [10].

tion. In this effect there is no heat dissipation since it is a transfer between hydraulic energy to gravitational potential energy. Physically it comes from the balance of forces on a volume element of fluid, which must carry the weight of the water column above him.

The pipe thermal model is described in section 3.1.4 which describes the thermal loss model. When considering the heat source from the friction dissipation, the pressure loss is only the one due to friction  $\Delta p_{fr}$ , without adding the one due to altitude change  $\Delta p_{alt}$  (the energy in converted into potential gravitational energy, not heat).

In reality there can also be a pressure loss due to the junctions of pipes : the node punctual pressure losses. These have not been modelled and are considered too complicated to implement for a non-significant correction compared to the error from uncertainties in materials or soil conductivity, etc. They could be implemented in future versions of the code.

### 3.2.8 Valves

The goal of substations is to extract the correct amount of thermal energy from the primary network, to heat the flow of its secondary network to a target temperature. As explained in section 2.2.1, our model substation has a desired thermal power needs  $\dot{Q}_{needs}$  [W], computed by the building's thermal model in order to get the correct space heating and domestic hot water heating powers. To get that power, the substation's strategy is to get a desired mass flow  $\dot{m}_p$  [kg/s] through the primary side of its heat exchanger (see Figure 2.2), which is a function of the desired thermal power needs  $\dot{m}_p = f(\dot{Q}_{needs})$ .

To get this desired mass flow, the substation can apply more or less hydraulic resistance in the pipes of the primary side. To apply this resistance, a valve is used. The valves are modelled as a passive component, that doesn't consume any electricity nor power. The equation that dictates the relation between the mass flow  $\dot{m}$  [kg/s] passing through it and the pressure loss

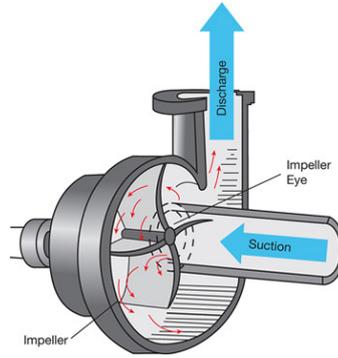


Figure 3.9 – Schematics of a centrifugal pump [37].

$\Delta p_{\text{loss}}$  [Pa] it creates is

$$\Delta p_{\text{loss}} = \Delta p_0 \frac{3600^2}{\rho_0 \rho K_v^2} \dot{m}^2, \quad (3.20)$$

with  $\Delta p_0 = 10^5$  Pa = 1 bar,  $\rho_0 = 1000$  kg/m<sup>3</sup>,  $\rho$  [kg/m<sup>3</sup>] is the mass density of the fluid passing through the valve and  $K_v$  [m<sup>3</sup>/h] is the flow coefficient of the valve. The flow coefficient is a measure of how "open" the valve is. More concretely it is by definition for volume flow in m<sup>3</sup>/h, of a liquid of density  $\rho_0 = 1000$  kg/m<sup>3</sup>, that would pass through the valve around which is imposed a pressure difference of  $\Delta p_0 = 1$  bar. A valve always causes a pressure drop in the flow direction, as it poses a resistance. Similarly to the friction in pipes, the pressure drop is proportional to the mass flow squared  $\Delta p_{\text{loss}} \propto \dot{m}^2$ .

The valve, similarly to the pipe, reduces pressure by friction, which heats the fluid as shown in equation (3.19). Therefore there is a temperature increase of

$$\Delta T_{\text{gain,friction}} = \left| \frac{\Delta p}{\rho c_p} \right|. \quad (3.21)$$

### 3.2.9 Pumps

The pump is an essential component to make the district heating network model work. Its goal is to provide hydraulic energy to the fluid, to make it flow through the pipes so that the substations get the desired flow and compensate the frictional losses.

The model used for this thesis is based on centrifugal pumps. A centrifugal pump works by consuming electricity to make a rotor rotate as shown in Figure 3.9. That rotor has a specific shape to make the fluid inside the pump go in a given direction by using the centrifugal force. In practice, the rotational speed  $n$  [1/min] of the rotor can be controlled by setting the corresponding frequency to the electrical supply of the pump's motor. The relation between the mass flow  $\dot{m} > 0$  and pressure increase  $\Delta p_{\text{raise}}$  [Pa] provided by a pump at rotational speed

$n$  is modelled as

$$\Delta p_{\text{gain}} = a_0 \frac{n^2}{n_0^2} + a_1 \frac{n}{n_0} \dot{m} + a_2 \dot{m}^2, \quad (3.22)$$

with  $n_0$  [1/min] the nominal rotational speed that the pump cannot exceed, technical performance parameters of the pump  $a_0 > 0$  [Pa],  $a_1$  [Pa·s/kg] and  $a_2 < 0$  [Pa·s<sup>2</sup>/kg<sup>2</sup>]. To understand the meaning of these parameters, the  $a_0$  corresponds to the pressure gain at nominal speed  $n = n_0$  against a huge resistance, so  $\dot{m} \rightarrow 0$ , so  $\Delta p_{\text{gain}}(n = n_0, \dot{m} = 0) = a_0$ . This is why  $a_0$  must be positive. The parameter  $a_1$  is the derivative of pressure gain with respect to mass flow at nominal speed  $n = n_0$  against a huge resistance, so  $\frac{\partial \Delta p_{\text{gain}}}{\partial \dot{m}}(n = n_0, \dot{m} = 0) = a_1$ . Finally the last coefficient  $a_2$  is negative, which means that for a given rotational speed, if the mass flow is increased sufficiently, the quadratic term  $a_2 \dot{m}^2$  will dominate, meaning the pressure gain will start decreasing. Indeed as the fluid flows faster, the pump's fixed rotational speed can apply less and less centrifugal force. And there is even a point at which the pressure gain goes through zero and becomes negative. Starting this point on, the pump acts as a resistance to the flow, because it rotates too slowly.

The hydraulic power  $\dot{W}_{\text{hyd}}$  [W] is the power that the pump gives to the fluid by increasing its pressure. It is

$$\dot{W}_{\text{hyd}} = \frac{\dot{m} \Delta p_{\text{gain}}}{\rho}. \quad (3.23)$$

However pumps are not ideal and usually also spend some energy heating the fluid through friction. Therefore the electric power needed depends on the efficiency  $\eta$  [-] of the pump, which is supposed fixed and within limits  $\eta \in [0, 1]$  in this thesis' model. So

$$\dot{W}_{\text{elec}} = \dot{W}_{\text{hyd}} + \dot{Q}_{\text{pump,friction}} = \frac{\dot{W}_{\text{hyd}}}{\eta} \quad (3.24)$$

Therefore the pump slightly preheats the fluid, which is why if a precise control over the input temperature is desired, the pump is placed before the boiler. The heat is

$$\dot{Q}_{\text{pump,friction}} = \dot{W}_{\text{hyd}} \frac{1 - \eta}{\eta} \quad (3.25)$$

### 3.2.10 Hydraulic convergence

The heart of the new model is to find the mass flows throughout the network. The variations of pressures and mass flows throughout the network propagate at close to the speed of sound in the fluid, at approximately  $10^3$  m/s. This is very fast compared to the other dynamics of the system. For example the temperature changes travel through the network at approximately

the speed of the fluid, because convection governs the dynamics, so at the order of 1 m/s. [19] For this reason the algorithm searches for a stationary hydraulic solution.

To improve the convergence, and make sure that the solution found is physically possible, the pump rotational speeds  $n$  [ $\text{min}^{-1}$ ] and the valve flow coefficients  $K_v$  [ $\text{m}^3/\text{h}$ ] (valve positions) remain constant during the hydraulic convergence. The goal of CitySim is to compute physical solutions, even if the thermal power plant cannot produce enough heat, or the valve is too small to let a desired mass flow through, or the operational regimes of the pumps are not compatible, etc. Letting these pump rotational speeds and valve positions change during the convergence, especially if valves and pumps have incompatible targets, leads can lead to oscillations during the convergence, that may be stuck without a physical solution.

The objective is therefore to find the loop mass flows  $\vec{m}_{\text{loop}}$  which are the roots of the error function

$$F(\vec{m}_{\text{loop}}) := \mathbf{B} \cdot \vec{\varphi}(\mathbf{B}^T \cdot \vec{m}_{\text{loop}}). \quad (3.26)$$

In other words solve equation

$$F(\vec{m}_{\text{loop}}) = \mathbf{0}. \quad (3.27)$$

We note that a solution exists for this physical problem and is unique. These boundary conditions make sense, the functions are continuous and well behaved. Pumps at a given rotational speed and valves at given openings, in real world result in a specific mass flow configuration in the network.

In our case, the equations are explicit and the derivatives can be computed. Note however that this is true when supposing that the Darcy friction factor's derivative with respect to mass flow is negligible, which is a good approximation. Therefore the Newton method is used to solve this problem and find the root of the error function. To do so the Jacobian matrix  $\mathbf{J} \equiv \left[ \frac{\partial F_i}{\partial \vec{m}_{\text{loop},j}} \right]_{L \times L}$  needs to be computed.

$$\mathbf{J}(\vec{m}_{\text{loop}}) := \mathbf{B} \cdot \frac{\partial \vec{\varphi}}{\partial \vec{m}_{\text{edge}}} \Big|_{\vec{m}_{\text{edge}} = \mathbf{B}^T \cdot \vec{m}_{\text{loop}}} \cdot \mathbf{B}^T \quad (3.28)$$

What is left to compute is the Jacobian matrix for the  $\vec{\varphi}$  function, which gives the pressure difference of each edge based on the mass flow taken as argument, as defined in equation (3.10). For each edge, the pressure difference can be computed using only the mass flow through that edge, as summarized in Table 3.2.

Table 3.2 – Pressure loss expressed as a function of mass flow, as well as the derivative, for the different types of edges.

| Edge Type | $\Delta p = \varphi(\dot{m})$  | $\frac{\partial \varphi}{\partial \dot{m}}$  |
|-----------|--|--|
| Pipe      | $\frac{L f_D}{4\pi^2 \rho R^5}  \dot{m}  \dot{m} + \rho g \Delta z$  | $\frac{L f_D}{2\pi^2 \rho R^5}  \dot{m} $  |
| Valve     | $\Delta p_0 \frac{3600^2}{\rho_0 \rho K_v^2}  \dot{m}  \dot{m}$  | $\Delta p_0 \frac{2 \cdot 3600^2}{\rho_0 \rho K_v^2}  \dot{m} $  |
| Pump      | $\begin{cases} -a_0 \frac{n^2}{n_0^2}, & \text{if } \dot{m} < 0, \\ -\left( a_0 \frac{n^2}{n_0^2} + a_1 \frac{n}{n_0} \dot{m} + a_2 \dot{m}^2 \right), & \text{else.} \end{cases}$ | $\begin{cases} 0, & \text{if } \dot{m} < 0, \\ -\left( a_1 \frac{n}{n_0} + 2a_2 \dot{m} \right), & \text{else.} \end{cases}$ |

In Table 3.2, one can notice that the dependence on mass flow is not quite quadratic  $\dot{m}^2$ , but rather  $|\dot{m}| \dot{m}$ . This is the case because a pipe and a valve will always reduce the pressure (so positive pressure loss  $\Delta p > 0$ ), in the direction of the flow. To solve the equations numerically, each edge has a direction, in which the mass flow is considered positive when flowing from the tail to the head of the edge. And the pressure loss is also considered positive in the direction of that flow  $\Delta p = p_{\text{tail}} - p_{\text{head}}$ . With this formalism, in pipes and valves, the pressure loss must have the same sign as the mass flow, justifying the absolute value notation.

One can also notice that the pump's second order polynomial approximation for the pressure is only valid when the mass flow is positive  $\dot{m} \geq 0$ . For negative mass flows, the pressure is constant and equal to  $\varphi(\dot{m} = 0)$ . This ensures that the function is continuous, which is better for convergence. This has been added to make more physical sense. Indeed if the polynomial is extended to negative mass flows, it can lead to non-physical results. To understand this, one can note that the polynomial has 2 roots, one is positive  $\dot{m}_{\text{root}+}$  and one is negative  $\dot{m}_{\text{root}-}$ . This is true because the pump rotation is always in range  $0 < n \leq n_0$ , which implies that  $0 < \frac{n}{n_0} \leq 1$ . It is also known that for small, positive mass flows, the pressure is increased through the pump. This means that the pressure loss is negative  $\Delta p = \varphi(\dot{m} \approx 0) \approx -a_0 n^2 / n_0^2 < 0$ , so  $a_0 > 0$ . It is also true that  $a_2 < 0$ , indeed if the pump has a given rotational speed, and that the mass flow increases to infinity  $\dot{m} \rightarrow +\infty$ , then the pump overall causes friction, and doesn't rotate fast enough to increase the pressure. This is what the positive root of the polynomial  $\dot{m}_{\text{root}+}$  corresponds to, it is the point where the mass flow becomes so large that the rotation of the pump is not fast enough to accelerate the flow, but instead opposes the flow, causing a pressure drop instead of an increase. Extending this polynomial to negative mass flows means that there is a second root  $\dot{m}_{\text{root}-}$ , and that for  $\dot{m} < \dot{m}_{\text{root}-}$ , a mass flow going fast enough backwards in the pump, the pressure is actually increased as that flow traverses the pump rotating in opposition to the flow. This indeed doesn't make physical sense, and could lead to non-physical solutions in the system. To avoid this, when the mass flow is negative through the pump, the pressure drop is set to a constant negative value  $-a_0 n^2 / n_0^2$ .

It is observed that for every edge, the pressure drop through itself depends only on the mass flow through itself, not on the mass flows through other edges. This means that  $\frac{\partial \varphi_{\text{edge},i}}{\partial \dot{m}_{\text{edge},j}} = 0$  if

$i \neq j$ , so the Jacobian is a  $E \times E$  diagonal matrix.

$$\frac{\partial \vec{\varphi}}{\partial \vec{m}_{\text{edge}}} = \text{diag} \left( \frac{\partial \varphi_{\text{edge},1}}{\partial m_{\text{edge},1}}, \frac{\partial \varphi_{\text{edge},2}}{\partial m_{\text{edge},2}}, \dots, \frac{\partial \varphi_{\text{edge},E}}{\partial m_{\text{edge},E}} \right) \quad (3.29)$$

The Newton scheme is an algorithm that aims to iteratively converge to the roots of an equation system. Its update formula is

$$\vec{m}_{\text{loop}}^{n+1} = \vec{m}_{\text{loop}}^n - \left[ \mathbf{J}(\vec{m}_{\text{loop}}^n) \right]^{-1} \cdot F(\vec{m}_{\text{loop}}^n) \quad (3.30)$$

We note that there is in general no need to invert the Jacobian matrix  $\mathbf{J}(\vec{m}_{\text{loop}}^n)$ . Indeed inverting a matrix demands more computations than simply solving the linear equation system

$$\mathbf{J}(\vec{m}_{\text{loop}}^n) \overrightarrow{\Delta m}^n = -F(\vec{m}_{\text{loop}}^n), \quad (3.31)$$

to obtain  $\overrightarrow{\Delta m}^n$ , the step vector of size  $L$ , such that

$$\vec{m}_{\text{loop}}^{n+1} = \vec{m}_{\text{loop}}^n + \overrightarrow{\Delta m}^n. \quad (3.32)$$

The Newton iteration as shown in equation (3.30) tries to linearize the non-linear function  $F$  around the trial step  $\vec{m}_{\text{loop}}^n$ . It then chooses the solution for that linearized system as next step  $\vec{m}_{\text{loop}}^{n+1}$ . This works very well and fast when the trial step is close enough to the solution  $\vec{m}_{\text{loop}}^*$ . Indeed the Newton method looks at the local derivatives and performs very well when these don't vary too much. But the Newton method doesn't have a very good global view of the function. If the trial step  $\vec{m}_{\text{loop}}^n$  is too far away from the solution  $\vec{m}_{\text{loop}}^*$ , the non-linearities can become predominant, making this step inefficient. This can sometimes lead to overshooting the solution, and getting stuck in a cycle by repeatedly overshooting the solution.

To improve the algorithm, the strategy is to try to stay inside the "trust region", where the linearisation is a good enough approximation. To do so, one can perform a damped newton iteration

$$\vec{m}_{\text{loop}}^{n+1} = \vec{m}_{\text{loop}}^n + \lambda_n \overrightarrow{\Delta m}^n, \quad \lambda_n \in ]0, 1]. \quad (3.33)$$

One can notice that this is a more general case of the basic Newton iteration in equation (3.32), in which there is always  $\lambda_n = 1$ . The goal now is to find an appropriate damping factor  $\lambda$ , which is not too big, which would leave the trust region, nor too small, which would be an inefficient step.

To do so, the NLEQ-RES algorithm presented in [12] is used, with some slight modifications. The notation  $\|\vec{x}\|$  stands for the Euclidean vector norm  $\|\vec{x}\| = \sqrt{\vec{x}^T \cdot \vec{x}}$ . CitySim therefore

implements the instructions presented in Algorithm 1.

Computing the update direction vector  $\overrightarrow{\Delta \dot{m}}^n$  requires to solving the linear system of equation, which can be quite computationally demanding. So it is only computed once, then multiples values of  $\lambda_n$  are tested, where the error function  $F(\vec{m}_{loop}^n + \lambda_n \overrightarrow{\Delta \dot{m}}^n)$  is evaluated. Indeed it is faster, and better to understand the function globally, to compute multiple times the error function, in order to get a good step size, rather than directly accepting the step and to compute the new direction  $\overrightarrow{\Delta \dot{m}}^n$ .

In this thesis, for practical reasons, some limits are implemented on the number of loops and "go to"s. This is to avoid that the simulation stays stuck in loops, or takes too long to converge. The chosen values are  $n_{max} = n_{reg\ test\ max} = n_{trial\ iter\ max} = 70$ .

For the first iterate  $\vec{m}_{loop}^0$  of the first time NLEQ-RES is called, the initial guess is taken as :

- For loops going through the first thermal power station and a substation: the nominal substation flow.
- For loops going through the first thermal power station and backwards trough another thermal power station: the sum of all nominal substation flows divided by the number of thermal power plants.
- For loops that are only in supply pipes, or only in return pipes: a random value uniformly chosen in interval [0.001, 0.002] kg/s.

This corresponds to the flow where every substation has the nominal flow, and all thermal power plants share equally that total demand. A small random value is given to the mass flow in other loops to avoid having mass flows of zero.

Once the NLEQ-RES algorithm has already been executed at least once, the first iterate  $\vec{m}_{loop}^0$  is taken as the solution returned by the previous execution of NLEQ-RES.

In this thesis, the stop criterion to decide that the algorithm has converged is that for each of the  $L$  loops, the sum of pressures around the loop (which we aim to be zero) is less than or equal to  $\epsilon = 100\ Pa = 0.001\ bar$ . Mathematically this translates to

$$\max_i F_i(\vec{m}_{loop}^n) \leq \epsilon. \quad (3.34)$$

It can be noted that the algorithm may not even need to perform the time consuming system resolution of equation (3.31) if the convergence criterion is already satisfied by the initial guess. This situation can actually occur quite often if there are no changes in the heat demand of substations from one iteration to the next.

After this algorithm has converged, the mass flows and pressure drops for each edge can be computed using equations (3.7) and (3.10). This algorithm could be improved further by rescaling the vectors and the matrices. This permits better convergence and less unnecessary numerical rounding errors, for example in the linear system resolution. What has already been done is to use the double precision of C++, where only floats are use throughout most of the program.

---

**Algorithm 1:** A slightly modified version of the NLEQ-RES Newton method, to find roots of a non-linear function  $F$  [12].

---

Initialize  $\vec{m}_{\text{loop}}^0$  with a guess,  $\lambda_0 = 1$ ,  $n = 0$  and  $\text{regTestFailed} = \text{false}$ .

Compute  $F(\vec{m}_{\text{loop}}^0)$

**while**  $n < n_{\text{max}}$  and  $\max_i F_i(\vec{m}_{\text{loop}}^n) > \epsilon$  and (not  $\text{regTestFailed}$ ) **do**

    Compute Jacobian  $\mathbf{J}(\vec{m}_{\text{loop}}^n)$ .

    Compute  $\vec{\Delta m}^n$  by solving linear system  $\mathbf{J}(\vec{m}_{\text{loop}}^n)\vec{\Delta m}^n = -F(\vec{m}_{\text{loop}}^n)$ .

**if**  $n > 0$  **then**

        Compute  $\mu_n = \frac{\|\vec{F}(\vec{m}_{\text{loop}}^{n-1})\|}{\|\vec{F}(\vec{m}_{\text{loop}}^n)\|} \mu'_{n-1}$  and  $\lambda_n = \min(1, \mu_n)$ .

**end**

    Set counters  $n_{\text{reg test}} = 0$  and  $n_{\text{trial iter}} = 0$ .

*Regularity test:*

**if**  $\lambda_n < \lambda_{\text{min}}$  **then**

$\text{regTestFailed} = \text{true}$

**else**

*Trial iterate:* Compute  $\vec{m}_{\text{loop}}^{n+1} = \vec{m}_{\text{loop}}^n + \lambda_n \vec{\Delta m}^n$  and  $F(\vec{m}_{\text{loop}}^{n+1})$

        Compute  $\Theta_n = \frac{\|\vec{F}(\vec{m}_{\text{loop}}^{n+1})\|}{\|\vec{F}(\vec{m}_{\text{loop}}^n)\|}$  and  $\mu'_n = \frac{\lambda_n^2 \|\vec{F}(\vec{m}_{\text{loop}}^n)\|}{2\|\vec{F}(\vec{m}_{\text{loop}}^{n+1}) - (1-\lambda_n)\vec{F}(\vec{m}_{\text{loop}}^n)\|}$ .

**if**  $\Theta_n \geq 1$  or  $\Theta_n > 1 - \lambda_n/4$  **then**

            Compute  $\lambda'_n = \min(\mu'_n, \lambda_n/2)$  and  $\lambda_n = \lambda'_n$ .

**if**  $n_{\text{reg test}} < n_{\text{reg test max}}$  **then**

                Increment  $n_{\text{reg test}} = n_{\text{reg test}} + 1$  and go to *Regularity test*.

**end**

**else**

            Compute  $\lambda'_n = \min(1, \mu'_n)$ .

**if**  $\lambda'_n \geq 4\lambda_n$  **then**

$\lambda_n = \lambda'_n$

**if**  $n_{\text{trial iter}} < n_{\text{trial iter max}}$  **then**

                    Increment  $n_{\text{trial iter}} = n_{\text{trial iter}} + 1$  and go to *Trial iterate*.

**end**

**end**

**end**

        Accept  $\vec{m}_{\text{loop}}^{n+1}$  as next iterate.

        Increment  $n = n + 1$ .

**end**

**end**

**Result:**  $\vec{m}_{\text{loop}}^{n+1}$  as approximation of  $\vec{m}_{\text{loop}}^*$ , the root of  $F$ .

---

### 3.2.11 Thermal convergence

There is a need to compute the temperatures throughout the system at a fixed hydraulic state, meaning with constant mass flows and pressures. To do so, a convergence is needed by looping through the network, following the mass flows.

Section 3.1.4 describes the basics of how the temperature computed through nodes and edges. Edges, such as pipes, substations or thermal stations, are given an "upstream" temperature that enters it. With that it can compute the "downstream" temperature that exits it, just before the next node. To compute the temperature at a node, one needs all the downstream temperatures from edges flowing to it, then compute an average weighted by the mass flows in each edge according to equation (3.2). This equation ensures energy conservation. The temperature at the node is then given as upstream temperature to all the edges it is connected to, towards which the flow goes.

The looped topologies are much more complex than the branched topologies. Also it is not impossible that the flow goes in the wrong direction. For example if there are two thermal stations near each other, one of them can impose a much bigger pressure difference than the other, meaning that mass flow can run backwards, from supply to return in the weaker thermal station. When the solar thermal injections will be added, flow may also intentionally flow in the wrong direction, for example a substation injecting solar power can pump fluid from return to supply.

To make sure that these cases are well treated, and to not get lost in infinite loops, the algorithm uses the following strategy :

- To initialise, suppose the temperatures in the supply nodes are the same as those from the previous time step.
- For all network components connecting a supply and a return node (thermal stations and substations), those where the flow goes from supply to return compute the downstream temperature.
- Propagate the temperature through all return nodes. In doing so, keep track of which nodes and pipes have already been traversed, to make sure they are only traversed once. Loop on all nodes, verify whether all the downstream temperatures from edges flowing towards him have been computed. If this is the case, compute the node temperature. Then compute the downstream temperatures for all pipes connected to it, towards which the flow goes. Then for all nodes where this leads, check whether the same computation can apply. This should be finished when the temperature of all return nodes have been computed.
- Similarly to second bullet point, propagate from return to supply.
- Similarly to third bullet point, propagate through the supply network.
- The convergence criteria is that node temperatures have changed by less than  $\epsilon$  [°C] during this loop. If it is not met, repeat from the second bullet point.

Unless mentioned otherwise, this parameter is set at  $\epsilon = 0.2^\circ\text{C}$ .

### 3.2.12 Overall convergence of valve positions and pump rotations

As we saw at each step the substations have a fixed valve flow coefficients  $K_v$ , the pumps have fixed rotational speed  $n$  and the hydraulics is converged to get a physical solution. With these fixed positions, the substations and the thermal power plants have specific targets that may not be met. Usually the substations have a target mass flow  $\dot{m}_{\text{target}}$  and the thermal power plants usually have a target pressure difference  $\Delta p_{\text{target}}$ , that can depend on other parameters, like the mass flow  $\Delta p_{\text{target}}(\dot{m})$ .

To converge to these target values, a loop is repeated as shown in Algorithm 2 where the  $K_v$  and  $n$  are updated, then the hydraulics convergence is performed, followed by the thermal convergence. The challenge is to find the appropriate strategy to update the  $K_v$  and  $n$  to converge overall in a stable and rapid manner.

---

**Algorithm 2:** Overall loop to converge to the valve flow coefficients  $K_v$  and pump rotational speeds  $n$  of the stationary solution where all targets are satisfied.

---

Initialize  $k = 0$ ,  $\text{notConverged} = \text{true}$

**while**  $k < k_{\text{max}}$  *and*  $\text{notConverged}$  **do**

    valves.updateKv()

    pumps.updateN()

    convergeHydraulic()

    convergeThermal()

$\vec{e}_{\text{rel}} = \text{computeRelativeError}(\text{substations}, \text{thermalPowerPlants})$

$\text{notConvergence} = (\max_i e_{\text{rel},i} \geq 0.02)$

$k = k + 1$

**end**

---

To measure for convergence, the relative errors are measured. All the elements that are being adjusted in order to reach a target (e.g. substations with mass flow, power plants with differential pressure) compute

$$e_{\text{rel}} = \left| \frac{x_{\text{current}} - x_{\text{target}}}{x_{\text{target}}} \right| \quad (3.35)$$

Different criteria are here possible, and could be implemented. The current algorithm requires that all relative errors be smaller than 0.02, so less than 2% of relative error, to qualify as a convergence. Usually one variable lags behind, and others go beneath that limit much earlier, leaving them iterations to converge even further. Therefore their relative errors are smaller than 2%. Mathematically the convergence criteria is

$$\max_i e_{\text{rel},i} < 0.02 \quad (3.36)$$

For practical reasons, to avoid the algorithm taking too much time, especially if it is stuck in a cycle, a maximum number of iterations is forced  $k_{\text{max}}$ . The current value is set to  $k_{\text{max}} = 100$ . The challenge is now to find a way to update the control variables,  $K_v$ s and  $n$ s, to obtain the

desired target values. The challenge is that the system is strongly coupled and non-linear. Changing one  $K_v$  or one  $n$  influences the rest of the system.

There is no simple, explicit, easy to compute the gradient of how the target quantities  $\dot{m}_{\text{target}}$  and  $\Delta p_{\text{target}}$  evolve as the control variables  $K_v$  and  $n$  are varied. Indeed to see the evolution one has to run the hydraulic convergence Algorithm 1. This excludes the possibility of an explicit gradient form.

There is however a possibility to quantify a gradient, and even the exact solution for a control variable, if one supposes that its modification doesn't change rest of the system. For example, if a substation whose goal is to find the correct flow coefficient  $K_v$  to get the desired  $\dot{m}_{\text{target}}$  can make the hypothesis that the pressure between its supply and return connections to the network stays at a constant  $\Delta p_{\text{ideal}}$ . Then in this ideal case, one can compute the exact

$$K_{v,\text{ideal}} = \dot{m}_{\text{target}} \sqrt{\frac{3600^2}{\rho_0 \rho} \frac{\Delta p_0}{\Delta p_{\text{ideal}}}}. \quad (3.37)$$

This hypothesis in essence is that the substation in particular has a very small influence on the rest of the whole network, which can adapt to the changes in that particular control variable. This hypothesis is indeed when the variation in control variable is small. This is particularly the case when the system is close to the converged solution. In general it also gives the correct direction in which to go (does the  $K_v$  need to be increased or decreased), as well as an intensity of the variation. This stays however a very local observation, that does not take into account the rest of the system.

An similar equation can be obtained for the pump rotational speed  $n$ . In this case, the hypothesis made is that the mass flow will not change as the control variable  $n$  is varied. Similarly, the hypothesis is that the rest of the system, substations and other thermal power plants will adapt in such a way that the mass flows doesn't vary in through that pump. With this ideal hypothesis, one can compute the exact rotational speed needed

$$n_{\text{ideal}} = n_0 \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad a = -a_0, \quad b = -a_1 \dot{m}_{\text{ideal}}, \quad c = -a_2 \dot{m}_{\text{ideal}}^2 - \Delta p_{\text{target}} \quad (3.38)$$

which results from the resolution of quadratic equation (3.22).

Simply updating using equations (3.38) and (3.37) works well in most cases standard easy cases, but easily gets stuck in cycles, without converging to the correct solution for more challenging district network configurations. For example, when there are multiple thermal power plants. Multiple approaches have been tested, the challenge being to converge to a solution fast, in as few as iterations as possible. The implemented solution is Robbins-Monroe annealing presented in section 3.2.13.

### 3.2.13 Robbins-Monroe annealing

The idea to converge to the correct control variables  $K_v$  and  $n$  is to update these variables using the ideal constant environment hypothesis with the equations (3.37) and (3.38), but

with a damping factor that increases at every iteration. For a control variable at  $x_{\text{prev}}$ , if the ideal value was taken for the new iteration, the step would be

$$\Delta x = x_{\text{ideal}} - x_{\text{prev}}. \quad (3.39)$$

Since all control variables change at once, there is a risk of over-steering which can prevent convergence, by being stuck in a cycle, stepping too far over the solution at each iteration or even by diverging away from the solution. To avoid this, the step can be damped, by multiplying the step by a learning rate  $\gamma$ , as such

$$x_{\text{new}} = x_{\text{prev}} + \gamma \Delta x. \quad (3.40)$$

If a constant  $\gamma$  is used at each iteration, the same problems can occur and convergence may not be obtained. Additionally, if the constant learning rate is small, it may also lead to very slow movements in the state space of control variables, which is also undesirable. The idea is therefore that the learning rate  $\gamma_k$  should be able to change depending on the iteration  $k$ , the same  $k$  as in Algorithm 2. The learning rate  $\gamma_k$  should decrease as the total number of iterations  $k$  increases.

To do so, the learning rate  $\gamma_k$  should decrease sufficiently fast so that the system is somewhat forced to converge, and cannot get stuck oscillating in a cycle. It should also decrease slowly enough so that the system can reach the solution, especially if it was initially far from it. A possibility is to use the Robbins-Monroe conditions

$$\sum_{k=0}^{\infty} \gamma_k = \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \gamma_k^2 < \infty \quad (3.41)$$

This can be achieved with functions of the form

$$\gamma_k = \frac{1}{(1+k)^r}, \quad \text{with } r \in (0.5, 1]. \quad (3.42)$$

In the current implementation,  $r = 0.75$  is used. This is a technique that is used in gradient descent algorithms in machine learning [47]. The name Robbins-Monroe comes from its original use in the stochastic approximation algorithm [42]. It was used to find the zero of a function while dealing with a stochastic noise, it was proved mathematically that such a series respecting these conditions would converge. It is similar to a probabilistic technique used to approximate the global optimum of functions called simulated annealing [27]. It consists of choosing iteratively random points where to evaluate the function, accepting the next step less and less as  $k$  increases, steps that are increasing the function. It is inspired from metallurgy where one slowly reduces the temperature in the metal to get bigger crystallization. The final

update formula implemented is

$$x_{k+1} = x_k + \frac{1}{(1+k)^{0.75}} (x_{\text{ideal},k} - x_k). \quad (3.43)$$

### 3.2.14 PID controller

The PID controller, short for proportional-integral-derivative, is a very widely used industrial control system employing feedback [5]. The mechanism's goal is to make a process variable  $y(t)$  follow as much as possible the setpoint  $r(t)$ , as often as possible in time  $t$ . To do so it can actuate a control variable  $x(t)$ . In our case, with a substation, the process variable  $y_t$  is the current mass flow  $\dot{m}_t$  through the substation, the setpoint  $r_t$  is the target mass flow  $\dot{m}_{\text{target},t}$  and the control variable is the flow coefficient  $K_v$ .

To achieve the setpoint, the PID controller continuously computes the error

$$e(t) = r(t) - y(t), \quad (3.44)$$

which is naturally the difference between the setpoint and the process variable. It then sets the control variable according to formula

$$x(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (3.45)$$

where  $K_p$ ,  $K_i$  and  $K_d$  are positive parameters defined by the engineer. The first term is the proportional term, meaning that if the process variable  $y(t)$  is below the setpoint  $r(t)$ , then the control variable is increased by an amount proportional to the current error  $e(t) = r(t) - y(t)$ . In second there is the integral term, if the error is on average non zero, the control variable  $x(t)$  will be increased, until the average error goes to zero. It is essential to get rid of the offsets. The last term takes into account the derivative of the error. Its goal is to accelerate the convergence and get rid of oscillations. In practice, the derivative term is not always used as it can cause instabilities.

It is commonly used because it is a very general controller that can work for a wide range of applications. It was initially built in analogue systems, such as automatically steering boats [34]. This mechanism is powerful as it can deal with stochastic noise such as gusts of wind, as well as varying offsets like changing marine currents. In our case the changing environment is made up of over substations and thermal power plants adjusting the valves and pumps throughout the network.

This mechanism doesn't need to know anything about the underlying mechanisms. The only requirement is that an increase in control variable  $x(t)$  should increase the process variable  $y(t)$  and vice versa. It works better when the couplings are linear, as can be deduced from the proportional term. In our case, this is rather true for the control variable  $K_v$ , which when the surrounding pressure stays constant, has a linear relationship with the process variable  $\dot{m}$  as

can be seen in equation (3.20). For the pump, the relation between the control variable  $n$  and the process variable  $\Delta p$  is quadratic when the mass flow stays constant, but the PID controller can handle it thanks to the integral term. The real challenge is the changing environment. The numerical implementation of the PID controller is shown in Algorithm 3

---

**Algorithm 3:** Pseudo-code of the PID controller class.

---

Initialize  $e_{prev} \leftarrow 0$  and  $e_{integral} \leftarrow 0$ .

Choose appropriate values for  $K_p$ ,  $K_i$  and  $K_d$ .

**Function** *computeControlVariable*( $r$ ,  $y$ )

```

     $e \leftarrow r - y$ 
     $e_{integral} \leftarrow e_{integral} + e$ 
     $e_{deriv} \leftarrow e - e_{prev}$ 
     $e_{prev} \leftarrow e$ 
     $x \leftarrow K_p \cdot e + K_i \cdot e_{integral} + K_d \cdot e_{deriv}$ 
    return  $x$ 

```

**end**

---

The downside of the PID controller is that the  $K_p$ ,  $K_i$ ,  $K_d$  parameters are non-trivial to tune. There are ways for tuning such as the Ziegler-Nichols method [59], but they usually require to disrupt the system many times, and between each disturbance slightly changing the parameters to observe how the fast the system reacts. In practice it would be too computationally demanding to do this for all the PID controllers. What is instead implemented in this thesis are constant coefficients, which are all the same for substation valves  $K_p = 0.5$ ,  $K_i = 0.3$ ,  $K_d = 0$ . For the thermal power plant pumps, these values are  $K_p = 0.3$ ,  $K_i = 0.2$ ,  $K_d = 0$ . One can notice that it is actually only a PI controller, not PID. The derivative term is not kept as it causes instabilities, preventing convergence.

In real life district networks where there are distributed hydraulic pumps using PID controllers, the hydraulic oscillation phenomenon can occur [57]. This is result of multiple uncoordinated PID controllers influencing each other. This can lead to damages in components such as pipes, pumps, etc.

A downside to using PID controllers is that to make sure they converge and stay stable in as many network configurations as possible, one needs to use rather small values for parameters  $K_p$ ,  $K_i$  and  $K_d$ . However, this leads to a very slow convergence, where many iterations, are necessary. This is especially penalizing in regimes where the heat demands don't change much, and the system starts close to the solution. In that case directly using the "ideal values" of equations (3.37) and (3.38) converges in almost one step. But the PID controller cannot make use of that value as help, and cannot take advantage of that knowledge to converge faster.

### 3.2.15 PID Continuous Action Reinforcement Learning Automata [22]

To deal with the problem of choosing the correct valve flow coefficients  $K_v$  and pump rotational speed  $n$ , the approach of reinforcement learning can be studied. Reinforcement learning is a domain of machine learning, in which agents have to learn behaviour against a dynamic

(sometimes hostile) environment [28]. Agents are entities that have means to observe their surroundings and take actions. In our case, it is the code that governs the valves or the pumps, that can observe the mass flow and differential pressure.

The agent must interact with its environment through trial and error, with the objective to maximize the cumulative reward. In the current case, the agent wants to minimize the error, that is the distance between the process variable and its setpoint. Minimizing the cumulative error would mean the sum total error summed at all time steps. To do so, the agent has to do exploration that is to say, try new actions to see if they improve the result, and learn how to improve its performance. Meanwhile it must also do exploitation, that is to take actions which have high reward. There is always a trade-off between exploration and exploitation. To deal with potential opponents, which can change the result of their actions, or in general with a stochastic, non-easily predictable environment, the agent usually uses itself stochastic strategies [44].

Most of these algorithms are using a discrete action space. In the current case, the action at each time step  $k$ , will be to choose a triplet  $K_p$ ,  $K_i$  and  $K_d$ . They can take continuous values between predefined min and max values. This leads to the name Continuous Action Reinforcement Learning Automata (CARLA) [22, 23] which uses continuous probability distributions to choose the continuous action. The idea is to have one CARLA for each of the variables  $K_p$ ,  $K_i$  and  $K_d$ , so three in total.

Each CARLA keeps, and makes evolve at each step  $k$  the probability distribution function  $f^k(x)$  over actions  $x$ . Initially it is uniformly distributed between the min and max values set by the user

$$f^k(x) = \begin{cases} \frac{1}{x_{\max} - x_{\min}}, & \text{if } x_{\min} \leq x \leq x_{\max}, \\ 0, & \text{otherwise.} \end{cases} \quad (3.46)$$

This function is kept in memory by saving the values at  $N$  positions  $\{x_1, x_2, \dots, x_N\}$ . The simplest way implemented here is to distribute them uniformly with  $x_i = \frac{N-i}{N-1}x_{\min} + \frac{i-1}{N-1}x_{\max}$ . Paper [23] suggests to sample the positions  $x_i$  in a way that depends on the distribution  $f^k(x)$ , so that the probability between each interval  $\int_{x_i}^{x_{i+1}} f^k(x)dx$  stays constant equal to  $\frac{1}{N-1}$ . This means sample more where the distribution function is big, and less where it is small. This could be implemented as improvement.

To select the action  $x^k$  of the  $k^{\text{th}}$  step from this probability distribution  $f^k(x)$ , one samples  $z^k$  from  $\mathcal{U}(0, 1)$ , the uniformly distributed random variable on interval  $[0, 1]$ . Then numerically find the value of  $x^k$  such that

$$\int_{x_{\min}}^{x^k} f^k(x)dx = z^k. \quad (3.47)$$

Each action  $x^k$  is then evaluated with a cost function  $J^k$ . For this thesis, the cost is the absolute value of the difference between the setpoint and the process variable of the PID process

resulting from step  $k$ .

The next step is from that cost  $J^k$ , to evaluate the performance  $\beta^k$ . To do so, the costs of the previous  $R$  steps are kept in memory  $\{J^{k-R}, J^{k-R+1}, \dots, J^{k-1}\}$ , and the performance is based on the comparison between the current cost to these previous costs. The formula used is

$$\beta^k = \min \left\{ \max \left\{ \frac{J_{\text{med}} - J^k}{J_{\text{med}} - J_{\text{min}}}, 0 \right\}, 1 \right\}, \quad (3.48)$$

where  $J_{\text{med}}$  and  $J_{\text{min}}$  are respectively the median and mode of the  $R$  previous costs kept in memory. This uses the reward/inaction rule. The performance is zero  $\beta^k = 0$  if the current cost is above the median cost  $J^k \geq J_{\text{med}}$ . Also, the maximum reinforcement reward is one,  $\beta^k = 1$ , when  $J^k \leq J_{\text{min}}$ .

The performance is used to update the probability density function according to formula

$$f^{k+1}(x) = \begin{cases} \alpha^k [f^k(x) + \beta^k H(x, x^k)], & \text{if } x_{\text{min}} \leq x \leq x_{\text{max}}, \\ 0, & \text{otherwise.} \end{cases} \quad (3.49)$$

where  $H(x, r)$  is a Gaussian centered at  $r$ ,

$$H(x, r) = \frac{g_h}{x_{\text{max}} - x_{\text{min}}} \exp \left( \frac{(x - r)^2}{-2[g_w(x_{\text{max}} - x_{\text{min}})]^2} \right), \quad (3.50)$$

with  $g_h$  and  $g_w$  the height and width parameters that determine respectively the speed and resolution of the learning. The values implemented in this thesis are those used by [23],  $g_h = 0.3$  and  $g_w = 0.02$ . To ensure that the probability density function stays normalized, equation (3.49) multiplies by the normalization factor  $\alpha^k$ , which is therefore defined by

$$\alpha^k = \frac{1}{\int_{x_{\text{min}}}^{x_{\text{max}}} [f^k(x) + \beta^k H(x, x^k)] dx} \quad (3.51)$$

which is computed by numerical integration.

The drawbacks of this method is that in the end it is very similar to a regular PID controller and have the same downside that it takes many iterations to converge. On top of that, the distribution functions for the parameters  $K_p$ ,  $K_i$  and  $K_d$  also take very long to converge, as many as tens of thousands of iterations. This is usually much less time then the number of pre-simulation days in CitySim. And the fact that sometimes random values are chosen means that the parameters can be too big, causing instabilities and fails to converge.

Some improvements of the CARLA have been proposed [43], which mainly changes the update formula for the probability density function in equation (3.49). This or other modern reinforcement learning techniques could be implemented to try and improve on this thesis' work.

### 3.2.16 Combination of Ideal and PID

As explained in section 3.2.12, under the "ideal" assumptions that the pressure drop stays constant through the valves, and the mass flows are constant through the pumps, then equations (3.37) and (3.38) can be used to update the control variables  $K_v$  and  $n$ . This update formula works well when the system is already near the solution, but will often lead to non converging oscillations if the system starts far from the solution.

Based on this observation, an intermediate solution has been implemented. Using the ideal update formula when the system is near the solution, but the PID controller when the system seems stuck in a cycle or far from the solution. The challenge from there is to define exactly how to separate these cases. To do so, every controller computes its relative error, as in equation (3.35). An additional indicator is used, the global relative error  $\bar{e}_{\text{rel}}$ , which averages the relative errors, with weights  $w_i$  proportional to the importance of the desired values. So that a big building with a target flow of 1 kg/s has a relative error that weighs 10 times more than a small house with target flow of 0.1 kg/s.

$$\begin{aligned}\bar{e}_{\text{rel}} &= \sum_i e_{\text{rel},i} w_i \\ &= \sum_i \frac{|x_{\text{current},i} - x_{\text{target},i}|}{x_{\text{target},i}} \frac{x_{\text{target},i}}{\sum_j x_{\text{target},j}} \\ &= \frac{\sum_i |x_{\text{current},i} - x_{\text{target},i}|}{\sum_j x_{\text{target},j}}.\end{aligned}\tag{3.52}$$

To avoid mixing quantities of different units in the denominator of the weights (in this case kg/s and Pa), there are two of these quantities  $\bar{e}_{\text{rel},\dot{m}}$  and  $\bar{e}_{\text{rel},\Delta p}$ . To compute  $\bar{e}_{\text{rel},\dot{m}}$ , take equation (3.52) and replace the  $x$  with mass flows through the substations  $\dot{m}$ . Similarly for  $\bar{e}_{\text{rel},\Delta p}$ , replace the  $x$  with pressure gains at the pumps  $\Delta p$ .

Another quantity worth monitoring to try and detect whether the system is stuck in a cycle is the relative fluctuation of the control variables  $K_v$  and  $n$ . It measures how big was the change from the previous to the current iteration.

$$f_{\text{rel}} = \frac{|x_{\text{current}} - x_{\text{previous}}|}{x_{\text{current}}}\tag{3.53}$$

Similarly, the global weighted averages can be performed  $\bar{f}_{\text{rel},K_v}$  and  $\bar{f}_{\text{rel},n}$  according to equation

$$\begin{aligned}
\tilde{f}_{\text{rel}} &= \sum_i f_{\text{rel},i} w_i \\
&= \sum_i \frac{|x_{\text{current},i} - x_{\text{previous},i}|}{x_{\text{current},i}} \frac{x_{\text{current},i}}{\sum_j x_{\text{current},j}} \\
&= \frac{\sum_i |x_{\text{current},i} - x_{\text{previous},i}|}{\sum_j x_{\text{current},j}}.
\end{aligned} \tag{3.54}$$

This can be parametrised by assigning a threshold to each : the relative fluctuation, the global relative fluctuation, the relative error and the global relative error. If either value is above its threshold, use the PID, otherwise use the ideal formulas. Automatically finding the appropriate thresholds can be challenging.

### 3.3 Solar thermal feed-in

This section describes how solar thermal energy is used to heat the network using centralised or decentralised configurations.

#### 3.3.1 Solar thermal collectors

The sun is a very interesting energy source, which can be of great help to decarbonise the energy supply. Its rays hit half of our planet at every instant, with a perpendicular power of  $1000 \text{ W/m}^2$  after going through the atmosphere [49], freely available to anyone below it. There is no need to be connected to a network to benefit from this energy source. Of course this is the perpendicular power, depending the location on the globe and time of the year, the ground is not oriented perpendicular to the sun and more atmosphere may need to be crossed, meaning the power per square meter of soil is less. This source is renewable as the sun will continue to shine for billions of years to come.

The environmental impact comes from the construction and disposal of the collectors, as well as the surface they take. During the operation they do not emit greenhouse gases. They can be placed on rooftops to limit the consumption of space. A disadvantage is the intermittent nature of the energy source, and the need of large amounts of materials to make enough panels to satisfy current human energy consumption.

The solar radiation is made up of photons which have a high quality of energy. This is why they can directly be used in photovoltaic panels to generate electricity. However, in practice, the sun is mostly used as a source of heat. It is one of the main heat providers for most houses. This is why most buildings in the northern hemisphere try to orient the windows towards the south.

There are also other ways to harvest the energy of the sun, which are less known to the common public. These methods usually collect energy from the sun, in the form of heat. The particularly interesting one, in the context of district heating are solar thermal collectors. There are multiple types, but the basic concept stays the same, make the sun rays heat the fluid in a pipe.

There are flat collectors as shown in Figure 3.10. It is a panel oriented towards the sun, similar

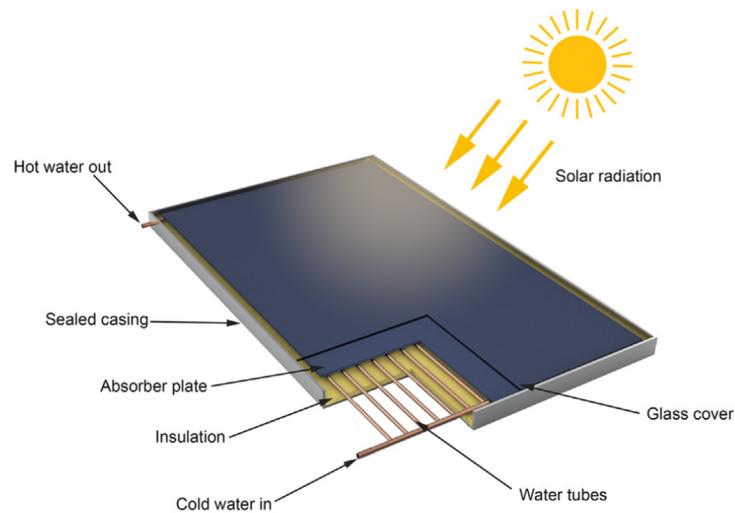


Figure 3.10 – Schematics of a flat plate solar thermal collector [45].

the photo-voltaic panels. At the bottom of the panel is the absorber. It is usually a black material that efficiently captures the photons and heats up in the process. At the top of the panel is a layer of glass that lets some photons through, while keeping the hot air inside the panel from escaping via convection. Inside the panel runs a tube with the heat transfer fluid that flows throughout the panel as it heats up.

The second common type of collectors is the evacuated tubular collector. As shown in Figure 3.11 they are made up of multiple tubes in which the fluid goes back and forth. To improve thermal insulation, there is a layer of vacuum between the fluid tube and the exterior air. This makes evacuated tube collectors usually more efficient compared to flat plate collectors especially when there is a big temperature difference between the outside air and the fluid. However, they are usually more expensive because of the complexity involved in creating the more complex shapes and the vacuum.

Solar thermal panels have the advantage of using cheap, easy to manufacture materials, that are more simple to recycle compared to the photovoltaic counterpart. Solar panels in areas like Switzerland require little maintenance, only to be cleaned from time to time. They are usually linked to heat storage tanks, which therefore offers a flexibility, to deal partly with the intermittent nature of this energy source. The goal in this thesis is to study how solar thermal panels could be used as an energy source in district heating networks.

The solar thermal panels are already implemented in CitySim as described in section 2.1. They are linked to surfaces, such as rooftops of buildings. The efficiency which depends on the fluid and outside temperatures as described in equation (2.1) will be used.

### 3.3.2 Integration in district heating networks

There are multiple ways to integrate solar thermal energy to a district heating network [20]. The first different lies in the centralized or decentralized nature. Centralized plants are similar to conventional means of heat production in district heating networks. A large field of solar

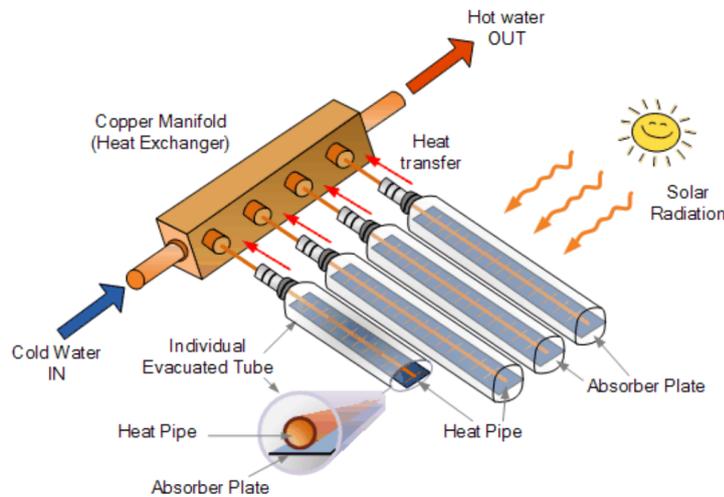


Figure 3.11 – Schematics of a evacuated tube solar thermal collector [53].

thermal collectors harvest the energy from the sun in one central location, and injects that energy into the network similarly to a conventional wood or gas thermal station. Usually they do not operate alone, they need to be assisted by a dispatchable means of production, such as a combined heat and power plant, which consumes wood, gas or wastes. This is especially important to deal with the peaks of heat demand by the consumers, and the intermittent nature of the sun. Without the additional dispatchable means of production, the solar installation would need to be much larger, or have large efficient means to store heat.

The other possibility is to have decentralized solar thermal production. This means that solar collectors of smaller size are installed in multiple locations in the district heating network. The typical case is that they are located on building rooftops, and they serve as a priority to satisfy the heating needs of the building. When buildings with solar thermal collectors are not connected to a solar thermal collector, they store the excess heat produced in a larger than usual storage tank. In the case where too much heat is stored, for example, the temperature of the water based storage is at the critical high limit, then the solar collector must evacuate the excess to the environment to protect the system from over-heating. However if the building is connected to a district heating network, the excess heat can be fed into the network. The heat consumers become heat producers, which gives rise to the name prosumers.

The centralized thermal plants use the classical scheme of taking cold fluid from the return pipes, heating it and re-injecting it into the supply pipe. In this return-supply connection (R/S), in this process to get from the low-pressure return pipes, to the high-pressure supply pipes, one needs to use a pump, which consumes electricity, therefore costing money. There are other possibilities to feed-in the heat as shown in Figure 3.12, the return-return (R/R), the supply-supply (S/S) and the supply-return (S/R). The connections that are most often used are the (R/S) and (R/R). The (S/S) and (S/R) connections on the other hand are rarely used. Advantages and disadvantages of the different connections are [20, 30] :

- The (S/R), (R/R) and (S/S) connections don't require a pump to function as they do not

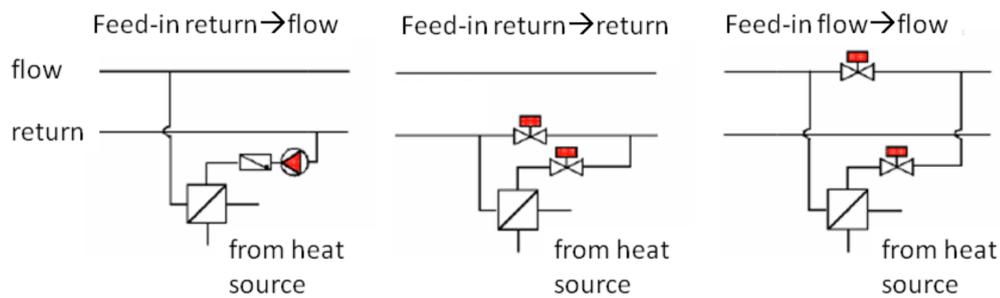


Figure 3.12 – Schematics of different solar thermal feed-in connections [20].

oppose the pressure drop. The (S/R) can use a single valve, similarly to a consuming substation. The (R/R) and (S/S) can use a pair of valves, in order to deviate the desired fraction of the flow from the regular pipe, to the solar thermal system.

- The (S/S) and (S/R) modes both take as input the fluid at hot supply temperature, then heat it further as solar heat is injected. The efficiency of the collector decreases when the fluid is hot, since the temperature difference between the fluid and the outside air is increased. This leads to bad efficiencies.
- The (R/R) and (S/R) connections both heat the return lines. This leads to more thermal losses through the pipes, especially since the return pipes have usually less thick insulation layers. The addition of insulation will be more expensive. The temperature arriving at the conventional thermal station will be higher. For most heating systems this is not desired, as it leads to lower efficiencies (especially in combined heat and power plants, where the Carnot coefficient depends on the temperature difference).
- The (S/S) and (R/R) connections are both don't need pumps at the feed-in location, but they are flow resistance, which is usually not desired by the district heating central operator, as it will cost more electricity for pumping. These also have the disadvantage that they can only work if the feed-in flow is small compared to the rest of the flow. Otherwise it will create a significant temperature increase, which can be problematic because misplaced and not easy to store if the total solar feed-in power becomes significant.

Rapid and frequent temperature variation cause damage to the pipes and network components. They are to be avoided.

There is a large freedom and complexity in the ways that the decentralised solar thermal power can be injected into the network [30]. The simple scheme of a centralized station doesn't apply any more. As solar power is fed-in it can cause oscillations in temperatures and flows throughout the network. Part of the complexity comes from the decentralisation. If the solar thermal system wants to inject heat, either it imposes the mass flow it wants with a pump, which means that flow direction will reverse in some pipes and perturb the current equilibria. Or the solar thermal system uses valves, and may not have enough flow running through its pipe to inject the available power.

Another part of the complexity comes from the fact that solar thermal power is intermittent. In addition, in efforts to decarbonise the energy, and save money, when solar power is available, it has priority and as much of it as possible is injected. That is, even if consumers don't need it. This means that either there should be a way to dissipate the extra heat, or what usually happens is that there is a large heat storage in the district network, to store it. This gives rise to big seasonal storages, that can store heat over multiple months. This is a way to store the heat overproduction in summer, and use it during the fall and winter.

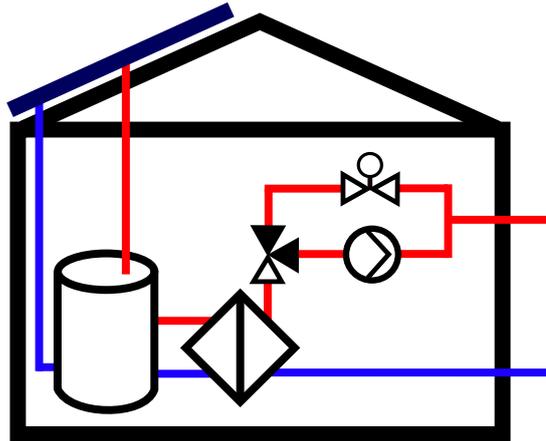


Figure 3.13 – Schematics of the hydraulic model of prosumer substation. When the substation is consuming, the three-way valve connects the heat exchanger to the simple valve, leaving the pump unused. In that case the flow is from supply to return, like a normal substation. When the substation is producing, the three-way valve connects the heat exchanger to the pump, leaving the simple valve unused. In that case the flow is from return to supply.

There are many different possibilities on how to operate a solar district heating system, none of which is universally accepted [17, 21, 6]. There are efforts to optimise the operation of such systems, but each network is different, having its own geometry and components, meaning the objective functions to optimise may vary, and that the results cannot be systematically applicable to other networks [9].

The choice made in this thesis for the control strategy has been decided using different guidelines found, among which [30]. Solar injections are done taking cold fluid from the return pipes, heating it and injecting it into the hot supply network. To fix degrees of freedom, it is imposed that the fluid is injected at the setpoint supply temperature, and the thermal power injected must be the total available solar thermal power.

This will be done via a new type of substation as described in Figure 3.13. When consuming, it uses the path with the simple valve to control the mass flow from supply to return, similarly to a normal substation. When producing, it uses the path with the pump, whose power is adjusted to match the desired mass flow.

When the sun shines very bright, some prosumers may be injecting more flow to the supply pipes than what is consumed by the rest of the consuming substations. To take care of this, the thermal stations equipped with seasonal storage will stop producing heat, and instead store

it. Figure 3.14 shows a schematic of the model. The thermal station can store the excess heat when the flow goes from supply to return, or heat up the fluid with both the thermal storage and the boiler, when the fluid goes from return to supply.

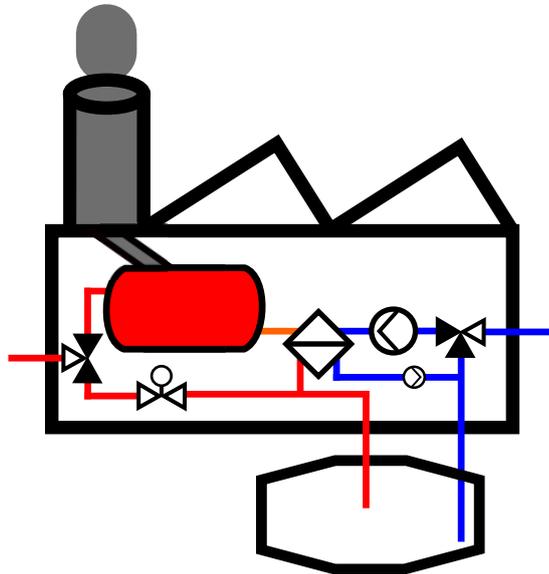


Figure 3.14 – Schematics of the hydraulic model of thermal station with seasonal storage. When the thermal station must store the excess heat, the three-way valves connect the seasonal storage to the outside network. The pumps, the boiler and the heat exchanger are not used, only the simple valve is used to regulate flow and pressure difference. In that case the flow is from supply to return, unlike a normal thermal station. When the thermal station is producing heat, the three-way valves connect the boiler to the outside network. The big pump is used to overcome the pressure difference to get the fluid from return to supply. The small pump is used to make the storage preheat the fluid through the heat exchanger.

### 3.3.3 Prosumer substation

This section details further the computations and algorithms behind the prosumer substation. In the C++ code, it is implemented by the class `ProsumerSubstation` which is a subclass of `Substation`. It is possible to make simulations with either of these two classes.

There are many possible ways to make hydraulic connections in the secondary side of the substation, to connect the solar thermal collector, the space heating tank, the domestic hot water tank and the substation. Since none of them is universally accepted, and to simplify the simulations, no precise hydraulic layout is used. Instead the overall energetic computations are made.

Algorithm 4 shows how this is done. Its goal is to attribute the solar thermal power to the different uses : to the space heating storage HS, the domestic hot water storage DHW and the power left will be fed into the network. The model uses the notion "fraction of solar power left", as a way to distribute the solar collector's power among these different uses. If  $f_{sol\ left} > 0$  then the prosumer substation can feed solar power into the district heating network.

---

**Algorithm 4:** Determine  $f_{\text{sol left}}$  the fraction of solar thermal power that is left to inject in the district heating network.

---

Initialize  $f_{\text{sol left}} \leftarrow 1$

Compute the heat demand  $\dot{Q}_{\text{need,HS}}$  and  $\dot{Q}_{\text{need,DHW}}$  for HS and DHW tanks.

Based on the  $\dot{Q}_{\text{need}}$ , compute the maximal power  $\dot{Q}_{\text{crit,HS}}$  and  $\dot{Q}_{\text{crit,DHW}}$  for HS and DHW tanks, above which they exceed the critical temperature after time step  $\Delta t$ .

Determine which of the HS and DHW tanks gets the solar power in priority (1  $\Rightarrow$  top priority, 2  $\Rightarrow$  least priority).

Compute the solar power that can be produced for the priority tank  $\dot{Q}_{\text{sol available},1} = f(T_1)$  (it is a function of its temperature).

**if**  $\dot{Q}_{\text{sol available},1} \leq \dot{Q}_{\text{crit},1}$  **then**

    Use all available solar power  $\dot{Q}_{\text{sol used},1} \leftarrow \dot{Q}_{\text{sol available},1}$ .

    No solar power is left  $f_{\text{sol left}} \leftarrow 0$ .

    Ask the heating unit for the remaining thermal power.

**else**

    Only use the critical power  $\dot{Q}_{\text{sol used},1} \leftarrow \dot{Q}_{\text{crit},1}$ .

    The fraction of solar power left is  $f_{\text{sol left}} \leftarrow 1 - \frac{\dot{Q}_{\text{sol used},1}}{\dot{Q}_{\text{sol available},1}}$ .

    Compute the solar power available of the least priority tank if only it is heated

$\dot{Q}_{\text{sol available},2} = f(T_2)$  (it is a function of its temperature).

    Since the top priority tank uses solar power, then  $\dot{Q}_{\text{sol available},2} \leftarrow f_{\text{sol left}} \cdot \dot{Q}_{\text{sol available},2}$ .

**if**  $\dot{Q}_{\text{sol available},2} \leq \dot{Q}_{\text{crit},2}$  **then**

        Use all available solar power  $\dot{Q}_{\text{sol used},2} \leftarrow \dot{Q}_{\text{sol available},2}$ .

        No solar power is left  $f_{\text{sol left}} \leftarrow 0$ .

        Ask the heating unit for the remaining thermal power.

**else**

        Only use the critical power  $\dot{Q}_{\text{sol used},2} \leftarrow \dot{Q}_{\text{crit},2}$ .

        The fraction of solar power left is  $f_{\text{sol left}} \leftarrow f_{\text{sol left}} \cdot \left(1 - \frac{\dot{Q}_{\text{sol used},2}}{\dot{Q}_{\text{sol available},2}}\right)$ .

**end**

**end**

**Result:**  $f_{\text{sol left}}$

---

When the prosumer substations feed-in hot fluid in the supply network, it must not affect negatively the other heat consumers. More specifically, the supply temperature at which the prosumer injects the fluid must be the same as the global supply temperature setpoint in the network. If the substation injects fluid in the supply network at a colder temperature, it could lower the overall supply temperature. This could be harmful to certain consumers, for example if the temperature falls below the safe DHW temperature, it may cause health issues as bacteria could develop in the water tank.

Of course the return temperature is also a constraint of the prosumer substation, as it cannot control the upstream temperature of the fluid arriving to him from the rest of the network.

The prosumer's third constraint comes from the sun. It is desired to inject all of the solar power available from the collector. This is in a goal to use as much renewable low-fossil-carbon energy as possible. Also it will save money in fuel since the energy is freely available to the network. If the consumer sells the heat to the network, it is also an advantage for him.

All these constraints on the supply temperature  $T_{\text{sup}}$  [°C], the return temperature  $T_{\text{ret}}$  [°C], the solar power to inject  $\dot{Q}_{\text{sol}}$  [W] determines the mass flow  $\dot{m}_{\text{des}}$  [kg/s] that the prosumer substation desires in the primary network

$$\dot{m}_{\text{des}} = -\frac{\dot{Q}_{\text{sol}}}{(T_{\text{sup}} - T_{\text{ret}})c_p}. \quad (3.55)$$

During the convergence of the system, the return temperature  $T_{\text{ret}}$  may change, and therefore the power available  $\dot{Q}_{\text{sol}}$  as well since it depends on the temperature of the fluid in the panel. This means that at each iteration, the mass flow desired is recomputed using Algorithm 5. After that, a positive  $\dot{m}_{\text{des}} \geq 0$  means that the valve is used, and a negative  $\dot{m}_{\text{des}} < 0$  means the pump is used (see Figure 3.13).

During the thermal computations to solve system, the algorithm to compute the heat exchanged at the prosumer substation who is feeding-in heat is explained in Algorithm 6. Note this algorithm is called when  $\dot{m}_{\text{des}} < 0$ , which in the code's convention means that the desired flow is from return to supply. The approximation is made that the temperature of the fluid entering and leaving the solar collector are the same as the temperatures entering and leaving the substation on the primary side:  $T_{\text{ret}}$  and  $T_{\text{sup}}$ . Usually there is an imperfect heat exchanger in between the primary circuit and the solar collector. The binary search in this algorithm is important because the thermal power available from the solar collector  $\dot{Q}_{\text{avai}}$  depends on the average fluid temperature in the collector, which is the average of the temperature entering and leaving the substation  $T_{\text{avg fluid}} \leftarrow \frac{1}{2}(T_{\text{sup}} + T_{\text{ret}})$ . But to compute the temperature leaving the substation  $T_{\text{sup}}$ , also depends on the available solar power  $\dot{Q}_{\text{avai}}$ . Therefore, a binary search is used to solve this.

To inject the desired solar mass flow  $\dot{m}_{\text{des}}$ , there are multiple possible control strategies that have been implemented and tested.

- Update the pump rotational speed  $n$  with aim to get  $\dot{m}_{\text{des}}$ . To do so use the "ideal" formula, of the rotational speed that the pump should have if it had the desired mass

---

**Algorithm 5:** Determine  $\dot{m}_{\text{des}}$  the mass flow desired by the prosumer.

---

**Function** *computeProsumerDesiredMassFlow*( $\dot{Q}_{\text{dem}}, f_{\text{sol left}}, T_{\text{ret}}, c_p, \text{climate}, \text{day}, \text{hour}$ )

```

if  $\dot{Q}_{\text{dem}} \neq 0$  then
  // The building demands heat from the substation.
  // Use regular consumer substation algorithm.
   $\dot{m}_{\text{des}} \leftarrow \text{Substation}::\text{computeDesiredMassFlow}(\dot{Q}_{\text{dem}})$ 
else
  // The building demands no heat from the substation.
  // Feed-in any available solar power.
   $T_{\text{sup}} \leftarrow \text{computeSetpointTemperature}(\text{climate}, \text{day}, \text{hour})$ 
   $T_{\text{avg fluid}} \leftarrow \frac{T_{\text{sup}} + T_{\text{ret}}}{2}$  // Average fluid temperature in solar collector.
   $\dot{Q}_{\text{sol}} \leftarrow f_{\text{sol left}} \cdot \text{computeSolarThermalPower}(T_{\text{avg fluid}}, \text{climate}, \text{day}, \text{hour});$ 
  if  $0 < \dot{Q}_{\text{sol}}$  and  $0.05 < T_{\text{sup}} - T_{\text{ret}}$  then
     $\dot{m}_{\text{des}} \leftarrow -\frac{\dot{Q}_{\text{sol}}}{(T_{\text{sup}} - T_{\text{ret}})c_p}$  // Negative => from return to supply.
  else
    // Don't inject heat, just run pump with a minimal flow.
     $\dot{m}_{\text{des}} \leftarrow \text{minimumDesiredPumpFlow}()$ 
  end
end
end

```

**end**

**Result:**  $\dot{m}_{\text{des}}$

---

flow under the current pressure

$$n_{\text{ideal}} = n_0 \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad a = -a_0, \quad b = -a_1 \dot{m}_{\text{des}}, \quad c = -a_2 \dot{m}_{\text{des}}^2 - \Delta p_{\text{current}} \quad (3.56)$$

Using the annealing method of section 3.2.13 to dampen the steps to help convergence.

- Update using a similar formula as the previous bullet point, but suppose that the rest of the system has a constant hydraulic resistance  $R = \Delta p_{\text{current}} / \dot{m}_{\text{current}}^2$ , so that at the desired mass flow the pressure drop around the rest of the system is  $\Delta p_{\text{const}R} = R \dot{m}_{\text{des}}^2$ . So use the annealing method of section 3.2.13 applied to the "ideal" formula

$$n_{\text{ideal}} = n_0 \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad a = -a_0, \quad b = -a_1 \dot{m}_{\text{des}}, \quad c = -a_2 \dot{m}_{\text{des}}^2 - \Delta p_{\text{const}R} \quad (3.57)$$

- To help control the small mass flows, a flow control valve is added right after the pump. The pump updates its rotational speed  $n$  in order to get a setpoint pressure  $\Delta p_{\text{setpoint}}$  around the substation, which preferably the same as the rest of the network producers. The valve updates its flow coefficient  $K_v$  to get the desired mass flow. They can both update using the "ideal" formulas along with the annealing damping of section 3.2.13. To update the valve, take the position that the valve should have to get the desired mass flow  $\dot{m}_{\text{des}}$ , supposing that the pressure drop across the substation (pump+flow control

---

**Algorithm 6:** Determine heat exchanged  $\dot{Q}_{\text{exchanged}}$  and the temperature supplied  $T_{\text{sup}}$  by the prosumer substation, with a given current mass flow and return temperature.

---

**Function** *computeProsumerHeatExchanged*( $c_p, \rho, T_{\text{ret}}, \dot{m}_{\text{cur}}, \text{climate}, \text{day}, \text{hour}$ )

```

 $T_{\text{setpoint}} \leftarrow \text{computeTargetTemperature}(\text{climate}, \text{day}, \text{hour})$ 
if  $T_{\text{setpoint}} \leq T_{\text{ret}}$  then
     $\dot{Q}_{\text{exchanged}} \leftarrow 0$ 
     $T_{\text{sup}} \leftarrow T_{\text{ret}}$ 
else
     $\dot{Q}_{\text{need}} \leftarrow -\dot{m}_{\text{cur}} c_p (T_{\text{setpoint}} - T_{\text{ret}})$ 
     $T_{\text{avg fluid}} \leftarrow \frac{1}{2} (T_{\text{setpoint}} + T_{\text{ret}})$  // Average fluid temperature in solar
        collector.
     $\dot{Q}_{\text{avai}} \leftarrow \text{computeSolarThermalPower}(T_{\text{avg fluid}}, \text{climate}, \text{day}, \text{hour})$ 
    if  $\dot{Q}_{\text{need}} \leq \dot{Q}_{\text{avai}}$  then
        // Sufficient solar power to reach the setpoint temperature.
         $\dot{Q}_{\text{exchanged}} \leftarrow \dot{Q}_{\text{need}}$ 
         $T_{\text{sup}} \leftarrow T_{\text{setpoint}}$ 
    else
        // Not enough solar power is available to reach the setpoint
            temperature, find supplied temperature with binary search.
         $T_{\text{sup,high}} \leftarrow T_{\text{setpoint}}$ 
         $T_{\text{sup,low}} \leftarrow T_{\text{ret}} + \frac{\dot{Q}_{\text{avai}}}{-\dot{m}_{\text{cur}} c_p}$ 
        do
             $T_{\text{out,mid}} \leftarrow \frac{1}{2} (T_{\text{sup,high}} + T_{\text{sup,low}})$ 
             $T_{\text{avg fluid}} \leftarrow \frac{1}{2} (T_{\text{sup,mid}} + T_{\text{ret}})$ 
             $\dot{Q}_{\text{avai}} \leftarrow \text{computeSolarThermalPower}(T_{\text{avg fluid}}, \text{climate}, \text{day}, \text{hour})$ 
             $T_{\text{sup}} \leftarrow T_{\text{ret}} + \frac{\dot{Q}_{\text{avai}}}{-\dot{m}_{\text{cur}} c_p}$ 
            if  $T_{\text{sup,mid}} > T_{\text{sup}}$  then
                 $T_{\text{sup,high}} \leftarrow T_{\text{sup,mid}}$ 
            else
                 $T_{\text{sup,low}} \leftarrow T_{\text{sup,mid}}$ 
            end
        while  $|T_{\text{sup}} - T_{\text{sup,mid}}| > 0.05$  and  $|T_{\text{sup,high}} - T_{\text{sup,low}}| > 0.05;$ 
         $\dot{Q}_{\text{exchanged}} \leftarrow \dot{Q}_{\text{avai}}$ 
    end
end
end

```

---

valve) stays the same as the current  $\Delta p_{\text{sub,current}}$ , and the pressure gain through the pump is the one with current rotational speed  $n_{\text{current}}$  and desired mass flow  $\dot{m}_{\text{des}}$

$$K_{v,\text{ideal}} = \dot{m}_{\text{des}} \sqrt{\frac{3600^2}{\rho_0 \rho} \frac{\Delta p_0}{\left[ a_0 \frac{n_{\text{current}}^2}{n_0^2} + a_1 \dot{m}_{\text{des}} \frac{n_{\text{current}}}{n_0} + a_2 \dot{m}_{\text{des}}^2 \right] - \Delta p_{\text{sub,current}}}}. \quad (3.58)$$

To update the rotational speed of the pump, take the ideal speed that the pump would need so that the current mass flow  $\dot{m}_{\text{current}}$  has pressure increased to get the setpoint pressure drop  $\Delta p_{\text{sub,setpoint}}$  across the total substation. Supposing that the control flow valve's current position  $K_{v,\text{current}}$  and the current mass flow stay the same, the pressure drop through the valve also stays the same as the current  $\Delta p_{\text{val,current}}$ .

$$n_{\text{ideal}} = n_0 \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad a = -a_0, \quad b = -a_1 \dot{m}_{\text{current}}, \quad c = -a_2 \dot{m}_{\text{current}}^2 + \Delta p_{\text{val,current}} + \Delta p_{\text{sub,setpoint}} \quad (3.59)$$

### 3.3.4 Thermal stations with seasonal storage

This section details further the computations and algorithms behind the thermal station with seasonal storage, that accompany the prosumer substations, to store the excess heat produced. A basic scheme is shown in Figure 3.14. In the C++ code, it is implemented by the class `SeasonalStorageHeatingThermalStation` which is a subclass of `ThermalStation`. It is possible to make simulations with either of these two classes.

#### Operation mode

The first the thermal station must now determine in which operation mode it is. There is the normal producer mode, where the pump is used to make liquid flow from return to supply and the fluid is heated with both the seasonal storage and the heating unit. And the new mode is when the prosumer substations are injecting too much flow, in which case the thermal station lets the flow run from supply to return through its valve, which makes it possible to get the sufficient pressure drop, and it stores the heat in the seasonal storage.

The function to determine which operation mode is used is shown in Algorithm 7. It sums all the desired mass flows of the substations. Prosumer substations that feed-in will contribute negatively. If the total is negative, then the storage mode is used, else the producer mode is used.

#### Update control variables

If the producer mode is on, the pump is used, and its rotational speed  $n$  needs to be updated to get the desired pressure setpoint  $\Delta p_{\text{setpoint}} = f(\dot{m}_{\text{current}})$  [Pa], which can depend on the current mass flow. This is done in the same way as a regular thermal station.

If the storage mode is on, the valve is used, and its flow coefficient  $K_v$  needs to be updated. There are multiple possible control strategies that have been implemented and tested.

- Update with aim to get the pressure difference setpoint  $\Delta p_{\text{setpoint}} = f(\dot{m})$ . To do so use

---

**Algorithm 7:** Determine for the thermal stations the mode to use.

---

**Function** *updateOperationMode(substations, thermalStations)*  
 flowSupplyToReturn ← 0  
**for** *s* **in** *substations* **do**  
 | totalFlow ← totalFlow + s.getDesiredMassFlow()  
**endfor**  
**for** *t* **in** *thermalStations* **do**  
 | **if** *flowSupplyToReturn* ≤ 0 **then**  
 | | t.turnStorageModeOn();  
 | **else**  
 | | t.turnStorageModeOff();  
 | **end**  
**endfor**  
**end**

---

the "ideal" formula, of the position that the valve should have to get the current mass flow with the desired pressure

$$K_{v,ideal} = \dot{m}_{current} \sqrt{\frac{3600^2}{\rho_0 \rho} \frac{\Delta p_0}{\Delta p_{setpoint}}}. \quad (3.60)$$

Using the annealing method of section 3.2.13 to dampen the steps to help convergence.

- Keep the valve at a constant position with no objective. It is considered to have converged. This static behaviour can help the prosumers to adapt more easily to impose their desired mass flows.
- The thermal stations add up all the mass flows that prosumers want to inject, and subtract all the mass flows that the consuming substations want to consume. This gives the total flow they should share. They then divide this load among all thermal stations with storage. For example divide equally, or proportional to the size of storage, etc. This defines a desired target mass flow  $\dot{m}_{target}$ . Then update the valve position with "ideal" formula of the position that the valve should have to get this mass flow under the current pressure  $\Delta p_{current}$  [Pa]

$$K_{v,ideal} = \dot{m}_{target} \sqrt{\frac{3600^2}{\rho_0 \rho} \frac{\Delta p_0}{\Delta p_{current}}}. \quad (3.61)$$

Use the annealing method of section 3.2.13 to dampen the step.

### Storage

In real life, different types seasonal storages exist and are used [20]. They can vary in price, efficiency and some can only be installed in specific locations, depending on the local geology. There are Tank Thermal Energy Storage (TTES) which in short are big tanks storing a fluid which will store the heat. Also there are Pit Thermal Energy Storage (PTES) in which the tank

is buried and can contain a water-gravel mixture to store the heat. Then the Borehole Thermal Energy Storages (BTES) and Aquifer Thermal Energy Storages (ATES) actually store the heat in the soil, heating up the ground. The BTES has vertical borehole heat exchangers which is a closed system, the fluid stays in the tube. The ATES is an open system, that sends the fluid in an aquifer which is used as a tank.

In this thesis a simple model for heat storage was implemented, but more complicated variations can be added in the future. the model is a completely isolated tank of fluid. Its total heat capacity is  $C_{sto}$  [J/K]. The fluid in the storage tank has a uniform temperature  $T_{sto}$  [K].

To exchange heat with storage, a flow of  $\dot{m}$  [kg/s] of fluid at temperature  $T_{in}$  [K], with specific  $c_p$  [J/(K kg)]. The fluid arriving mixes with the fluid in the storage tank. And a stream of the same mass flow  $\dot{m}$  exits the tank with a temperature of  $T_{out} = T_{sto}$ . This gives a differential equation on how the temperature of the storage  $T_{sto}$  should evolve in equation (3.62).

$$\dot{m}c_p(T_{in} - T_{sto}) = C_{sto} \frac{dT_{sto}}{dt} \quad (3.62)$$

This can be solve analytically, as shown in equation (3.63). Basically the temperature of the storage starts a initial  $T_{sto}(t = 0)$ , then exponentially decays to the  $T_{in}$ , the temperature at which the fluid is injected. The decay rate is  $\frac{\dot{m}c_p}{C_{sto}}$  [1/s]. Quite intuitively, the rate increases as the mass flow injected  $\dot{m}$  increases, but the rate decreases as the thermal capacity of the tank  $C_{sto}$  increases.

$$T_{sto}(t) = T_{in} + [T_{sto}(t = 0) - T_{in}] \exp\left(-\frac{\dot{m}c_p}{C_{sto}} t\right) \quad (3.63)$$

Since CitySim is discretized in time in steps of  $\Delta t$ , and the temperature cannot evolve continuously. The current goal not to determine what the temperature of the tank is after  $\Delta t$ , but to compute an approximate constant power  $\bar{Q}$  [W] given to the storage during that time step. This thesis' model approximates that a final temperature reached by the storage is  $T_{sto}(t = \Delta t) = T_{in} + [T_{sto}(t = 0) - T_{in}] \exp\left(-\frac{\dot{m}c_p}{C_{sto}} \Delta t\right)$ , so therefore the total energy given to the tank is  $C_{sto}[T_{sto}(t = \Delta t) - T_{sto}(t = 0)]$ . Meaning that the average power given is

$$\bar{Q} = \frac{C_{sto}[T_{sto}(t = \Delta t) - T_{sto}(t = 0)]}{\Delta t} \quad (3.64)$$

With this approximation of a constant heat exchanged, one can also approximate the temperature of the fluid exiting the tank  $\bar{T}_{out}$  as constant and equal to

$$\begin{aligned} \bar{T}_{out} &= T_{in} - \frac{\bar{Q}}{\dot{m}c_p} \\ &= T_{in} - \frac{C_{sto}}{\dot{m}c_p \Delta t} [T_{sto}(t = 0) - T_{in}] \left[ \exp\left(-\frac{\dot{m}c_p \Delta t}{C_{sto}}\right) - 1 \right] \end{aligned} \quad (3.65)$$

Once the system has converged for the given time step, the temperature of the storage is updated with

$$T_{\text{sto}}(t = \Delta t) = T_{\text{in}} + [T_{\text{sto}}(t = 0) - T_{\text{in}}] \exp\left(-\frac{\dot{m}c_p}{C_{\text{sto}}} \Delta t\right) \quad (3.66)$$

Of course this heat exchange is only performed if  $T_{\text{in}} > T_{\text{sto}}$  when the thermal station is in storage mode, or if  $T_{\text{in}} < T_{\text{sto}}$  when the thermal station is in heat production mode. Otherwise, no heat is exchanged, for example by deviating the flow through a bypass, and  $\bar{T}_{\text{out}} = T_{\text{in}}$ .

When the thermal storage is in production mode, it uses the storage with this kind of heat exchange to possibly preheat the fluid.

### 3.3.5 Overall Control

A new version of Algorithm 2 is required as shown in Algorithm 8. During producer mode, the prosumers must adapt the desired mass flow to the temperatures at its return node. This means that there objective changes, so it must be updated at each step. Similarly this may affect the operation mode of the thermal stations, which depends on the sum of the desired mass flows of substations.

---

**Algorithm 8:** Overall loop to converge to the valve flow coefficients  $K_v$  and pump rotational speeds  $n$  of the stationary solution where all targets are satisfied. Similar to Algorithm 2, but adapted for prosumer substations.

---

```

Initialize  $k = 0$ , notConverged=true
updateProsumerDesiredMassFlow()
updateThermalStationOperationMode()
while  $k < k_{\text{max}}$  and notConverged do
    // Update control variable positions.
    valves.updateKv()
    pumps.updateN()
    // Converge the system with the new control variables.
    convergeHydraulic()
    convergeThermal()
    // Update the desired mass flows and operation modes.
    updateProsumerDesiredMassFlow();
    updateThermalStationOperationMode();
    // Check for convergence.
     $\vec{e}_{\text{rel}} = \text{computeRelativeError}(\text{substations}, \text{thermalPowerPlants})$ 
    notConvergence = ( $\max_i e_{\text{rel},i} \geq 0.02$ )
     $k = k + 1$ 
end

```

---

## 3.4 Miscellaneous contributions

Additions miscellaneous codes and scripts have been implemented. during this thesis

The algorithm to distribute the solar thermal energy among the heat storage tank and the domestic hot water tank has been modified. The idea is to establish a priority among the two. Since the efficiency of the solar collector decreases as the temperature of the fluid to heat increases, the priority tends to favour the space heating tank, which is colder. Also has to be taken into account the fact that the space heating systems are turned on only during a user determined set of days  $\mathcal{D}_{\text{heating days}}$ , usually during the winter. In section 2.1, it is explained that the heating unit only heats a tank when it goes below its lower boundary  $T_{\text{min}}$  [°C] and brings it to an upper boundary  $T_{\text{max}}$  [°C]. The solar thermal can be used to heat past this upper boundary if sufficient solar thermal power is available, however, there is a critical temperature that must not be exceeded  $T_{\text{crit}}$  [°C], for example around 90°C to protect the material from being damaged by overheating. If the solar thermal reaches this critical temperature, it stops heating that tank, and computes the solar fraction left  $f_{\text{sol left}}$  [-], which can be seen as the fraction of the time step left where solar thermal power is unused, as explained in section 3.3.3. It then gives the rest of the solar thermal fraction to the non-prioritised tank. If this tank also reaches the critical temperature, it computes the solar fraction still left, to potentially inject into the district heating network. Algorithm 9 describes how the priority is set between the heat storage tank (HS) and the domestic hot water tank (DHW).

---

**Algorithm 9:** Instructions to determine which tank has priority to benefit from the solar thermal power.

---

```

if  $\dot{Q}_{\text{space heating needs}} \leq 0$  or  $\text{day}_{\text{current}} \notin \mathcal{D}_{\text{heating days}}$  then
  |  $b_{\text{prioritise DHW}} \leftarrow \text{true}$ 
else
  |  $T_{\text{next HS}} \leftarrow \text{computeTempAtEndOfTimeStepWithoutSolarPowerAdded}(\text{tank}_{\text{HS}})$ 
  |  $T_{\text{next DHW}} \leftarrow \text{computeTempAtEndOfTimeStepWithoutSolarPowerAdded}(\text{tank}_{\text{DHW}})$ 
  | if  $T_{\text{next HS}} > T_{\text{min HS}}$  and  $T_{\text{next DHW}} < T_{\text{min DHW}}$  then
  | |  $b_{\text{prioritise DHW}} \leftarrow \text{true}$ 
  | else
  | |  $b_{\text{prioritise DHW}} \leftarrow \text{false}$ 
  | end
end

```

**end**

**Result:**  $b_{\text{prioritise DHW}}$

---

There have also been scripts implemented to determine building thermal properties. As input, it takes basic statistical data of buildings such as the year of construction, of renovation, the type of activity in the building, the ground footprint etc. It uses these information to approximate the infiltration rate, the insulation materials of walls and roofs, window percentage of the walls, and so on. This is done using the results from Perez [38]. This can be used to directly create an input file in a format compatible with CitySim. It is useful to make the input files to simulate real case studies with numerous buildings.

## 4 Results and discussions

The objective of this section is to make sure that the software and models developed in section 3 produce results with sufficient accuracy and similarities with reality. This way, it can be used to answer the problematic brought in section 1.3. More specifically the importance is to validate, verify and compare the models added to the CitySim software by this thesis, that is to say, the district network simulations.

Sources of error can come from the models themselves. The models can over-simplify reality, in a way that real-life effects cannot be recreated.

Errors can also come from the discretisation. Computers cannot model continuous reality, so a discretisation process is necessary and there is a possibility to overlook physical realities. In the present case, the discretisation in time, with steady-state solutions found at each iteration, limits the possibility to model the non-equilibrium phenomena, for example when consumers change rapidly their consumption, meaning that the thermal station temporarily doesn't produce the correct input temperature in the network as the boiler takes time to adapt.

Errors can come from the algorithms that are used to solve certain system of equations. In the present case, this applies for example to the Newton method described in section 3.2.10. Under certain circumstances, they can not converge to the correct solution. Possibly because the system is too complex or non-linear, or the machine precision is not sufficient in the case of an ill-conditioned problem.

Programming such a complex model can also lead to errors or bugs. There can be typing mistakes, special cases that have not been thought of, and so on. This calls for many verifications of the multiple modules, possibly separating them individually to check their behaviors and detect problems.

In computational science and engineering, the validation is to check whether the mathematical model and equations are good enough to recreate real physical events or effect. The verification process is to make sure that the program implementation of the mathematical model is done with enough precision [41].

To validate and verify the model, there are multiple methods [36] :

- Comparing to measured data. This is one of the best and most desirable method to test against a real life district network. Unfortunately it can be complicated to get good data. Private companies operating the network have sometimes no use in taking frequent measurements of different variables of the network. And even less interest in making it

available for research. It can also cause confidentiality problems, if they do not want to show the performance of the network they operate. Also confidentiality issue can arise from the private nature of the heat consumption data of the users. Mostly difficulties can come from the reliability, quality and completeness of the measurements. For practical issues there can be missing data, if apparatus has stopped working for a certain period of time, or if the information is so old that it got lost over time (the depth of a pipe). Also it can be inaccurate if a cheap sensor is used for non-precise applications, or if a sensor is poorly calibrated.

- Plausibility check. This can involve checking whether the results contradict theories considered as certain. For example checking that the energy is overall conserved in the system. This approach can be useful to detect errors in the implementation of the program. But it is limited as it doesn't check specifically the system modelled, only certain global aspects.
- Comparing against other higher class models or exact mathematical computations. In the present case it will be too complicated to compare the whole network to a perfect mathematical model. The computations required are often too complicated. This can be applied to small parts of the problem, such as a toy network with few components. Other higher class models for this applications are not easily available and implementable, or compatible.

For this thesis, a comparison with measured data in the district heating network of Broc, Switzerland has been made.

## 4.1 Branched network Broc case study

Broc is a city in the Fribourg canton of Western Switzerland. It is located at the upstream end of the Lake of Gruyère, at the foot of the Alps and is famous for its chocolate factory. As Figure 4.1 shows, the village stands at the top of a hill.

Our interest is the Broc district heating network, which is plotted in Figure 4.2. This network is branched, so it doesn't have a mesh topology with any loop. Most of the network is flat, located on top of the hill at an elevation of about 715 meters above sea level. Although the thermal station is located lower, near the river, at an altitude of about 680 meters.

### 4.1.1 Network geographic data

The data obtained to describe the geography and topology of the graph was obtained and formatted by Giona Galizia, who has been of great help and support during this thesis. The information is stored in SQL type databases.

A table in the database described the "Network Points" as shown in Figure 4.3. This corresponds best to what is referred in this thesis by a node (the half of a node pair). In this representation, the network point can be a simple junction between two pipes, either a straight, or an elbow of  $90^\circ$  or  $135^\circ$ . The points can also joint three pipes, usually in a T joint. Finally points can also be end points of the network, such as the entry to a thermal station, a

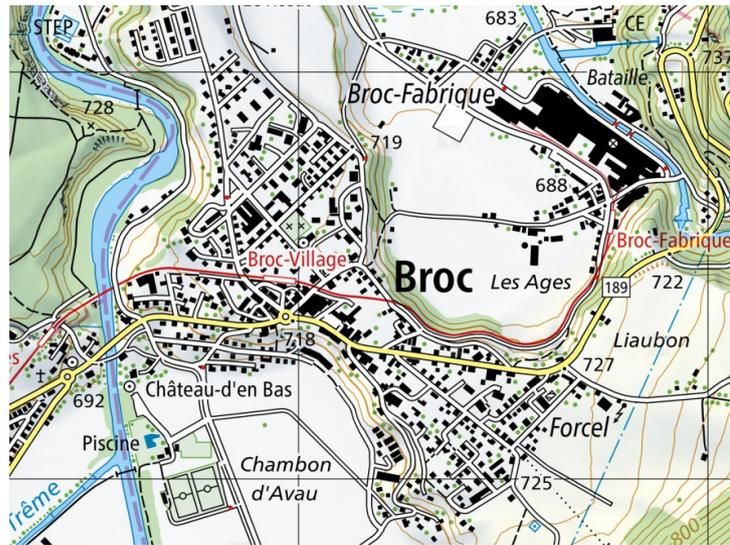


Figure 4.1 – Topographic map of Broc [51].



Figure 4.2 – Layout of Broc district heating network. The thermal station is the green building at the top left of the map. Plot made using QGIS software [39].

to a consumer's heat exchanger (HX). The table contained a unique id (npid), the geometry in GIS format, the "warmth", which indicates if it is in the supply or return pipeline network, the type, the altitude, the longitude (X), the latitude (Y). In Figure 4.3, the temperature ("temp") and pressure columns are results from another simulation model called Eguzky, but were not used apart from checking the overall sanity of the simulation results.

| npid       | geom                     | warmth | type                  | temp    | pressure | altitude | X           | Y           |
|------------|--------------------------|--------|-----------------------|---------|----------|----------|-------------|-------------|
| A.39.44    | 0104000020E6100000010000 | warm   | HX                    | 83.9084 | 432.015  | 720      | 7.100421497 | 46.60715284 |
| A.39.31.1  | 0104000020E6100000010000 | warm   | elbow 90°             | 84.6569 | 440.427  | 718.9    | 7.100407223 | 46.60663838 |
| A.39.31.3  | 0104000020E6100000010000 | warm   | straight              | 84.654  | 440.382  | 718.78   | 7.100412259 | 46.60665747 |
| A.39.31.4  | 0104000020E6100000010000 | warm   | straight              | 84.6413 | 440.304  | 718.82   | 7.100361101 | 46.60668741 |
| A.75.5.1   | 0104000020E6100000010000 | warm   | straight              | 84.5074 | 437.157  | 718.19   | 7.098279327 | 46.60371208 |
| A.39.14.14 | 0104000020E6100000010000 | warm   | straight              | 84.7091 | 441.398  | 717.9    | 7.099392075 | 46.60725658 |
| A.1        | 0104000020E6100000010000 | warm   | start heating station | 85      | 536.418  | 682.38   | 7.094499082 | 46.60573539 |

Figure 4.3 – Database table containing information of the Broc network points.

The next important table used contains the pipe information as shown in Figure 4.4. The columns used are the identification number (pid), the two network points that it connects (startpoint and endpoint), the length and the nominal diameter (dn). The nominal diameter is a European convention to describe the size of pipe diameters, a table describing the linear conversion to millimeters can be found at [33]. In Figure 4.4, the "dn" column contains two numbers separated by a slash. The first number is the inner diameter of the pipe. The second number is the outer diameter, taking into account the insulation thickness. A small script extracted all this information to put it in the XML format compatible with the CitySim software developed in this thesis.

| pid | geom                   | startpoint | endpoint | length   | dn      | material | pressure_head | flow_rate | delta_t | flow_speed | T-Straight |
|-----|------------------------|------------|----------|----------|---------|----------|---------------|-----------|---------|------------|------------|
| 602 | 0105000020E61000000100 | A.1        | A.2      | 15.62603 | 150/315 |          | 142.772       | 125.73    | 0       | 1.73053    | False      |
| 657 | 0105000020E61000000100 | A.2        | A.3      | 17.78091 | 150/315 |          | 142.772       | 125.73    | 0       | 1.73053    | False      |
| 803 | 0105000020E61000000100 | A.4        | A.5      | 2.39938  | 150/315 |          | 142.772       | 125.73    | 0       | 1.73053    | False      |
| 853 | 0105000020E61000000100 | A.5        | A.6      | 9.873019 | 150/315 |          | 142.772       | 125.73    | 0       | 1.73053    | False      |
| 603 | 0105000020E61000000100 | A.10       | A.11     | 1.188506 | 150/315 |          | 142.772       | 125.73    | 0       | 1.73053    | False      |

Figure 4.4 – Database table containing information of the Broc pipes.

The "Heating Stations" table contained the information about the only thermal station of this network as shown in Figure 4.5. The "Substations" table contains information on the substations linked to the network as shown in Figure 4.6. Not all information is needed for our purposes. Among the most useful are the construction year, to help make estimations for the building thermal performances (e.g. insulations types), and size of the heat exchanger of the substation.

| chsid | geom                     | t_go | t_return | power_tot | static_p | dynamic_p | t_delta |
|-------|--------------------------|------|----------|-----------|----------|-----------|---------|
| 1     | 010600002015550000010000 | 85   | 55       |           |          |           | 25      |

Figure 4.5 – Database table containing information of the Broc heating station.

| id      | geom | coverage | ssid      | height | egid | altitude | infiltration_rate | gbau | volume | occupancy | type | n_floors | blinds    | lambda   | blinds | radiance | cecutoff | nopeople | zone   | NPID | in | NPID | out | walltype | id | heating_year | max_heating_hour | Kvs | heating_hour | temp |
|---------|------|----------|-----------|--------|------|----------|-------------------|------|--------|-----------|------|----------|-----------|----------|--------|----------|----------|----------|--------|------|----|------|-----|----------|----|--------------|------------------|-----|--------------|------|
| 120106* | 1279 | BCA*     | 0.8402475 | 0      | 1.01 | 1950     | 2000              | 1    | 3      | 0.017     | 300  | 5        | 1A.78.3   | R.78.3   | 2      | 34000000 | 17000    | 1.6      | 25000  |      |    |      |     |          |    |              |                  |     |              |      |
| 940106* | 770  | BCA*     | 0.4840257 | 0      | 1.01 | 1950     | 2000              | 1    | 3      | 0.017     | 300  | 5        | 1A.35.9   | R.35.9   | 2      | 28500000 | 157350   | 4.9      | 100000 |      |    |      |     |          |    |              |                  |     |              |      |
| 340106* | 966  | BCA*     | 0.4759329 | 0      | 1.01 | 1950     | 2000              | 1    | 3      | 0.017     | 300  | 5        | 1A.38.9.6 | R.38.9.6 | 2      | 42000000 | 21000    | 1.6      | 25000  |      |    |      |     |          |    |              |                  |     |              |      |

Figure 4.6 – Rows from the database table containing information of the Broc substations (anonymised data, this is an example).

In this thesis' representation of a network, there is no need for the nodes that connect two pipes with a straight joint or  $90^\circ$ . This would be useful if the punctual charge losses were computed in these areas, but this isn't currently the case. Therefore a small script has been implemented to detect and delete such nodes, and to fuse the two consecutive pipes into one with a combined length. If the diameters changed, the two pipes were kept separate.

For the branched version of the code, there was a need for a script to determine the direction of each pipe. The algorithm starts from the thermal station and propagates outward to determine which end of the pipe goes towards the heating station.

The information on the operational control of the thermal station was obtained by the work of Jason Galley [16]. These control settings were taken on April 26 2019, but they rarely significantly change, if they are altered. Figure 4.7 shows how the temperature that the thermal station injects into the district network is determined. It is a continuous function linear by parts of the exterior temperature averaged over the last 24 hours. The general idea is that if the exterior temperature is decreases, it is colder outside, so there will be more heat losses in pipes, and consumers need more heat, so to make sure that all needs are satisfied, the supply temperature is increased. There is also a minimum and maximum supply temperature : above  $15^\circ\text{C}$  of exterior temperature, the supplied temperature setpoint stays constant at  $75^\circ\text{C}$  and below  $-10^\circ\text{C}$  of exterior temperature, the supplied temperature setpoint stays constant at  $90^\circ\text{C}$ . The exterior temperature is averaged over the last 24 hours to avoid frequent changes and smooth out the setpoint temperature. Specifically it will average out the day-night oscillation, which can have quite a big amplitude. This setpoint using Figure 4.7 is the operation mode during winter. There is also an operation mode in summer which is a constant supply temperature of  $75^\circ\text{C}$ . The system changes from winter to summer mode when the average exterior temperature over the last 24 hours passes above  $16^\circ\text{C}$ . And it changes from summer to winter mode when the average exterior temperature over the last 24 hours passes below  $5^\circ\text{C}$ . This hysteresis ensures that the during the spring and autumn the transition between the two modes occurs once, and that it doesn't switch back and forth between the two. This setpoint will determine the power asked to the heat producer, which in this case are wood and gas boilers.

Figure 4.8 shows how the setpoint of the differential pressure is chosen. This controls the power supplied to the pump that makes the fluid go in the district network. The differential pressure is function linear by parts of the mass flow averaged over the last 15 minutes. Again the average is to smooth out the fluctuations, while still being an appropriate reaction time to a changing demand from the consumers. The general idea is that if the consumers need more heat, they open the valves to demand more flow. In this case the mass flow through the thermal station increases, to make sure the consumers get as much flow as needed, the differential pressure is increased, to overcome the increasing pressure losses (which are generally proportional to the

square of the mass flow in pipes).

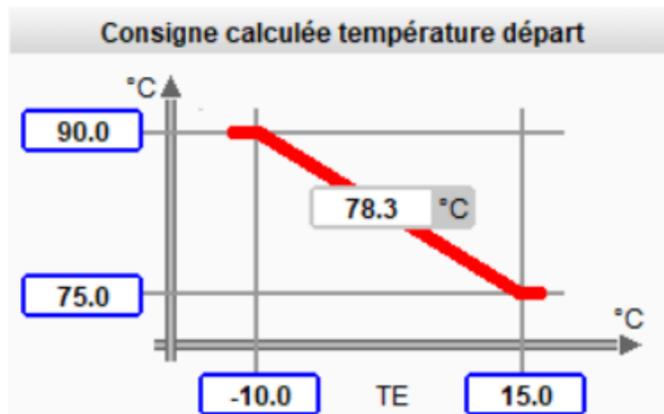


Figure 4.7 – Setpoint for the temperature inputted at the supply of the Broc district network [16]. This temperature is a function linear by parts of the exterior temperature averaged over the last 24 hours.

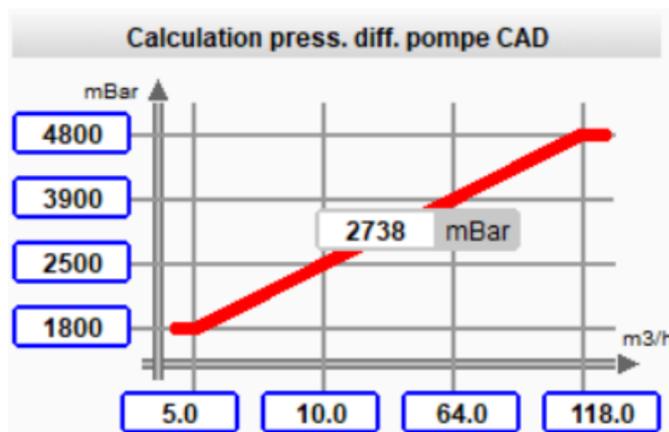


Figure 4.8 – Setpoint for the differential pressure in the Broc thermal station [16]. The differential pressure is a function linear by parts of the mass flow that the consumers are taking.

These two types of setpoints are implemented in the CitySim program, in a more general version, in order to have any linear by parts function of exterior temperature or mass flow.

#### 4.1.2 Exterior temperature

To improve the simulation results, it is important to have the correct parameters such as the exterior temperatures. The exterior air temperature is used in the computation of the setpoint supply temperature, and the surface temperature of the soil is used for the computations of thermal losses in the pipes.

For the exterior air temperature, two sources were available. The first source is a Meteonorm file [3] which has air temperature for a given year along with multiple other variables such as soil temperature. The second source is are files from the Broc district network thermal station,

that contains the recorded exterior temperature. Figure 4.9 shows the comparison of the two sources. They differ slightly as the Meteonorm file is not a direct measure of the temperature in Broc, but it is actually an interpolation based on the temperature measurements of nearby cities, with the help of a weather model. The supply setpoint temperature is computed using the temperature measured at the thermal station, not the Meteonorm file. So for future purposes, the temperature in the CitySim climate file was changed, it does not take the Meteonorm value anymore, but the measured value at Broc thermal station.

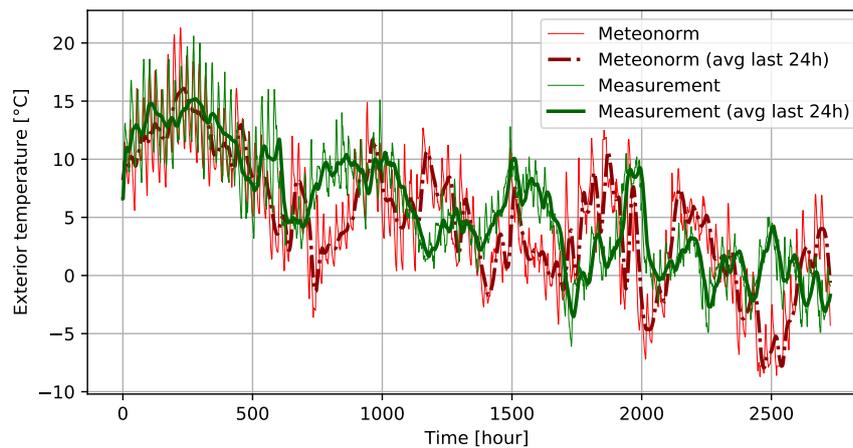


Figure 4.9 – Comparison of the exterior temperature from October 2<sup>nd</sup> 2018 to January 23<sup>rd</sup> 2019, between the Meteonorm [3] climate file and the measured data at the Broc thermal station.

### 4.1.3 Substation consumption

In the data received from the Broc network, there are the substation consumption from October 2<sup>nd</sup> 2018 to January 23<sup>rd</sup> 2019. This information is of great importance. CitySim can predict the heating and cooling needs of the buildings, and this is an important feature that has motivated the development of this software. However, the goal of this thesis is to add the simulation of district network. During this validation and verification, it is important to isolate only the simulation of the network, to measure its errors, not to couple it with the errors of the building thermal models. To achieve this goal, it is decided to impose in CitySim the heating consumption values of the buildings. This functionality has therefore been added to the CitySim program : when the `Building` class asks its heating unit for a given power, it verifies if an imposed value exists, which must have been inputted in the XML file. If such an imposed value exists, it asks the heating unit this value.

Figure 4.10 shows the table containing the consumption information. The top row has incremental numbers which stand for the hours of the day. For example, the column with a "2", stands for measurements made at 2 o'clock in the morning. The first column contains headers. The "T Ext" row describes the measured exterior temperature in degree Celsius at the thermal station. The "T CAD" row corresponds to the temperature in degree Celsius inputted

to the district network supply pipes (in French CAD stands for "Chaleur à distance", which translates to district heating). The following rows contain in the first cell the id number of a substation, and the rest are cumulative heat consumption in kWh. The value in itself can be very big, as it probably is the total heat consumption of that substation since it was first turned on many years ago. The interesting information is how this value increases as it describes the consumption.

|       | 0            | 1            | 2            | 3            | 4            | 5            | 6            | 7            |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| T Ext | 5.2          | 5.2          | 4.9          | 4.9          | 4.6          | 4.6          | 4.9          | 5.2          |
| T CAD | 80.300003    | 80.599998    | 80.599998    | 80.699997    | 80.400002    | 80.699997    | 80.699997    | 80.5         |
| 20    | 0            | 0            | 0            | 0            | 0            | 0            | 0            | 0            |
| 21    | 0            | 0            | 0            | 0            | 0            | 0            | 0            | 0            |
| 22    | 121965.00781 | 121965.00781 | 121965.00781 | 121965.00781 | 121965.00781 | 121965.00781 | 121965.00781 | 122068.00781 |
| 23    | 852974.0625  | 853095.0625  | 853095.0625  | 853095.0625  | 853215       | 853215       | 853215       | 853346.0625  |
| 24    | 768412.125   | 768513.0625  | 768513.0625  | 768513.0625  | 768614       | 768614       | 768614       | 768714.0625  |
| 25    | 51368.003906 | 51368.003906 | 51368.003906 | 51368.003906 | 51470        | 51470        | 51470        | 51470        |

Figure 4.10 – Table containing data of the Broc district network at different hours. Each column contains the measurement for a specific hour. The "T Ext" row gives the exterior temperature in degree Celsius. The "T CAD" column gives the temperature inputed into the supply pipes by the thermal station. The other rows each correspond to a substation. The first cell contains its ID, and the rest contain the cumulative energy consumed in kWh.

It can be noticed however that this value tends to stay constant multiple hours in a row, then makes jumps of approximately 100 kWh in one hour. This is probably linked to the fact the the measurement equipment can only count in increments of this big value. This is unfortunate as it decreases the quality of this data. To overcome this, these time series are smoothed, using a moving average. The window length of the moving averaged is determined differently for each building. Some substations consume much more power than others, meaning that the increment in 100 kWh takes place more often, so the smoothing window size can be smaller. The smoothing window size for a substation is computed by taking the maximal number of consecutive hours where no increment in consumption take place, multiplied by 2 (to get a good enough smoothing).

This smoothing actually means that the daily variations of the consumption of substations is lost. The effect is particularly strong for small substations that consume less. To get the power consumed there is still the need to perform a numerical derivative of this cumulated consumption curve.

In CitySim, the simulation of buildings requires data on structure, materials used, number of occupants, etc. This data is complicated to obtain, and was not available. Using approximations for this missing information, the demands can be estimated by CitySim. There can be however quite a significant difference between the predicted heat demand and the measured consumption data. For example, the church of Broc probably doesn't apply a 20°C temperature in the whole volume, explaining why CitySim overestimates its consumption. There are indeed differences due to approximations in evaluating missing data on buildings. Figures 4.11, 4.12, 4.13, 4.14, 4.15 and 4.16 show some comparisons between the CitySim demand prediction and the measurements. One can notice the power consumption in the measurements varies less, this is because the data had to be smoothed due to the 100kWh precision of measurements.

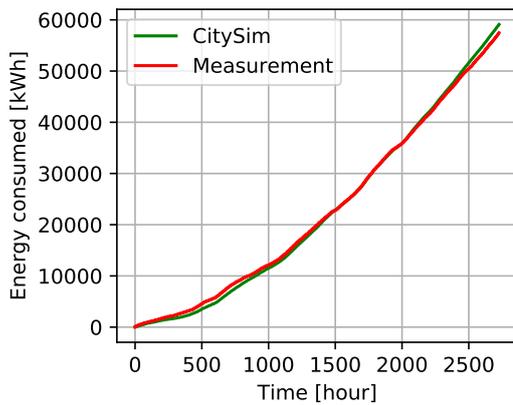


Figure 4.11 – Time evolution of cumulated energy consumption of building 74, comparison between the measured data and the CitySim prediction. The hour zero corresponds to October 1<sup>rst</sup> at midnight. CitySim predicts well the cumulated energy consumption.

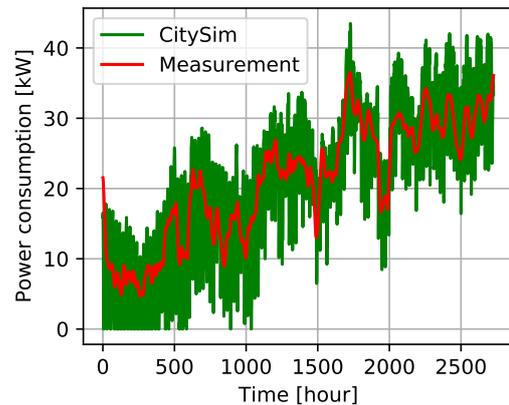


Figure 4.12 – Time evolution of power consumption of building 74, comparison between the measured data and the CitySim prediction. The hour zero corresponds to October 1<sup>rst</sup> at midnight. The power from measurements had to be smoothed, but they agree well with the CitySim prediction.

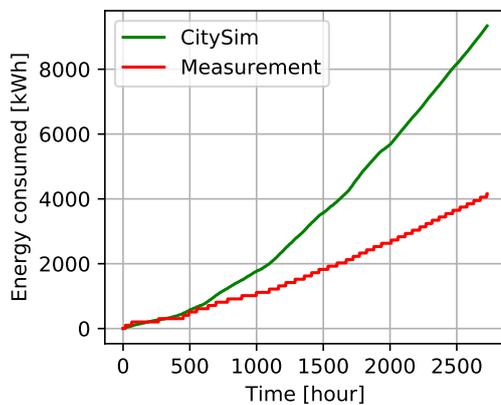


Figure 4.13 – Time evolution of cumulated energy consumption of building 29, comparison between the measured data and the CitySim prediction. The hour zero corresponds to October 1<sup>rst</sup> at midnight. CitySim over-predicts the energy consumption by a factor 2. This is a small consumer, so the measurement steps of 100kWh can be seen.

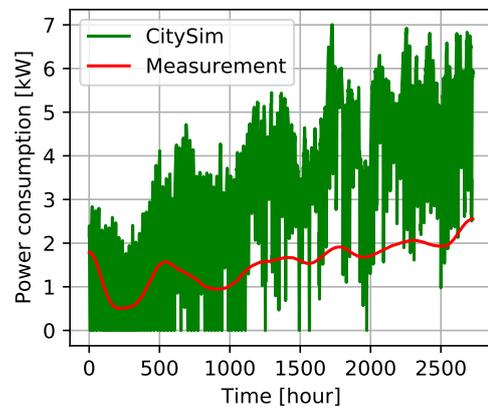


Figure 4.14 – Time evolution of power consumption of building 29, comparison between the measured data and the CitySim prediction. The hour zero corresponds to October 1<sup>rst</sup> at midnight. CitySim over-predicts the energy consumption by a factor 2. This is a small consumer, so derivative of the measurement steps of 100kWh can be seen as the measured curve doesn't oscillate much.



Figure 4.15 – Time evolution of cumulated energy consumption of building 53, comparison between the measured data and the CitySim prediction. The hour zero corresponds to October 1<sup>st</sup> at midnight. CitySim over-predicts the energy consumption by a factor 2.

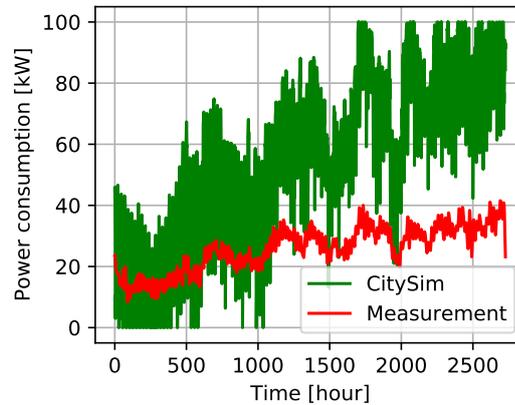


Figure 4.16 – Time evolution of power consumption of building 53, comparison between the measured data and the CitySim prediction. The hour zero corresponds to October 1<sup>st</sup> at midnight. CitySim over-predicts the energy consumption by a factor 2.

This shows the importance of having correct building data to estimate the thermal demand. Detailed data in the correct form is often difficult to obtain since it requires time to harvest, time to process, it comes with privacy agreements, and so on.

To be precisely simulated, the exact mechanism of the system operating the substation would need to be known. More specifically, the consumption values have been imposed, but there is still a degree a freedom in the way to get a given power. The power consumed  $\dot{Q}$  [W] depends on the mass flow passing through the primary side  $\dot{m}$  [kg/s] and the temperature difference on the primary side  $\Delta T$  [K], such that  $\dot{Q} = \dot{m}c_p\Delta T$ . The degree of freedom comes from the possibility to increases the mass flow  $\dot{m}$ , and lower the temperature difference  $\Delta T$ , or vice versa. This specific information for all individual substations has not been found precisely. The general instructions for the overall network are that old buildings, meaning those connected a long time ago, should try to aim for a  $\Delta T = 20$  K. On the other hand, newer buildings should aim for  $\Delta T = 30$  K. In practice this is not rigorously followed. In substations the PID controller do not specifically aim for this goal, but rather its objective is that the hot water on the supply side has a given temperature. The information was not found how precisely the mass flows in primary and secondary networks are changed to achieve this.

#### 4.1.4 Thermal station measurements

To evaluate how our simulation model performs compared to this real life case study, ideally one would compare the measurements of temperature and pressure at many locations in the network, to see the differences. In the present case the data on substation energy consumption have been used as an input to the CitySim simulation as explained in Section 4.1.3. The other data available concerns the thermal station in the form of a table as shown in Figure 4.17. The table contains the supply temperature  $T_{\text{sup}}$ , the return temperature  $T_{\text{ret}}$ , the thermal power

| Datum/Uhrzeit  | WCB r : F1 Mesure de température RL Istwert |    | WCB r : F1 Mesure de température VL Istwert |    | WCB r : F1 Waermezaehler Leistung |    | WCB r : F1 Waermezaehler Volumenstrom |                   |
|----------------|---|----|---|----|-----------------------------------|----|---------------------------------------|-------------------|
| 10/2/2018 0:23 | 58.39                                       | °C | 75  | °C | 229.92                            | kW | 12.73                                 | m <sup>3</sup> /h |
| 10/2/2018 1:23 | 59.71                                       | °C | 74.96                                       | °C | 239.32                            | kW | 13.21                                 | m <sup>3</sup> /h |
| 10/2/2018 2:23 | 58.45                                       | °C | 75.36                                       | °C | 239.28                            | kW | 12.67                                 | m <sup>3</sup> /h |
| 10/2/2018 3:23 | 59.63                                       | °C | 75.05                                       | °C | 229.83                            | kW | 13.29                                 | m <sup>3</sup> /h |
| 10/2/2018 4:23 | 58.43                                       | °C | 75.21                                       | °C | 243.63                            | kW | 13.95                                 | m <sup>3</sup> /h |
| 10/2/2018 5:23 | 56.91                                       | °C | 75.31                                       | °C | 320.43                            | kW | 14.59                                 | m <sup>3</sup> /h |
| 10/2/2018 6:23 | 56.74                                       | °C | 74.64                                       | °C | 321.32                            | kW | 14.9                                  | m <sup>3</sup> /h |

Figure 4.17 – Table of measurements of the Broc thermal station. The first column contains the time stamps. The second and third columns contain respectively the return and supply temperatures of fluid arriving at and leaving the thermal station in degree Celsius. The fourth column contains the thermal power provided to the fluid in kiloWatts. Finally the fifth column contains the volume flow of fluid in m<sup>3</sup>/h.

produced  $\dot{Q}$  and the fluid volume flow  $\dot{V}$ . The CitySim simulation tool gives results of mass flows in kilograms per second, the conversion can be made using the formula

$$\dot{m} = \frac{3600}{\rho} \dot{V}. \quad (4.1)$$

It can be noted that in this table, data is missing between December 7<sup>th</sup> 2018, 8:23 AM until January 21<sup>st</sup> 2019 at 9:23 AM. The table also contains some series on differential pressure. However these are not measurements, they are only an attempt of reconstructing the pressure using an affine function of the volume flow. It is therefore used as a means of plausibility check, to see if the orders of magnitudes are correct, but not as a truly measured data.

### Supply temperature

Figure 4.18 shows the comparison between the measured temperature supplied to the network and the result of the CitySim simulation. One can observe that at the start, they are in summer mode, since it is October 2<sup>nd</sup> and they seem to follow the 75°C setpoint. Then near hour 700, the pass to winter mode, following the setpoint defined by Figure 4.7.

It can be observed that the measurements oscillate much more than the simulation. Indeed, the CitySim simulation always manages to have exactly the setpoint temperature. This is because the simulation converges to a steady state. The boiler is sufficiently powerful and can change instantaneously from minimum to maximum power. This means that it can always reach exactly the setpoint temperature. On the other hand, the measurements oscillate more. This is because the real thermal station has to deal with the changing consumptions of the substations, so there is some inertia in the heating unit which means that it doesn't always perfectly reach the setpoint. It is noted however that the 24 hour average of the measurement curve is indeed quite close to the setpoint value. The average difference between the CitySim and the measurement curves is 0.37°C. The root mean square of the difference is 1.50°C.

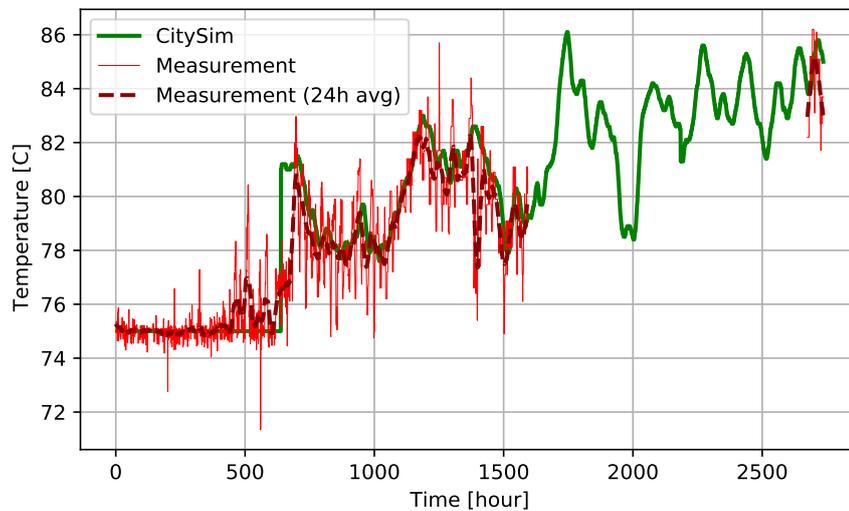


Figure 4.18 – Time evolution of supply temperature injected by thermal station. Comparison between measurements and CitySim simulation. Start on October 2<sup>nd</sup> and finish on January 23<sup>rd</sup>. Measurements have missing values between December 7<sup>th</sup> and January 21<sup>st</sup>.

Since the exact algorithm for the setpoint temperature is known, the simulation on average is very similar to reality. But it doesn't reproduce the small oscillations due to the thermal station trying to correct its error.

### Return temperature

Figure 4.19 shows the comparison between the measured return temperature entering the thermal station and the result of the CitySim simulation. First one can observe that the return temperature oscillates more than the supply temperature of Figure 4.18. The return temperature measured oscillate with an amplitude mostly above 6°C, while the supply temperatures measured oscillate with an amplitude mostly smaller than 4°C. This is because unlike the supply, the return temperature has no setpoint, it depends on how the consumers consume. Both the 24 hour averaged measured data and the simulation results start similarly up to roughly 400 hours. After that the difference is larger. The average difference between the CitySim and the measurement curves is 2.00°C. The root mean root mean square of the difference is 4.36°C.

The cause for these differences may come from the fact that the exact algorithm for the substations on how they extract the thermal power is not known. As said in section 4.1.3, to extract a given power, there is a degree of freedom : increase the flow and decrease the temperature difference, or vice versa. So it is normal to have differences in return temperature.

### Mass flow

Figure 4.19 shows the comparison between the measured flow through the thermal station and the result of the CitySim simulation. Similarly to the return temperature, a clear difference can

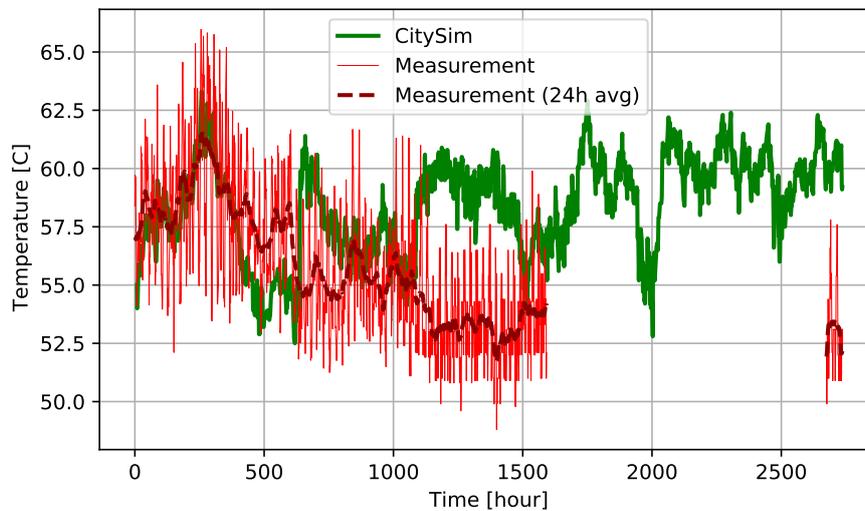


Figure 4.19 – Time evolution of return temperature entering the thermal station. Comparison between measurements and CitySim simulation. Start on October 2<sup>nd</sup> and finish on January 23<sup>rd</sup>. Measurements have missing values between December 7<sup>th</sup> and January 21<sup>st</sup>.

be observed, and can be explained by the fact that the exact inner workings of the substations are not known. The average relative error of the CitySim results is 1.1%, but the root mean square of the relative error is 17%.

### Temperature difference

Figure 4.21 shows the comparison between the measured temperature difference between the supply and return streams of the thermal station and the result of the CitySim simulation. The average relative error of the CitySim results is -4.1%, but the root mean square of the relative error is 17%.

### Thermal power

Based on the degree of freedom in the choice of mass flow and temperature difference, an interesting property to observe is the thermal power provided by the thermal station to the fluid. Figure 4.22 shows a comparison of this quantity measured, and the CitySim result. The measured quantities oscillates with a greater amplitude, but the 24 hour averages are very similar. The fact that the CitySim results oscillate less is because the consumption data of the consumers had to be smoothed, meaning that the daily oscillations. Indeed the consumption is stronger in the morning and the evening, as people are exiting and entering buildings at that moment, and consume more hot water, e.g. for showers. The average relative error of the CitySim results is -4.4%. The root mean square of the relative error is 19%, due to the oscillations of the measured power.

This brings up the fact that this thermal power provided to the fluid must equal the thermal losses in pipes, plus the consumption of the substations. Since the substations had imposed

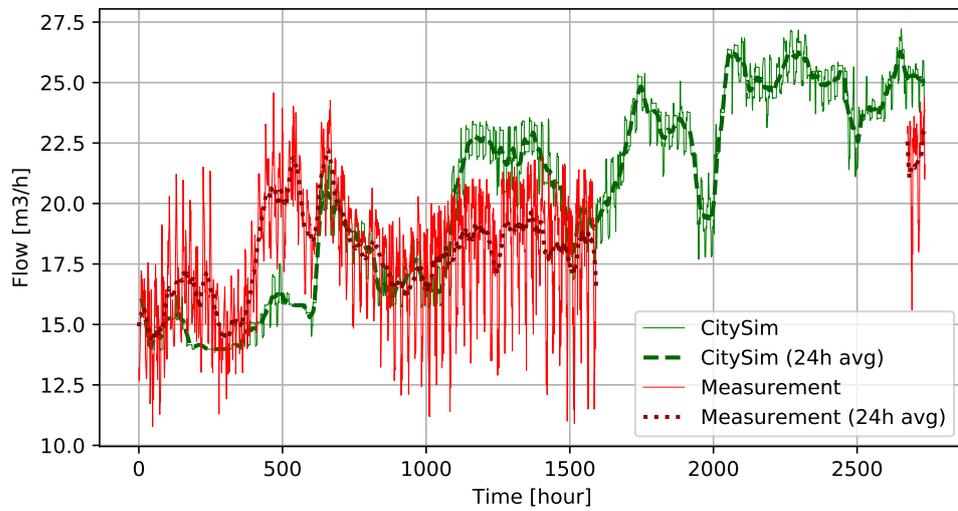


Figure 4.20 – Time evolution of flow through the thermal station. Comparison between measurements and CitySim simulation. Start on October 2<sup>nd</sup> and finish on January 23<sup>rd</sup>. Measurements have missing values between December 7<sup>th</sup> and January 21<sup>st</sup>.

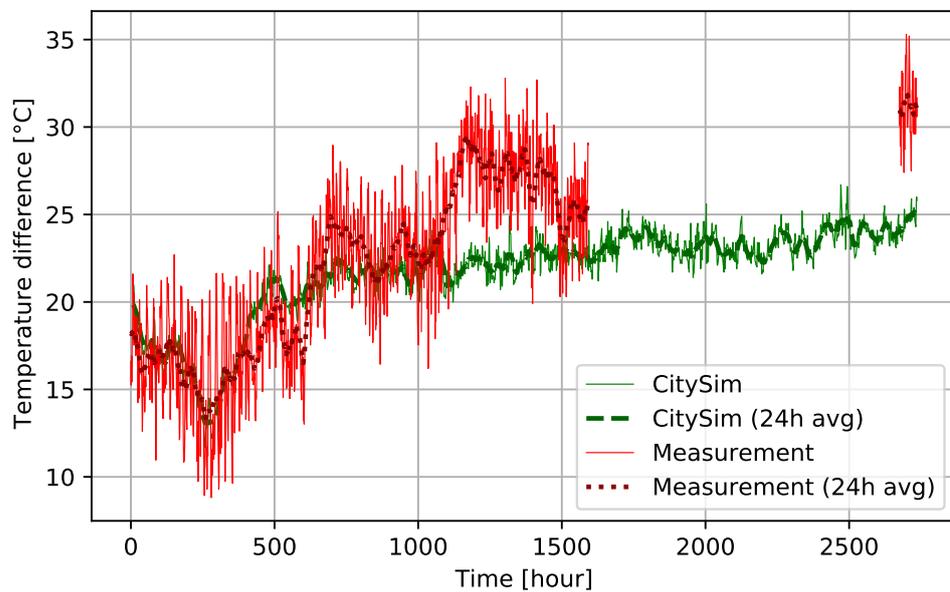


Figure 4.21 – Time evolution of temperature difference between the supply and return streams of the thermal station. Comparison between measurements and CitySim simulation. Start on October 2<sup>nd</sup> and finish on January 23<sup>rd</sup>. Measurements have missing values between December 7<sup>th</sup> and January 21<sup>st</sup>.

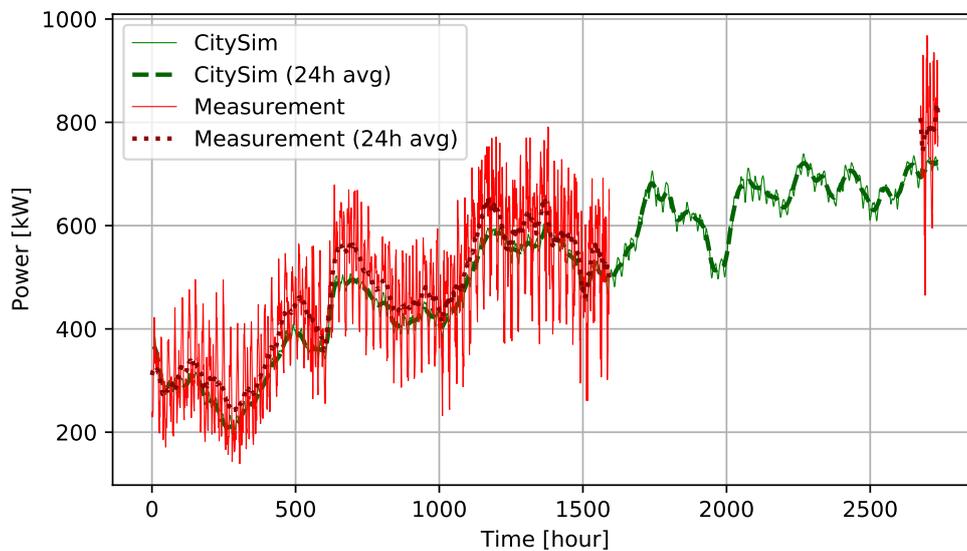


Figure 4.22 – Time evolution of thermal power provided by the thermal station to the fluid. Comparison between measurements and CitySim simulation. Start on October 2<sup>nd</sup> and finish on January 23<sup>rd</sup>. Measurements have missing values between December 7<sup>th</sup> and January 21<sup>st</sup>.

consumptions, from the measured values. Then the only source of error comes from modelling the thermal losses. Figure 4.23 shows the power separated into these categories.

It can be observed that indeed the thermal losses in the network pipes increase with time, as Broc enters winter, so soil temperatures decrease. The CitySim model underestimates the thermal losses to the environment.

### Thermal losses

There is no direct measurement of the thermal losses in pipes, but they can be estimated by subtracting the heat consumption of substations to the thermal power provided by the thermal station. However this is imperfect as values were sometimes negative, so the time series has been smoothed with a 24 hour moving average.

The climate file that used here does have one time series of the surface soil temperature in Broc, but it is not always the case. To analyse the sensitivity of the thermal losses to the surface soil temperature, a comparison with constant soil temperatures of 8 and 15°C is made, as shown in Figure 4.24.

The orders of magnitude are correct, but the error becomes quite significant at the end of the time slot. This is probably due to errors in the "measurement" data, which is actually reconstructed from other measurements, including the substation consumptions which have strongly been smoothed due to low frequency of measurements. This may also come from the fact that there are uncertainties in insulation materials types and soil conductivity around each pipe. Also the soil temperature at each pipe location is not known, only one value is given for the whole city of Broc.

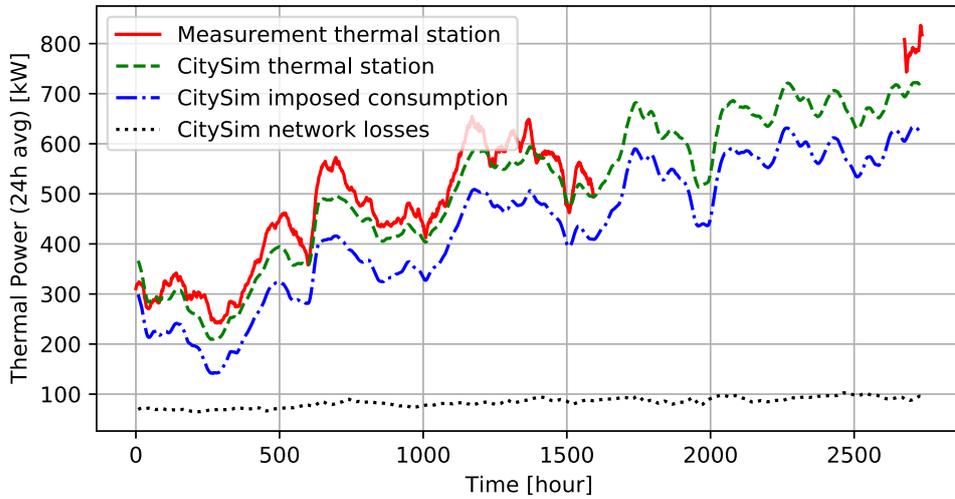


Figure 4.23 – Time evolution of thermal power provided by the thermal station to the fluid. Comparison between measurements and CitySim simulation. CitySim power separated into two its contributions : imposed building consumptions and losses in network pipes. Start on October 2<sup>nd</sup> and finish on January 23<sup>rd</sup>. Measurements have missing values between December 7<sup>th</sup> and January 21<sup>st</sup>.

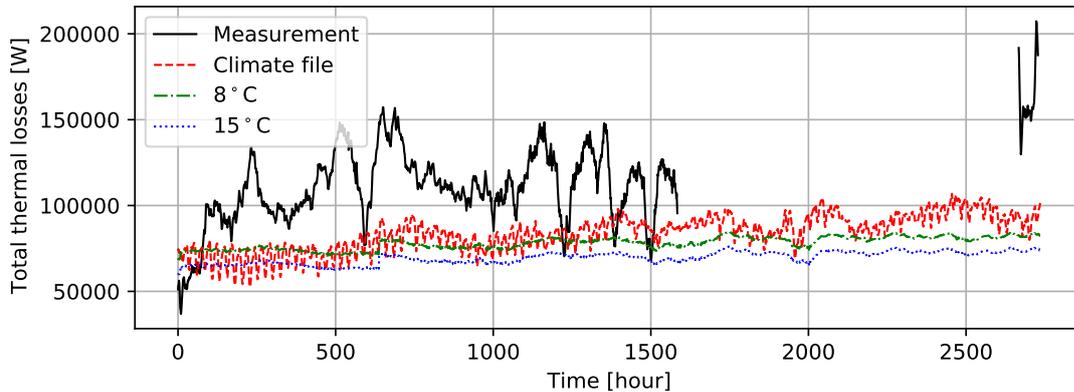


Figure 4.24 – Time evolution of thermal losses in pipes. Hour zero corresponds to October 1<sup>st</sup> at midnight. The "measurements" are actually the measured power produced at thermal station, minus the measured heat demand of consumers that has been averaged over 24 hours. The 3 simulations define respectively the surface soil temperature as the one recorded in the climate file, as a constant 8°C and as a constant 15°C.

## 4.2 Looped network

Ideally, to validate the software's ability to simulate looped topologies with multiple thermal stations, the simulations should be compared to real data, similarly to the Broc case study of section 4.1. Data for such a network has been obtained. However, due to confidentiality reasons this case study is not in the final written thesis, but it will be presented during the oral examination of the thesis. General conclusions on the control strategy have been obtained from simulations on toy networks.

### 4.2.1 Control strategy

Many different control strategies can be implemented by the user. For each thermal station, one can choose a constant pressure setpoint, a pressure setpoint that varies with mass flow, and much more can be easily implemented into CitySim. However not all of these make a stable well-behaved network, especially when there are multiple thermal stations. There are also the multiple algorithms to update the control variables to make the system converge to the correct state that have been tested.

It has been observed that the most robust way to converge to a solution rapidly, without getting stuck in loops, is to use the annealing method described in section 3.2.13. At each iteration, the damping factor increases, forcing the system to vary less and less, which can be sufficient to escape from cycles. This method is also fast to converge if the solution of the current time step is very similar to the previous one, because during the first iterations there is no or little damping. This works also quite well because the "ideal" formulas take into account the current pressure difference around the valve (see equation (3.37)) and the current mass flow around the pump (see equation (3.38)). They can adapt well to the current situation. Through plausibility checks it has been observed that the best setpoint strategies for pressure is for all thermal stations to try to establish the same differential pressure between supply and return pipes. This is similar to an electric network, where there is a uniform potential difference imposed throughout the whole grid. If different thermal stations try to impose different pressures, it usually leads to very large, unrealistic mass flows in the pipes, this is to get enough pressure drop through the pipes by friction, to make up for that pressure difference.

## 4.3 Solar thermal

No data is been obtained for the verification and validation of the solar district network. Networks like these are not numerous, and the control strategies vary greatly. However simulations have been conducted, to test for plausibility of the result and show the potential use of this tool.

### 4.3.1 Control strategy

Finding a good control strategy to make the system converge at each time step is challenging. Especially for prosumers and thermal stations with storage through which the direction of the mass flow can reverse.

For prosumers it has been found that using only a pump to impose a mass flow is not feasible in practice. The centrifugal pumps modelled are useful for controlling pressure but not flow.

This is especially true for small flows, where the characteristic curve  $\Delta p = f(\dot{m})$  is almost flat. This means using equations (3.56) and (3.57) is not desirable. To control flow, real pumps often use a flow control valve, positioned right after the pump. This is what has inspired equations (3.58) and (3.59) for the update strategy of control variables. This means a pressure setpoint needs to be added to the substation, preferably the same as the rest of the network. It works better because the pump updates to control a pressure difference, and the valve updates to control a mass flow. This is more robust, as the pump increases the pressure by too much, then the valve lowers it again. This means that if the pressure of the rest of the network varies slightly, there is less risk of overpowering the pump. Without the valve, other pumps would often overpower each other causing flows to reverse chaotically, making convergence difficult. This actually consumes more electricity, but it is the price that the pumps have to pay to gain control. If all producers impose the same pressure difference, the system is quite stable. This strategy could still be improved, especially the update formulas of control variables, as the convergence can be quite slow under certain conditions.

There have also been challenges in how to act when the prosumer is not consuming, but doesn't know yet if he will be able to inject solar heat. Indeed, the solar power to inject depends on the return temperature that will flow into the substations. While the system has not converged for that time step, this return temperature can vary. To avoid oscillation problems, the strategy that has been found is to pump from return to supply a small flow, of 0.001 kg/s. This way, the cold temperature injected into the supply does not significantly impact the supply temperature. And it is a way to keep informed on the temperature at the return side of the substation. Otherwise if instead the flow is strongly stopped, by shutting a valve, the mass flow to the substation stagnates, and its temperature drops significantly, giving a false impression that a lot of thermal power can be injected. This results in a large mass flow desired by the substation, which will then realise that the return temperature is much hotter, and that it cannot inject much solar power. An improvement could be that the substations could completely shut the flow, but still monitor the supply temperature by looking at the temperature of neighbouring nodes, where the mass flow is bigger.

For thermal stations with seasonal storage, it has been observed that using the valve update formula to try to obtain a given pressure, as done in equation (3.60), does not work very well. Indeed valves are better designed to regulate a flow not a pressure difference. The method that has been working the best is to decide for each thermal station to take a portion of the excess mass flow injected into the supply pipes, and to update the valve position to obtain that mass flow, according to equation (3.61). Indeed keeping the valve at a constant position did not work well, especially when the prosumers also try to impose a pressure setpoint.

### 4.3.2 Decentralisation

When building a solar district heating network, one needs to decide where to locate the solar collectors. To get the required area, they can all be located at a unique central location, usually near the dispatchable means of heat production. The other possibility is to dispatch the panels in many locations, usually making use of the available space of the rooftops. A central location is usually administratively simpler, and can increase efficiency of panel installations

and reduce costs. However they make the space unavailable for other activities.

The simulation tool has been used to compare the performances of networks with centralised and decentralised panels. A toy district looped network is used, with 8 buildings of different sizes and consumption patterns. The cumulated rooftop area is 2600 m<sup>2</sup>. The three scenarios of the network are

- No solar collectors.
- 2600 m<sup>2</sup> of solar collectors distributed over the building rooftops.
- 2600 m<sup>2</sup> of solar collectors in one centralized location near the thermal station and seasonal storage.

Otherwise most aspects stay the same, with a setpoint temperature of 90°C, a storage of 10<sup>10</sup> J/K, etc. Table 4.1 shows that the decentralisation is more effective in decreasing the boiler energy consumption. It also decreases the overall thermal losses, while adding the centralized solar station has no effect of thermal losses. Figures 4.25, 4.26, 4.27, 4.28, 4.29 show the time evolution of different variables in the network.

Table 4.1 – Reduction in the yearly boiler energy consumption and thermal losses through pipes when adding solar energy. Comparison between centralised and decentralised solar production.

| Solar collector distribution          | centralised | decentralised |
|---------------------------------------|-------------|---------------|
| Boiler fuel consumption reduction [%] | 36.5        | 49.8          |
| Pipe thermal losses reduction [%]     | -0.14       | 18.9          |

Overall, the basic operation of a centralised solar plant is the same as a classical centralised boiler thermal station. As can be seen in Figures 4.27, 4.28 and 4.29, the mass flow to the substations and the supply temperature are the same, explaining why the thermal losses are similar.

A difference between regular central boilers and centralised solar can be seen in Figure 4.26. Indeed the pressure is more difficult to stabilise as the prosumer solar field tries to send solar heated fluid in reverse through the thermal station with seasonal storage. This shows the limits of the developed algorithms, and reflects an unstable reality of such systems, as sometimes it is difficult for to converge when the solar field wants to impose a given mass flow at the correct pressure. So there are time steps with greatly varying pressure. What could be implemented, which would reflect quite well real centralised installations, is directly making a thermal station with with a solar collector field, instead of having the solar field as a prosumer substation. In which case the solar field is not hydraulically connected to the rest of the network, it has a heat exchanger with the seasonal storage. The computations could only be from a global energetic point of view, and there would be no need for hydraulic computations.

The energy injected into and taken from the storage are very similar for both centralised and decentralised as shown on Figure 4.25. This is because that surplus of solar energy happens at the same moments in both cases. It happens at moments when space heat demand

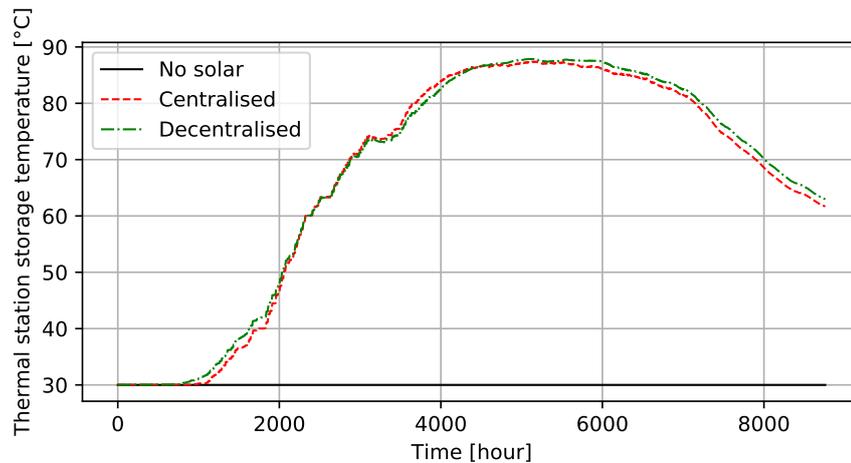


Figure 4.25 – The time evolution of the thermal station's seasonal storage temperature is compared between a non-solar, a centralised solar and a decentralised solar district heating network. The zero<sup>th</sup> hour is the beginning of January 1<sup>st</sup>.

is low, because the sun radiation heats the buildings. The centralised and decentralised collectors inject similar excess energy, at the same supply temperature. The decentralised configuration manages to inject only a little more after the whole year. The general reason why the decentralised configuration reduces more the fuel consumption is because it delivers the solar power more efficiently. Indeed the temperatures needed are lower, for example the space heating only requires temperatures around 30°C, there is no need to inject into the network at high supply temperature of 90°C. Also there are no losses in transport through the pipes.

When the solar power received is only a little more than the power consumed by a prosumer, the surplus is small and therefore more difficult to inject into the network. Indeed the efficiency of the panel drops to inject at high supply temperature. So the prosumer substations are in standby mode, where they pump a small flow from return to supply, to get a correct measure of the return temperature. If all substations are in this case, then the entire mass flow drops throughout the network, which means that thermal losses are smaller, but they dramatically decrease the temperature of the quasi-static flow. This results in a drop of the supply temperature in the network. This phenomena can be seen in Figures 4.27, 4.28, 4.29, as the decentralised curve really stands out from the others.

In the centralised configuration, when the consumer substations don't consume power, they let a minimal flow through which is greater (a fourth of the nominal flow as seen in equation (2.4)). This is better for the stability of the network as variables don't change as dramatically, but it is at the cost of greater thermal losses. In future works, it would be interesting to improve further these operation strategies.

A downside to the decentralised configuration is its high variability. The desired mass flows of prosumer substation changes much more often, and even reversing its direction. This is more complicated to operate physically, and reflects on the performance of the simulation. Indeed the system sometimes has trouble converging to the desired state of mass flows and pressures,

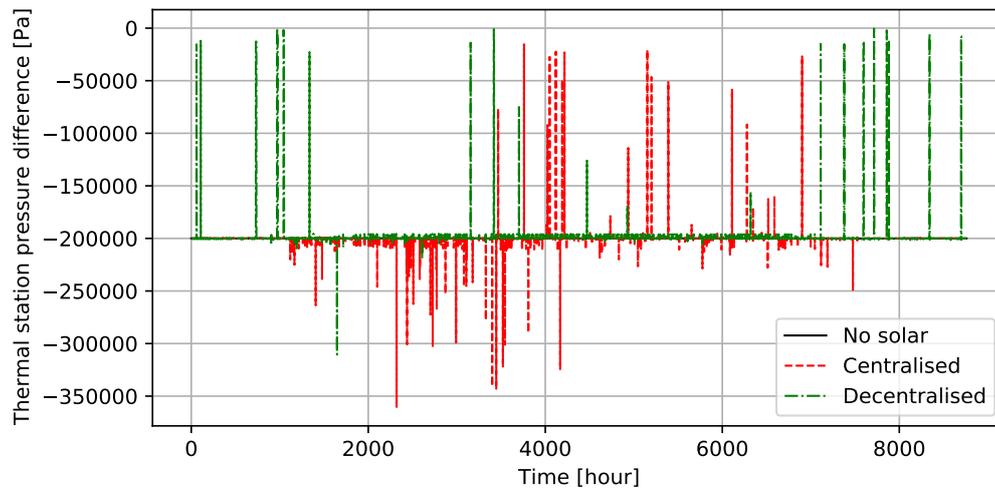


Figure 4.26 – The time evolution of the differential pressure at the thermal station is compared between a non-solar, a centralised solar and a decentralised solar district heating network. The zero<sup>th</sup> hour is the beginning of January 1<sup>st</sup>.

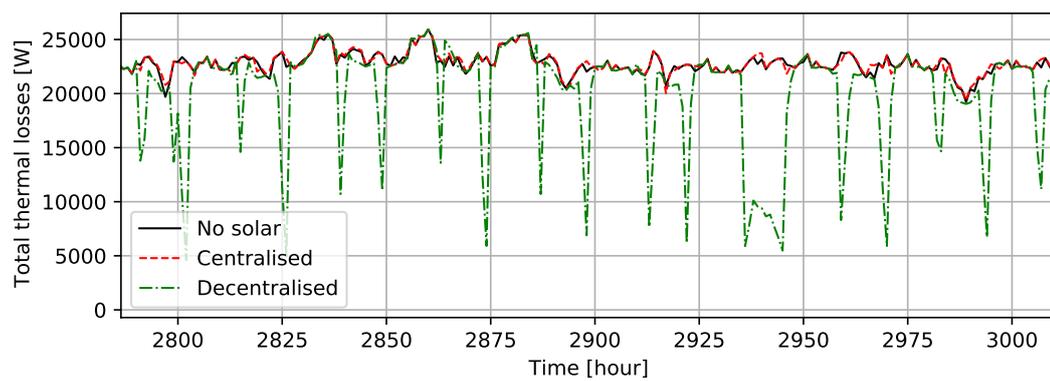


Figure 4.27 – The time evolution of the thermal losses in pipes is compared between a non-solar, a centralised solar and a decentralised solar district heating network. The zero<sup>th</sup> hour is the beginning of January 1<sup>st</sup>.

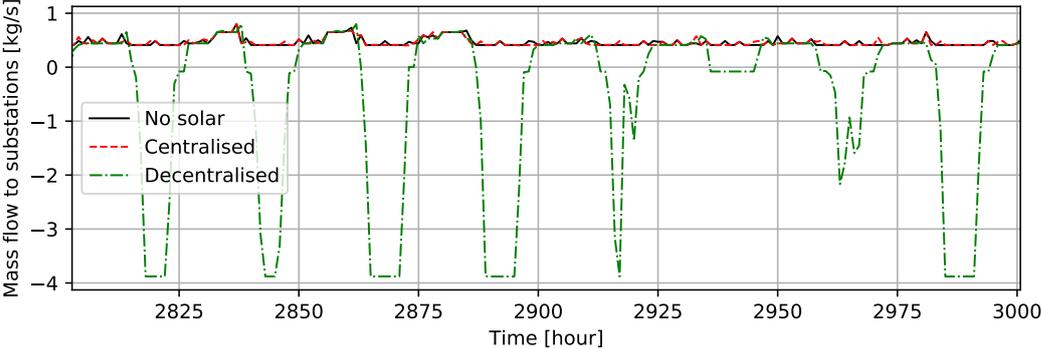


Figure 4.28 – The time evolution of the total mass flow being supplied to the substations is compared between a non-solar, a centralised solar and a decentralised solar district heating network. The zero<sup>th</sup> hour is the beginning of January 1<sup>st</sup>.

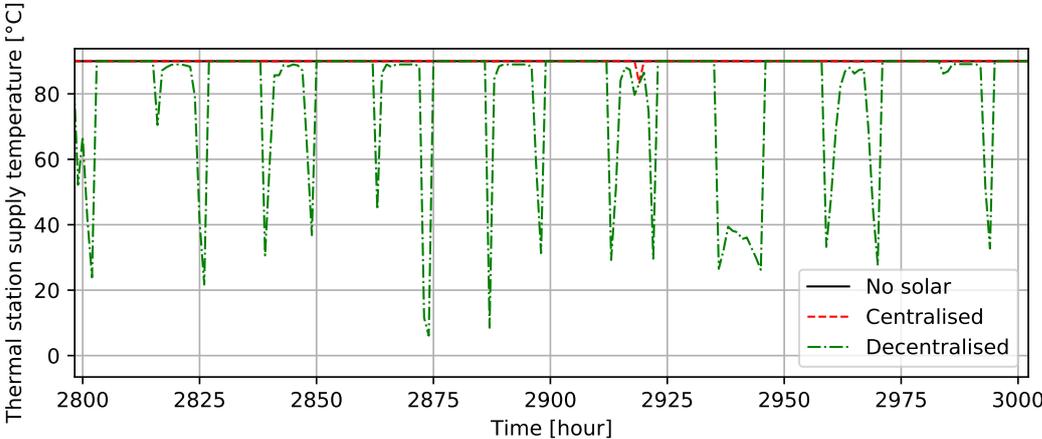


Figure 4.29 – The time evolution of the supply temperatures at the thermal station is compared between a non-solar, a centralised solar and a decentralised solar district heating network. The zero<sup>th</sup> hour is the beginning of January 1<sup>st</sup>.

which sometimes leads pressure drops, as seen in Figure 4.26, sometimes the pressure almost being in the wrong direction.

This variability also poses difficulties to the convergence algorithm. Since the desired mass flows are very different from one step to the next, the initial condition is not very good, taking more iterations to reach the final state. The simulations of the decentralised were 10 times longer than the centralised counterparts.

### 4.3.3 Setpoint supply temperature

The simulation tool has been used to study of the impact of the setpoint supply temperature on the network performances.

The solar thermal collectors have performances that vary with the temperature of the fluid that is heated as shown in equation (2.1). The efficiency decreases quadratically as the temperature difference between the average fluid temperature  $T_{fl}$  and the outside air temperature  $T_a$  increases. The prosumer substation has to set the temperature at a fixed pre-determined setpoint, which is usually common to the whole network, to ensure that all consumers get a sufficiently hot enough fluid coming in from the primary network. It can even reach zero efficiency, meaning that even if the solar panel has positive solar gain  $G$  [ $W/m^2$ ], it cannot inject any energy to the network, to avoid lowering the temperature of the supply pipes.

To observe this phenomena, simulations of a toy district network are made. The only varied variable between the simulations is the target supply temperature. The values chosen are  $T_{sup} \in \{60, 70, 80, 90\}$  [ $^{\circ}C$ ]. The heat demand of the consumers is the same for all simulations. The setpoint temperature is chosen constant throughout the year for simplicity. The simulations last a whole year.

At different supply temperatures in the network, the thermal losses in pipes change. To measure the performance of the solar injection, for each supply temperature, two simulations are made :

- a simulation without solar injection, where the only heat source is the boiler.
- a simulation with solar injection coming from one centralised solar plant with 2500  $m^2$  of panels.

The reduction in fuel consumption between these two scenarios is shown in Table 4.2. Figure 4.30 shows the time evolution of the boiler power produced, comparing with and without solar at  $T_{sup} = 60^{\circ}C$ .

Table 4.2 – Reduction in the yearly boiler energy consumption when adding a central solar power plant. Comparison between different setpoint supply temperature.

| Setpoint temperature [ $^{\circ}C$ ]                               | 60   | 70   | 80   | 90   |
|--|------|------|------|------|
| Boiler production without solar $E_{no\ solar}$ [MWh]              | 235  | 254  | 271  | 289  |
| Boiler production with solar $E_{solar}$ [MWh]                     | 144  | 166  | 189  | 215  |
| Energy reduction $(E_{no\ solar} - E_{solar}) / E_{no\ solar}$ [%] | 38.9 | 34.6 | 30.2 | 25.7 |

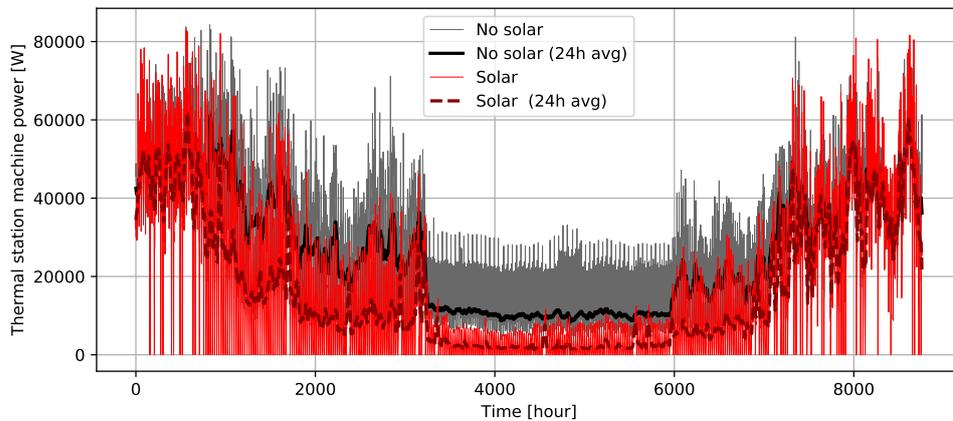


Figure 4.30 – Comparisons of the thermal station boiler production with and without the help from a centralised solar power plant. Time evolution starting on January 1rst, ends on December 31rst. Setpoint temperature of 60°C.

This brings up an important aspect of incorporating solar thermal energy to district heating networks. To avoid wasting resources, and get the most out of these panels, it is important to have collectors that are efficient at high fluid temperatures. These are usually collectors which have good insulation of the fluid, usually by surrounding it with a layer of vacuum. These collectors are more expensive of course. Tools such as the one developed in this thesis can help make the thermo-economic optimisations.

Also it is important to evolve towards district networks with lower supply temperatures. This is exactly what is recommended for the newer generations of district heating. District heating networks are usually classified in generations [7]. The old first generations had very high supply temperatures. Each new generation has objectives to improve the performances, which means reduction of thermal losses, and reduction of temperatures in the network. The current generations have arrived at the lower limit of supply temperature. Indeed, to heat domestic hot water, at least a temperature of 55°C is needed for health concerns, to avoid bacteria such as Legionella. The fifth generation district networks avoid this problem and manage to supply temperatures close to ambient, such as 25°C. To do so, users extract the heat from the fluid with a heat pump. Using a heat pump with the flowing water of the district network is much more efficient than regular air-heat pumps. This can be seen as the "electrification of heating", similarly to how electric vehicles electrify transport. There is a large reduction in thermal losses, and the the network can even be used for cooling. In general with low temperatures like these, the network can act as a means of transporting heat between many different prosumers.

#### 4.3.4 Seasonal storage

Solar energy is abundant in summer, when the heat demand is low, and little present in winter, when the heat demand is strong. This is one of the challenges of the intermittent solar energy. This issue is addressed by using a seasonal heat storage. However they are expensive, and can

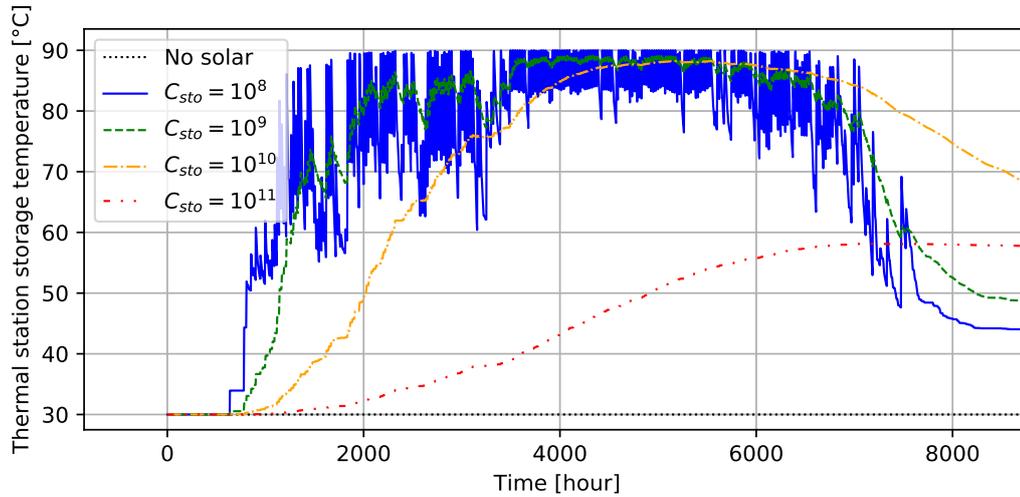


Figure 4.31 – Comparisons of the thermal station seasonal storage temperature for different storage heat capacities  $C_{sto}$  [J/K]. Time evolution starting on January 1rst, ends on December 31rst. Setpoint temperature of 90°C.

take up quite a bit of space. This calls for careful sizing of these storages.

The simulation tool has been used on a toy district network, to study the effect of the storage heat capacity  $C_{sto}$  [J/K] on the performance of the network. In the current simple storage model implemented, which can be considered as a tank of water, this is proportional to the volume of water that it can contain. The simulation is over a whole year, the supply temperature is constant at 90°C. Table 4.3 shows the fuel consumption reduction that can be performed with different heat capacities. Figure 4.31 shows the time evolution of the storage temperature during the year simulated.

Table 4.3 – Reduction in the yearly boiler energy consumption when adding a central solar power plant and a seasonal storage. Comparison between different heat capacities.

| Heat capacity $C_{sto}$ [J/K]         | $10^7$ | $10^8$ | $10^9$ | $10^{10}$ | $10^{10}$ |
|---------------------------------------|--------|--------|--------|-----------|-----------|
| Boiler fuel consumption reduction [%] | 15.8   | 35.1   | 41.8   | 44.1      | 13.9      |

It can be observed that for small heat capacities, the storage temperature varies greatly. It quickly goes up to the supply temperature, and is then stuck and cannot be heated further. The solar heat is then lost, dissipated to the environment via thermal losses in the pipes. Then during the nights, when the sun is not shining, the storage's energy is used to preheat the fluid for the boiler. But again, the temperature varies pretty fast, and the tank has too low of a temperature to help the boiler. Overall this doesn't reduce the fuel consumption by as much as the bigger tanks as shown in Table 4.3. The simulation tool shows that under these conditions, the best storage heat capacity for the district is around  $10^{10}$  J/K.

It can also be noted that if the tank's heat capacity is too big, it doesn't contribute in a significant reduction of fuel consumption. This is because not enough energy has been produced over the

summer to heat the tank to high enough temperatures. This points out a possible improvement in the simple storage model developed in this thesis. The tank only has one global temperature. When the solar thermal power is injected at the hot supply temperature  $T_{\text{sup}}$ , and it gives heat to the storage at  $T_{\text{sto}}$ , there is a lot of exergy loss in the process. This is because the heat is exchanged at a large temperature difference.

In real thermal storage tanks, the temperature throughout the tank is not uniform, there is usually a stratification that takes place. For example in water tanks, hotter water being less dense, moves to the top and stays there, while the cold water sinks to the bottom. This stratification is quite stable and can be maintained, throughout the whole year in seasonal storages. The advantage of such tanks is that the heat of the incoming solar heated fluid is first exchanged at with the hot water at the top of the tank, with little temperature difference. Then as the "solar" fluid is cooled, it goes down the tank to exchange heat with the lower temperature water in the tank. This is the same mechanics explaining why counter-current heat exchangers are so often used.

The thermal energy is therefore partly stored at high temperature, and partly in at low temperature. Unlike the implemented simple model where all the energy is stored at the same temperature. This is useful because even when the tank doesn't have a lot of energy stored, it can heat the fluid to higher temperature, emptying itself more rapidly in the process. For example, suppose a constant temperature tank is at  $70^{\circ}\text{C}$  and a stratified tank of same volume, has  $50^{\circ}\text{C}$  at the bottom, the  $90^{\circ}\text{C}$  at the top. For the stratified tank, the temperature is supposed to be a linear function of height. They both store the same energy, but the second tank has more exergy. Now suppose an infinitesimal flow needs to be heated. The first tank can only heat the flow up to  $70^{\circ}\text{C}$ , while the second tank can give more energy and heat it all the way up to  $90^{\circ}\text{C}$ . Such stratified tanks could be implemented in CitySim as improvements.

To make it more realistic, the thermal losses of the heat storage could also be implemented, as the current simple model is supposed perfectly insulated.

## 5 Conclusion

Energy is the key factor to power the modern societies that ensures comfortable ways of life. The energy supply in most countries, including Switzerland, is dependent on non-renewable, greenhouse gas emitting, imported energy sources. The non-renewable aspect will be problematic when these sources run out, if the dependence is too strong, and the transition not well-prepared. Similarly these importations create a dependence on the exporting countries. The greenhouse gas emissions are causing climate change and global warming which is becoming an increasing threat to humans, making problems far worst to humanity than the comfort provided from these fossil energy sources.

Heating represents a significant part on the energy used in Switzerland. For example, 79% on the energy consumed in households is for heating [15]. This primary energy sources used for heating is largely dominated by fossil gas and fuel oil. An alternative solution explored to generate heat is the use of low-fossil-carbon district heating networks. Some existing networks are already powered by biomass, which can be exploited in renewable ways with low fossil-carbon footprint. The newer technologies increasingly explore the use of solar thermal energy to provide renewable heat to the network. However there are technical challenges due to the intermittent nature of this source.

It is important to push this technology as part of a global plan to reduce the fossil-based heating impacts. However, in order to build networks with the correct sizing, the complications due to the intermittent solar source require detailed simulations of such solar district heating networks. Few software tools have been found that can simulate in detail the building thermal physics to get the heat demand and simulate the district heating networks with complex topologies as well as the possibility to inject solar thermal power. This thesis has aimed to build such a tool based on the existing CitySim, urban physics simulation software. The software is free and open source.

The original state of the CitySim code is described with solar irradiance computations that are used to simulate the thermal physics of buildings and solar thermal collectors. It also contains a pre-existing simplified district heating simulation code that had not been validated due to lack of data.

From there, this thesis implements many additions and more robust code to simulate district heating and cooling networks with branched topologies. Overall this step requires no complicated mathematics and converges efficiently.

The next step is to implement more complex topologies, which are not uncommon in real cases of district networks. It is important for the software to be able to model them. The bigger complexity of looped topologies with multiple thermal stations is explored. The mathematical models behind the code implemented are described. Challenges arise in the ways to converge to results rapidly. Multiple algorithms are proposed to face these challenges.

The simulation tool is tested against real data from a district heating network in Broc, Switzerland. The results show satisfying results in simulating thermal losses and power consumption. It also shows that it is of great importance to have detailed enough data. Characteristics of the buildings are important to simulate the correct heat demand. Dimensions and thermal insulation information of the pipes are important to model the thermal losses.

The Broc network has a branched topology and is powered by wood and gas. A validation has also been made on a looped network. There is still a need to validate and verify the model against real data for the solar injections. The simulation tool has nevertheless been tested on toy networks, to conduct plausibility tests and explore the potential of the tool. The better control strategies have been found. More particularly, when all producers aim to supply the same differential pressure. Also when prosumers have pump and a flow control valve.

Simulations with solar injections analyse important aspects of such networks. The importance of the supply temperature is pointed out. Lower temperatures are more desirable to increase the efficiency of solar collectors. Choosing the correct size and type of the seasonal storage used to store the excess solar energy can increase the performance of the system. The complexity of operation of such networks is pointed out, and a strategy is proposed.

To build further on this simulation tool, it could be interesting to implement more components. For example other types of seasonal storage. There is also a need for further research on control strategies and algorithms to converge better and faster to the solution state in solar district networks.

## Bibliography

- [1] Mohamed Hany Abokersh et al. “A framework for the optimal integration of solar assisted district heating in different urban sized communities: A robust machine learning approach incorporating global sensitivity analysis”. In: *Applied Energy* 267 (2020), p. 114903. ISSN: 0306-2619. DOI: <https://doi.org/10.1016/j.apenergy.2020.114903>. URL: <http://www.sciencedirect.com/science/article/pii/S0306261920304153>.
- [2] U.S. Energy Information Administration. *Natural gas explained*. <https://www.eia.gov/energyexplained/natural-gas/use-of-natural-gas.php>. Accessed: 2020-07-31. 2020.
- [3] Meteotest AG. *Metenorm*. URL: <https://meteonorm.com/en/> (visited on 07/27/2020).
- [4] ASHRAE. *ASHRAE Handbook - Fundamental*. 2009.
- [5] S. Bennett. “A brief history of automatic control”. In: *IEEE Control Systems Magazine* 16.3 (1996), pp. 17–25.
- [6] Lisa Brand et al. “Smart district heating networks - A simulation study of prosumers’ impact on technical parameters in distribution networks”. In: *Applied Energy* 129 (2014), pp. 39–48. ISSN: 0306-2619. DOI: <https://doi.org/10.1016/j.apenergy.2014.04.079>. URL: <http://www.sciencedirect.com/science/article/pii/S0306261914004425>.
- [7] Simone Buffa et al. “5th generation district heating and cooling systems: A review of existing cases in Europe”. In: *Renewable and Sustainable Energy Reviews* 104 (2019), pp. 504–522. ISSN: 1364-0321. DOI: <https://doi.org/10.1016/j.rser.2018.12.059>. URL: <http://www.sciencedirect.com/science/article/pii/S1364032118308608>.
- [8] Vanessa Burg et al. “Analyzing the potential of domestic biomass resources for the energy transition in Switzerland”. In: *Biomass and Bioenergy* 111 (2018), pp. 60–69. ISSN: 0961-9534. DOI: <https://doi.org/10.1016/j.biombioe.2018.02.007>. URL: <http://www.sciencedirect.com/science/article/pii/S0961953418300382>.
- [9] E. Carpaneto, P. Lazzeroni, and M. Repetto. “Optimal integration of solar energy in a district heating network”. In: *Renewable Energy* 75 (2015), pp. 714–721. ISSN: 0960-1481. DOI: <https://doi.org/10.1016/j.renene.2014.10.055>. URL: <http://www.sciencedirect.com/science/article/pii/S096014811400682X>.
- [10] Wikimedia Commons. *File:Density of ice and water (en).svg* — *Wikimedia Commons, the free media repository*. [Online; accessed 6-August-2020]. 2018. URL: [https://commons.wikimedia.org/w/index.php?title=File:Density\\_of\\_ice\\_and\\_water\\_\(en\).svg&oldid=286971273](https://commons.wikimedia.org/w/index.php?title=File:Density_of_ice_and_water_(en).svg&oldid=286971273).

- [11] Wikimedia Commons. *File:Moody EN.svg* — *Wikimedia Commons, the free media repository*. [Online; accessed 6-August-2020]. 2018. URL: [https://commons.wikimedia.org/w/index.php?title=File:Moody\\_EN.svg&oldid=297049471](https://commons.wikimedia.org/w/index.php?title=File:Moody_EN.svg&oldid=297049471).
- [12] Peter Deuffhard. *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Publishing Company, Incorporated, 2011. ISBN: 364223898X.
- [13] Swiss Federal Office for Energy. URL: <https://www.bfe.admin.ch/bfe/en/home/research-and-cleantech/research-programmes/solar-heat-and-heat-storage.html> (visited on 08/06/2020).
- [14] EPFL. *CitySim Software*. 2015. URL: <https://www.epfl.ch/labs/leso/transfer/software/citysim/> (visited on 08/03/2020).
- [15] eurostat. *Energy consumption in households*. [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy\\_consumption\\_in\\_households](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households). Accessed: 2020-07-31. 2020.
- [16] Jason Galley. *Simulation et optimisation SIG du réseau CAD de Broc*. Haute Ecole d'ingénierie et d'architecture de Fribourg, 2019.
- [17] "German central solar heating plants with seasonal heat storage". In: *Solar Energy* 84.4 (2010). International Conference CISBAT 2007, pp. 612–623. ISSN: 0038-092X. DOI: <https://doi.org/10.1016/j.solener.2009.05.013>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X09001224>.
- [18] Luc Girardin et al. "EnerGis: A geographical information based system for the evaluation of integrated energy conversion systems in urban areas". In: *Energy* 35.2 (2010). ECOS 2008, pp. 830–840. ISSN: 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2009.08.018>. URL: <http://www.sciencedirect.com/science/article/pii/S0360544209003582>.
- [19] Loïc Giraud. "Dynamic modeling and advanced management of heat networks". Theses. Université Grenoble Alpes, Oct. 2016. URL: <https://hal.archives-ouvertes.fr/tel-01687562>.
- [20] Solar District Heating. *Solar district heating guidelines*. URL: [https://www.solar-district-heating.eu/wp-content/uploads/2018/06/SDH-Guidelines\\_update\\_09.2017.pdf](https://www.solar-district-heating.eu/wp-content/uploads/2018/06/SDH-Guidelines_update_09.2017.pdf) (visited on 07/28/2020).
- [21] M. Heymann et al. "Concept and Measurement Results of Two Decentralized Solar Thermal Feed-in Substations". In: *Energy Procedia* 149 (2018). 16th International Symposium on District Heating and Cooling, DHC2018, 9–12 September 2018, Hamburg, Germany, pp. 363–372. ISSN: 1876-6102. DOI: <https://doi.org/10.1016/j.egypro.2018.08.200>. URL: <http://www.sciencedirect.com/science/article/pii/S187661021830496X>.
- [22] M.N. Howell et al. "Continuous action reinforcement learning applied to vehicle suspension control". In: *Mechatronics* 7.3 (1997), pp. 263–276. ISSN: 0957-4158. DOI: [https://doi.org/10.1016/S0957-4158\(97\)00003-2](https://doi.org/10.1016/S0957-4158(97)00003-2). URL: <http://www.sciencedirect.com/science/article/pii/S0957415897000032>.

- [23] M.N Howell and M.C Best. “On-line PID tuning for engine idle-speed control using continuous action reinforcement learning automata”. In: *Control Engineering Practice* 8.2 (2000), pp. 147–154. ISSN: 0967-0661. DOI: [https://doi.org/10.1016/S0967-0661\(99\)00141-0](https://doi.org/10.1016/S0967-0661(99)00141-0). URL: <http://www.sciencedirect.com/science/article/pii/S0967066199001410>.
- [24] Eric Johnson. “Goodbye to carbon neutral: Getting biomass footprints right”. In: *Environmental Impact Assessment Review* 29.3 (2009), pp. 165–168. ISSN: 0195-9255. DOI: <https://doi.org/10.1016/j.eiar.2008.11.002>. URL: <http://www.sciencedirect.com/science/article/pii/S0195925508001637>.
- [25] kaemco. *CitySim Pro*. URL: <http://www.kaemco.ch/download.php> (visited on 08/03/2020).
- [26] The Solar Keymark. URL: <http://www.estif.org/solarkeymarknew/> (visited on 08/06/2020).
- [27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. In: *Science* 220.4598 (1983), pp. 671–680. ISSN: 0036-8075. DOI: 10.1126/science.220.4598.671. eprint: <https://science.sciencemag.org/content/220/4598/671.full.pdf>. URL: <https://science.sciencemag.org/content/220/4598/671>.
- [28] A. W. Moore L. P. Kaelbling M. L. Littman. “Reinforcement Learning: A Survey”. In: *Journal of Artificial Intelligence Research* 4 (1996), pp. 237–285. DOI: <https://doi.org/10.1613/jair.301>. URL: <https://www.jair.org/index.php/jair/article/view/10166>.
- [29] Bruce E Larock, Roland W Jeppson, and Gary Z Watters. *Hydraulics of pipeline systems*. CRC press, 1999.
- [30] Gunnar Lennermo, Patrick Lauenburg, and Sven Werner. “Control of decentralised solar district heating”. In: *Solar Energy* 179 (2019), pp. 307–315. ISSN: 0038-092X. DOI: <https://doi.org/10.1016/j.solener.2018.12.080>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X18312775>.
- [31] B.S. Massey. *Mechanics of Fluids*. Springer US, 1989. ISBN: 9780412342806. URL: <https://books.google.ch/books?id=CQNEAQAIAAJ>.
- [32] Georgios Mavromatidis. “Model-based design of distributed urban energy systems under uncertainty”. PhD thesis. ETH Zürich, 2017. URL: <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/182697/PhDthesis.pdf?sequence=2&isAllowed=y>.
- [33] Matt Milbury. *Pipe Inch to Meter Conversion*. 2016. URL: <http://www.piping-designer.com/index.php/disciplines/mechanical/stationary-equipment/pipe/493-pipe-inch-to-meter-conversion> (visited on 07/26/2020).
- [34] N. Minorsky. “DIRECTIONAL STABILITY OF AUTOMATICALLY STEERED BODIES”. In: *Journal of the American Society for Naval Engineers* 34.2 (1922), pp. 280–309. DOI: 10.1111/j.1559-3584.1922.tb04958.x. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1559-3584.1922.tb04958.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1559-3584.1922.tb04958.x>.

- [35] Camilo et al. Mora. “Global risk of deadly heat”. In: *Nature Climate Change* (2017), pp. 501–506. DOI: <https://doi.org/10.1038/nclimate3322>.
- [36] Thomas Oppelt. “Modell zur Auslegung und Betriebsoptimierung von Nah- und Fernkältenetzen”. PhD thesis. Oct. 2015.
- [37] Bon Lambert - OEM Panels. *VFD Variable Frequency Drive and Centrifugal Pump*. [Online; accessed 6-August-2020]. 2018. URL: <https://www.oempanels.com/vfd-variable-frequency-drive-and-centrifugal-pump>.
- [38] Diane Perez. *A framework to model and simulate the disaggregated energy flows supplying buildings in urban areas*. Tech. rep. EPFL, 2014.
- [39] QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation. 2009. URL: <http://qgis.osgeo.org>.
- [40] Renaldi Renaldi and Daniel Friedrich. “Techno-economic analysis of a solar district heating system with seasonal thermal storage in the UK”. In: *Applied Energy* 236 (2019), pp. 388–400. ISSN: 0306-2619. DOI: <https://doi.org/10.1016/j.apenergy.2018.11.030>. URL: <http://www.sciencedirect.com/science/article/pii/S0306261918317379>.
- [41] P. J. ROACHE. *Verification and validation in computational science and engineering*. Albuquerque: Hermosa, 1998. ISBN: 978-0-913478-08-0.
- [42] Herbert Robbins and Sutton Monro. “A Stochastic Approximation Method”. In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. ISSN: 00034851. URL: <http://www.jstor.org/stable/2236626>.
- [43] Abdel Rodríguez et al. *Improving the performance of Continuous Action Reinforcement Learning Automata*.
- [44] G. Santharam, P.S. Sastry, and M.A.L. Thathachar. “Continuous action set learning automata for stochastic optimization”. In: *Journal of the Franklin Institute* 331.5 (1994), pp. 607–628. ISSN: 0016-0032. DOI: [https://doi.org/10.1016/0016-0032\(94\)90039-6](https://doi.org/10.1016/0016-0032(94)90039-6). URL: <http://www.sciencedirect.com/science/article/pii/0016003294900396>.
- [45] J. Sarwar et al. “Performance analysis of a flat plate collector to achieve a fixed outlet temperature under semi-arid climatic conditions”. In: *Solar Energy* 207 (2020), pp. 503–516. ISSN: 0038-092X. DOI: <https://doi.org/10.1016/j.solener.2020.06.088>. URL: <http://www.sciencedirect.com/science/article/pii/S0038092X20307039>.
- [46] R.K. Shah and D.P. Sekulic. *Fundamentals of Heat Exchanger Design*. Wiley, 2003. ISBN: 9780471321712. URL: <https://books.google.ch/books?id=beSXNAZblWQC>.
- [47] James C. Spall. “Stochastic Optimization, Stochastic Approximation and Simulated Annealing”. In: *Wiley Encyclopedia of Electrical and Electronics Engineering*. American Cancer Society, 1999. ISBN: 9780471346081. DOI: 10.1002/047134608X.W1044. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/047134608X.W1044>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W1044>.
- [48] *Statistique globale suisse de l'énergie 2017*. Office fédéral de l'énergie, 2017.

- [49] Greg Stickler. *Solar Radiation and the Earth System*. URL: <https://web.archive.org/web/20160425164312/http://education.gsfc.nasa.gov/experimental/July61999siteupdate/inv99Project.Site/Pages/science-briefs/ed-stickler/ed-irradiance.html> (visited on 07/28/2020).
- [50] Zhiyong Tian et al. “Analysis and validation of a quasi-dynamic model for a solar collector field with flat plate collectors and parabolic trough collectors in series for district heating”. In: *Energy* 142 (2018), pp. 130–138. ISSN: 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2017.09.135>. URL: <http://www.sciencedirect.com/science/article/pii/S0360544217316572>.
- [51] Federal Office of Topography swisstopo. URL: <https://map.geo.admin.ch/> (visited on 07/26/2020).
- [52] Victor Tulus et al. “Economic and environmental potential for solar assisted central heating plants in the EU residential sector: Contribution to the 2030 climate and energy EU agenda”. In: *Applied Energy* 236 (2019), pp. 318–339. ISSN: 0306-2619. DOI: <https://doi.org/10.1016/j.apenergy.2018.11.094>. URL: <http://www.sciencedirect.com/science/article/pii/S0306261918318105>.
- [53] Alternative Energy Tutorials. *Solar Evacuated Tube Collectors for Solar Hot Water*. URL: <http://www.alternative-energy-tutorials.com/solar-hot-water/evacuated-tube-collector.html> (visited on 07/28/2020).
- [54] Bundesamt für Umwelt BAFU. *Kenngrossen zur Entwicklung der Treibhausgasemissionen in der Schweiz 1990-2018*. [https://www.bafu.admin.ch/dam/bafu/en/dokumente/klima/fachinfo-daten/kenngrossen\\_thg\\_emissionen\\_schweiz.pdf.download.pdf/Kenngr%C3%B6ssen\\_2020\\_D.pdf](https://www.bafu.admin.ch/dam/bafu/en/dokumente/klima/fachinfo-daten/kenngrossen_thg_emissionen_schweiz.pdf.download.pdf/Kenngr%C3%B6ssen_2020_D.pdf). Accessed: 2020-07-31. 2020.
- [55] Mattias Vesterlund, Andrea Toffolo, and Jan Dahl. “Simulation and analysis of a meshed district heating network”. In: *Energy Conversion and Management* 122 (2016), pp. 63–73. ISSN: 0196-8904. DOI: <https://doi.org/10.1016/j.enconman.2016.05.060>. URL: <http://www.sciencedirect.com/science/article/pii/S0196890416304381>.
- [56] Danhong Wang, Kristina Orehounig, and Jan Carmeliet. “A Study of District Heating Systems with Solar Thermal Based Prosumers”. In: *Energy Procedia* 149 (2018). 16th International Symposium on District Heating and Cooling, DHC2018, 9–12 September 2018, Hamburg, Germany, pp. 132–140. ISSN: 1876-6102. DOI: <https://doi.org/10.1016/j.egypro.2018.08.177>. URL: <http://www.sciencedirect.com/science/article/pii/S1876610218304739>.
- [57] Hai Wang, Haiying Wang, and Tong Zhu. “A new hydraulic regulation method on district heating system with distributed variable-speed pumps”. In: *Energy Conversion and Management* 147 (2017), pp. 174–189. ISSN: 0196-8904. DOI: <https://doi.org/10.1016/j.enconman.2017.03.059>. URL: <http://www.sciencedirect.com/science/article/pii/S0196890417302704>.

- [58] Yaran Wang et al. “Hydraulic performance optimization of meshed district heating network with multiple heat sources”. In: *Energy* 126 (2017), pp. 603–621. ISSN: 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2017.03.044>. URL: <http://www.sciencedirect.com/science/article/pii/S0360544217304103>.
- [59] Nichols N.B. Ziegler J.G. “Optimum settings for automatic controllers”. In: *Transactions of the ASME* 64 (1942), pp. 759–768.