



# MASTERARBEIT | MASTER'S THESIS

Titel | Title

Making the Leap from an OSM-based Model of the Austrian  
Power Grid to a Power Flow Analysis Framework

verfasst von | submitted by

Max Lorenz Neuendorf

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of  
Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree  
programme code as it appears on the  
student record sheet:

UA 066 921

Studienrichtung lt. Studienblatt | Degree  
programme as it appears on the student  
record sheet:

Masterstudium Informatik

Betreut von | Supervisor:

Univ.-Prof. Dipl.-Ing. Mag. Dr.techn. Edgar Weippl



# Acknowledgements

First of all, I would like to thank Edgar Weippl and Johanna Ullrich for the opportunity to write my thesis on this interesting topic. I would especially like to thank them for their great supervision during this journey. Firstly, I always had the opportunity to ask questions and discuss ideas. Secondly and most importantly, working together was fruitful, pleasant and at eye level. I would also like to thank my family, my friends and my girlfriend for their constant support and encouragement.



# Abstract

Electric power and its constant availability is one of the main pillars of our modern civilization. Hardly any area of our lives today is not dependent on it, be it simple things like electric light or complex communication systems such as the Internet. Consequently, maintaining the power supply is of central importance. This task includes not only the generation of electricity itself, but also its transmission and distribution to consumers. Unfortunately, recent years have shown that reliable transmission using the electricity grid cannot be taken for granted. There are numerous examples where even minor faults or accidents have led to widespread power outages. However, the grid operators keep a low profile when it comes to the condition and security of their infrastructure. Transparency is particularly important at this point in order to uncover shortcomings and hold the operators accountable. There are various approaches in research to close this knowledge gap. One of these is to use freely available information to model the electricity grid. In the case of the Austrian power grid, Klauzer et al. did this by filtering the open-source geo-database OpenStreetMaps (OSM) for elements of the power grid and creating a model based on it. In this paper, we make the model even more meaningful by extending it and transforming it into a load flow problem. Load flow analysis is a method often used in research to comprehensively describe and analyse electrical systems. It takes into account both the electrotechnical parameters of the power grid and the level of power consumption and generation in the grid. This extension is important because power grids are large systems whose stability follows complex relationships. Our contribution is twofold: Firstly, we are developing a framework to automate the process from a simple model of the electricity grid to load flow analysis. To do so, we use the open source tools Python, Octave and Matpower. In addition, visualizations with different emphases are created to facilitate understanding and working with the system. In addition, a comprehensive data set is generated for each load flow analysis performed, which can form the basis for further analyses. Secondly, we apply the framework for the months of January and July 2019, solving the load flow problem for each 15-minute interval. The results show a mixed picture. On the one hand, we achieve a good success rate of up to 85% when solving the load flow problems. Certain correlations can also be identified, for example that the losses increase with a higher transmission load, which indicates that the modeling is going in the right direction. On the other hand, there are weaknesses in the modeling that reveal areas for future research. For example, there is a group of substations that stand out with extreme values in the load flow analysis. One reason for this could be that the Austrian electricity grid is part of the European interconnected grid and interacts with it. This fact has not yet been adequately reflected in the model. Another anomaly is the high level of lost reactive power in the system. One reason for this could be the incomplete modeling of power generation. While there is no data basis

## *Abstract*

for the reactive power capacities, electricity generation can only be modeled as constant. In the real power grid, however, this is controlled in such a way that generation and consumption are roughly balanced.

# Kurzfassung

Elektrischer Strom und dessen allzeitige Verfügbarkeit ist einer der Hauptpfeiler unserer heutigen Zivilisation. Kaum ein Bereich in unserem heutigen Leben ist nicht abhängig davon, seien es simple Dinge wie elektrisches Licht oder komplexe Kommunikationssysteme wie das Internet. Folglich ist das Aufrechterhalten der Stromversorgung von zentraler Bedeutung. Diese Aufgabe umfasst dabei nicht nur die Stromerzeugung selbst, sondern auch die Übertragung und Verteilung zu den Verbrauchern. Leider hat sich in den vergangenen Jahren gezeigt, dass die zuverlässige Übertragung mithilfe des Stromnetzes keine Selbstverständlichkeit ist. So gibt es zahlreiche Beispiele, bei denen teilweise selbst kleine Störungen oder Unfälle zu weitflächigen Stromausfällen geführt haben. Doch obwohl es immer wieder Probleme mit der Stromversorgung gibt, halten sich die Netzbetreiber bedeckt, was den Zustand und die Sicherheit ihrer Infrastruktur angeht. Dabei ist an dieser Stelle Transparenz besonders wichtig, um Missstände aufzudecken und die Betreiber in die Pflicht nehmen zu können. In der Forschung gibt es verschiedene Ansätze, um diese Lücke zu schließen. Einer davon ist, frei verfügbare Informationen zu benutzen, um das Stromnetz zu modellieren. Im Falle von dem österreichischen Stromnetz haben das Klauzer et al. getan, indem sie die open-source Geo-Datenbank OpenStreetMaps (OSM) nach Elementen des Stromnetzes gefiltert und darauf ein Modell erstellt haben. In dieser Arbeit machen wir das Modell noch aussagekräftiger, indem wir es erweitern und in ein Lastflussproblem umwandeln. Die Lastflussanalyse ist ein in der Forschung oft gewähltes Mittel, um elektrische Systeme umfassend zu beschreiben und zu analysieren. Dabei werden sowohl die elektrotechnischen Parameter des Stromnetzes als auch die Höhe des Stromverbrauches sowie der Stromerzeugung im Netz berücksichtigt. Diese Erweiterung ist wichtig, da Stromnetze große Systeme sind, deren Stabilität komplexen Zusammenhängen folgt. Dabei bedienen wir uns der open-source Software Python, Octave und Matpower. Unser Beitrag ist zweifältig: Erstens entwickeln wir ein Framework, mit dem der Prozess vom einfachen Modell des Stromnetzes hin zur Lastflussanalyse automatisiert wird. Zusätzlich werden Visualisierungen mit verschiedenen Schwerpunkten erstellt, die ein Verstehen und Arbeiten mit dem System fördern sollen. Außerdem wird zu jeder durchgeführten Lastflussanalyse eine umfangreicher Datensatz erzeugt, der die Grundelage für weitere Analysen bilden kann. Zweitens bringen wir das Framework für die Monate Jänner und Juli 2019 zur Anwendung und lösen dabei für jedes 15-Minuten-Intervall das Lastflussproblem. Die Ergebnisse zeigen ein gemischtes Bild. Einerseits erzielen wir bei der Lösung der Lastflussprobleme eine gute Erfolgsrate von bis zu 85%. Auch bestimmte Zusammenhänge lassen sich erkennen, beispielsweise dass mit einer größeren übertragenen Energie auch die Verluste steigen, die darauf hindeuten, dass die Modellierung in die richtige Richtung geht. Andererseits zeigen sich Schwächen der Modellierung, die Baustellen für zukünftige Forschung offenbaren. Beispielsweise gibt

## *Kurzfassung*

es eine Gruppe von Umspannwerken, die in der Lastflussanalyse mit extremen Werten auffallen. Ein Grund dafür könnte sein, dass das österreichische Stromnetz Teil des europäischen Verbundnetzes ist und mit diesem interagiert. Die Tatsache ist bisher nicht ausreichend in dem Modell abgebildet. Eine weitere Auffälligkeit ist das hohe Maß an verlorener Blindleistung im System. Eine Ursache dafür könnte die lückenhafte Modellierung der Stromerzeugung sein. Während es für die Blindleistungskapazitäten keinerlei Datengrundlage gibt, ist die Stromerzeugung nur als konstant modellierbar. Im echten Stromnetz allerdings wird diese so gesteuert, dass sich Erzeugung und Verbrauch in etwa die Waage halten.



# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Kurzfassung</b>	<b>v</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of Figures</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>5</b>
2.1 The Power Grid . . . . .	5
2.2 Power flow analysis . . . . .	7
2.2.1 System components . . . . .	8
2.2.2 Formulating the power flow problem . . . . .	10
2.2.3 Solving power flow problems in theory . . . . .	11
2.3 Introduction to solving power flow problems with Matpower . . . . .	13
2.3.1 Solving power flow problems with simple Matpower commands . . . . .	13
2.3.2 The Matpower case format in detail . . . . .	15
2.4 Related Work . . . . .	23
<b>3 Methodology</b>	<b>25</b>
3.1 Create power model . . . . .	26
3.2 Construct Matpower case . . . . .	31
3.3 Solve power flow problem . . . . .	37
3.4 Save and process results . . . . .	39
3.5 Visualization and analysis . . . . .	43
<b>4 Results and discussion</b>	<b>53</b>
4.1 January 2019 . . . . .	53
4.2 July 2019 . . . . .	61
<b>5 Conclusion</b>	<b>65</b>
<b>Bibliography</b>	<b>69</b>



# List of Tables

2.1	Different bus types with known and unknown variables . . . . .	12
3.1	Solving algorithms provided by Matpower . . . . .	38
3.2	Real power loss ranges . . . . .	42
3.3	Reactive power loss ranges . . . . .	42
3.4	Voltage magnitude ranges . . . . .	43
3.5	Voltage angle ranges . . . . .	43



# List of Figures

2.1	General layout of electric power grids[JMe08]	6
2.2	Different power injections at bus $j$ . Balance should be given, i.e. $S_{bus}^j + s_d^j - s_g^i = 0$ .	11
2.3	Example row from <i>mpc.bus</i>	16
2.4	The space of possible combinations of an example generator's real and reactive power output depicted as blue area based on [RDZ20].	17
2.5	Columns 1-10 of an example row from <i>mpc.gen</i> .	18
2.6	Columns 11-21 of an example row from <i>mpc.gen</i> .	18
2.7	The $\pi$ branch model as introduced in the Matpower manual [RDZ20].	20
2.8	Example row from <i>mpc.branch</i>	21
2.9	Example row from <i>mpc.gencost</i>	23
3.1	Overview of the methodology, how the power grid base model is transformed into a power flow problem, how it is solved and how the results are processed.	25
3.2	<i>Line</i> class	27
3.3	<i>Substation</i> class	28
3.4	<i>PowerGenerator</i> class	29
3.5	<i>PowerConsumer</i> class	29
3.7	Example of interconnected base class objects	30
3.6	<i>PowerSimulation</i> class	30
3.8	The connection of substations by lines is shown in the upper part of the graphic. The resulting Matpower buses and branches are shown in the lower part.	32
3.9	Example for information displayed when hovering over a substation node.	44
3.10	Example for information displayed when hovering over a line edge.	45
3.11	Example for information displayed when hovering over a power generator node.	45
3.12	Example for information displayed when hovering over a power consumer node.	46
3.13	Visualization of the power system for the purpose of highlighting reactive power losses.	47
3.14	Visualization of the power system for the purpose of highlighting the highest bus voltage magnitude deviation at every substation.	49
3.15	Visualization of the power system for the purpose of highlighting the highest bus voltage angle deviation at every substation.	51
4.1	Average real power load at different times and days of the week in January	53

## List of Figures

4.2	The level of real power load for every simulation interval in January. . . .	54
4.3	Overview of which simulation instances a solution can (green) and cannot (red) be found for. . . . .	55
4.4	Simulation instances arranged according to the given real power load (x-axis) and the total real power losses (y-axis) . . . . .	56
4.5	Overview of the convergence of simulation instances between 17:45 and 19:45. Red boxes correspond to <i>convergence</i> = 0 and green and orange boxes correspond to <i>convergence</i> = 1. Also, orange boxes correspond to the outlier simulation instances in Figure 4.4 . . . . .	57
4.6	Distribution of how often different branches occur in the context of the single largest branch loss ( <i>max_loss_P</i> across all simulation instances. . .	58
4.7	Cutout from the real power loss visualization for the 2019/01/09 19:00 simulation instance, showing the high loss branches. . . . .	59
4.8	Cutout from the voltage magnitude visualization for the 2019/01/31 20:30 simulation instance, showing the high voltage magnitude buses in the border triangle of Switzerland, Germany and Austria. . . . .	60
4.9	Average real power load at different times and days of the week in July . .	62
4.10	The level of real power load for every simulation interval in July. . . . .	62
4.11	Overview of which simulation instances a solution can (green) and cannot (red) be found for. . . . .	63
4.12	Simulation instances arranged according to the given real power load (x-axis) and the total real power losses (y-axis) . . . . .	64
4.13	Distribution of how often different branches occur in the context of the single largest branch loss ( <i>max_loss_P</i> across all simulation instances.) .	64

# 1 Introduction

Electrical energy is one of the main pillars of our civilization as we know it today. Almost all areas of modern day life depend on the supply of electricity, ranging from mundane things like artificial light to complex communication systems such as the internet. Since the industrial revolution in the 19th century, there has been a steady electrification of industry, agriculture, infrastructure, communication and the everyday life of every individual. Hand in hand with this is the growing demand for electrical energy. From 1980 to 2022, it has tripled worldwide to more than 25,000 terawatt hours in 2022[Sta23] and demand is expected to continue to rise. In addition to advancing automation and digitization, the transport transition to e-mobility and the sharp rise in the use of artificial intelligence will also have a significant impact.

If you look at the process of electricity supply, it consists of three essential steps: firstly, a sufficient supply of energy sources, such as coal or gas; secondly, the conversion of these energy sources into electricity using power plants; and thirdly, the electrical power grid as a transmission and distribution infrastructure to deliver the electricity to the end users. Considering our dependence on a reliable power supply, it is of great importance to guarantee each of these three steps. Even short power outages can have drastic consequences, e.g. hospitals being unable to work, interrupted cold chains, offline communication systems and traffic chaos due to out of action traffic lights. Some countries struggle with obtaining enough energy sources or with providing sufficient power generation capabilities. South Africa, for example, has been experiencing an energy crisis for several years, with the population and economy suffering from hours-long power cuts almost every day. The reasons are complex, but years of mismanagement and corruption most certainly play a major role [Ngu23]. Most industrialized nations however, do not have such major problems with electricity generation itself since the procurement of energy sources and the provision of power plant capacities can be planned well ahead.

However, ensuring that the power grid is stable in the electrotechnical sense and that the underlying infrastructure is protected from external influences is significantly more challenging. The reasons for this lie in the nature of electricity transmission itself: Firstly, the power grid must be in balance at all times, i.e. the level of electricity generation must match the level of consumption. This is difficult because demand can fluctuate greatly depending on the time of day and other factors. It becomes even more difficult with the increased integration of renewable energies, such as solar and wind energy, whose energy feed into the grid can vary greatly and is hard to plan. Secondly, the transmission and distribution infrastructure is spatially distributed by nature and very large. This makes it difficult to protect it completely from environmental disasters or targeted attacks. Time and time again, it has been shown that even highly developed countries in central Europe have not yet been able to solve this problem conclusively.

## 1 Introduction

On November 4, 2006, for example, there were widespread power outages in Central and Western Europe. These were caused by a scheduled shutdown of a 380 kV line that crosses the river Ems in Lower Saxony in northern Germany to enable the delivery of a cruise ship from the Meyer shipyard. The shutdown caused load shifts in the power grid, which led to overloads on individual power lines and resulted in them being switched off. The power grid was split into several parts and this also affected surrounding countries. According to the German Federal Network Agency and the Union for Coordination of Transmission of Electricity, more than 15 million people in Germany, France, Belgium, Italy, Austria and Spain were temporarily without electricity [UCT07, Bun07]. The main cause was later identified as an untested and violated n-1 criterion, which states that the electricity grid must be able to function even if any one line fails, not leading to major outages.

A more recent example is the power outage in Dresden on September 13, 2021, which affected 300,000 households. It was triggered by only a metal-coated balloon that caused a short circuit in a substation [ONL21].

However, it is not only accidental or environmental triggers that pose a threat to the power grid, but also targeted attacks on the transmission infrastructure. In addition to cyber attacks, attacks on critical infrastructure are another way for hostile actors to destabilize a country or enforce certain interests. A very recent example is the deliberate attack on an electrical transmission tower near the Tesla Gigafactory in Brandenburg, Germany. The failure of this one tower brought production at the Tesla factory to a standstill for several days and caused damage estimated in the high nine-figure millions [Tag24a, Tag24c]. Another example is the Russia-Ukraine conflict, in which Ukraine's energy infrastructure is repeatedly the target of Russian attacks. There was a cyber attack with malware on companies in the Ukrainian energy sector back in 2015 [Age21] and Russia is currently increasingly attacking the energy infrastructure in the Ukraine war [Tag24b]. The aim is always to weaken and destabilize the country.

As these examples show, the robustness and resilience of the energy supply is of central importance. Countries and their energy supply companies must comprehensively protect their critical infrastructure and ensure a largely uninterrupted energy delivery. Both random, unintentional disruptions, such as those caused by storms, and targeted attacks must be prevented and intercepted. Unfortunately, as the examples show, this is often not the case.

In order to analyse the resilience of the grid, a meaningful model is required. Unfortunately, most infrastructure operators rely on security-by-obscurity. This means that they disclose as little information as possible about the power grid in order to make attacks on it more difficult. However, this lack of transparency can be the breeding ground for negligence and mismanagement, as the example of South Africa shows. Every citizen of a country has an interest in ensuring that the critical infrastructure is resistant to disruptions or attacks and functions reliably at all times. It is therefore important to have a certain degree of transparency about the security of supply. However, as there is currently too little information available on the electricity infrastructure, we take an alternative approach in this work by leveraging open-source intelligence (OSINT) to fill the knowledge gap.



OSINT bases on consulting, filtering and processing information from one or multiple openly available and independent data sources. In the context of this work, this is mainly OpenStreetMap (OSM)<sup>1</sup>, an open-source geographic database. This data is collected by volunteers in the community and also contains precise information on electrical power infrastructure. OSM currently has more than 10 million users and has already been used for research purposes in the past[IHK<sup>+</sup>16]. This use of open data brings with it the great advantage of independence from companies in the energy sector. When it comes to constructing an accurate model of the power grid from this data, Klauzer et al. have done great work in the case of the Austrian power grid [KMAHU24]. Their model contains the most important components of the power grid, including power plants, substations, power lines and consumers.

In this work, we extend the model and construct a framework with which power flow analysis (PFA) can be performed. The goal of PFA is to provide a complete description of the grid, taking into account both the topology of the grid and the electrotechnical parameters as well as the power generation and consumption in the system. This work is intended to lay the foundation for future investigations into the robustness and security of the network against targeted attacks and random failures. Power flow analysis is essential for this because power grids are highly complex systems whose stability and functionality depend on a large number of factors. Since even partial overloads of the grid are sufficient to have major negative effects, the safety and security analysis must go down to power flow level.

As tools for building this analysis framework, we rely on the open-source programming language Python<sup>2</sup> as well as on the software GNU Octave<sup>3</sup> and Matpower<sup>4</sup>. Octave is an open-source tool for the numerical calculation of mathematical problems and is comparable to Matlab. Matpower is an open-source tool with which Matlab and Octave can be extended so that power system simulations can be performed and optimization problems can be solved [ZMST11]. Analyzing the power grid model with Matpower yields the complete description of the state of a system. For example, it can be used to identify supply shortages or line overloads. Furthermore, the model can be modified, e.g. power lines can be 'switched off' or power plants can be taken off the grid.

The contribution presented in this work is twofold: First, we design and implement the framework for transforming the power grid base model constructed by Klauzer et al. [KMAHU24] into the Matpower case format to enable power flow analysis. In addition, this framework contains useful utilities that aim to ease and improve working with the data at hand, e.g. visualizations. Second, by consulting different data sources and making considerations we state an initial parameter configuration for constructing Matpower cases from the given model. Furthermore, we analyse the behavior of the resulting power system under various aspects, aiming to provide a reasonable starting point for future research. In condensed form, we derive the following research questions:

---

<sup>1</sup><https://www.openstreetmap.org>

<sup>2</sup><https://www.python.org/>

<sup>3</sup><https://octave.org/>

<sup>4</sup><https://matpower.org/>

## 1 Introduction

1. How well can the given power grid base model constructed with OSM data be translated into the domain of power flow analysis?
2. What are suitable parameter configurations that make the model reflect reality well?
3. What parameters need further investigation and improvement?

The remainder of this work is structured as follows: Chapter 2 gives an introduction to the multitude of areas that are relevant in this work. That is first the power grid and the foundations of electricity transmission. Second, the power flow problem is formally defined, the solution of which is the focus of the power flow analysis performed in this work. Third, a brief introduction to Matpower and its power flow problem solving capabilities is given. Finally, the chapter provides an overview over related work in the field of power grid research combined with open source intelligence. It also deals with different solving methods utilized in power flow analysis. In Chapter 3 we explain the methodology we follow in this work and which is implemented by the framework. This chapter is structured in such a way that an example is used to explain and illustrate the workflow that generates a power flow problem from the power grid base model, finds a solution to that problem with Matpower and then saves and further analyses the results. Furthermore, simulation runs are done for the months of January and July 2019 in order to gain the data foundation for better understanding the system. Chapter 4 then continues with the presentation and discussion of the results, before Chapter 5 concludes this work by giving a summary of our approach and its results.

## 2 Background

This section provides an overview of the concepts that need to be understood in order to read this thesis. We start with the transmission of electrical energy and look at how power grids are designed and what components they consist of. We also explain the power flow problem from electrical engineering, with which electrical systems are described and analysed mathematically. Furthermore, we take an in-depth look at Matpower, an open-access library for Matlab and Octave that can be used to solve such power flow problems. Finally, related work in the field of power grid research combined with open source intelligence is examined. That includes different solving methods utilized in power flow analysis.

### 2.1 The Power Grid

With electric current, charge carriers, usually electrons, are transferred via a conductor. The current strength  $I$  in Amperes (A) measures how much charge is transferred per time. A distinction is made between direct current (DC) and alternating current (AC). While the current always flows in one direction with DC, it changes periodically with AC. The voltage  $U$  is the potential difference between the two end points of the transfer and indicates how difficult it is to transfer a charge carrier. This is measured in Volts (V). The electrical power  $P$ , i.e. work per time, is specified in Watts (W) and is calculated as the product of current and voltage:

$$P = I \cdot U \quad (2.1)$$

In the context of power grids, the order of magnitude 1 kV ( $= 10^3$  V) is used for the voltage and the order of magnitude 1 MW ( $= 10^6$  W) for the power. The ohmic resistance  $R$ , measured in ohms ( $\Omega$ ), is of decisive importance. A charge carrier that is transferred via a conductor experiences resistance. This leads to energy being converted to heat that is released and thus lost from the electrical system. The energy loss  $P_w$  is calculated as

$$P_w = I^2 \cdot R, \quad (2.2)$$

i.e. it is proportional to the resistance  $R$  and the square of the current. When transferring energy, it is desirable to minimize the transfer loss. Therefore, conductors with a low ohmic resistance are used. However, in comparison it is much more effective to reduce the current because for example, reducing the resistance by a factor of two would also reduce the power loss by a factor of two. If, on the other hand, the current is reduced by a factor of two, the power loss is reduced by a factor of four. To achieve the same level of power

## 2 Background

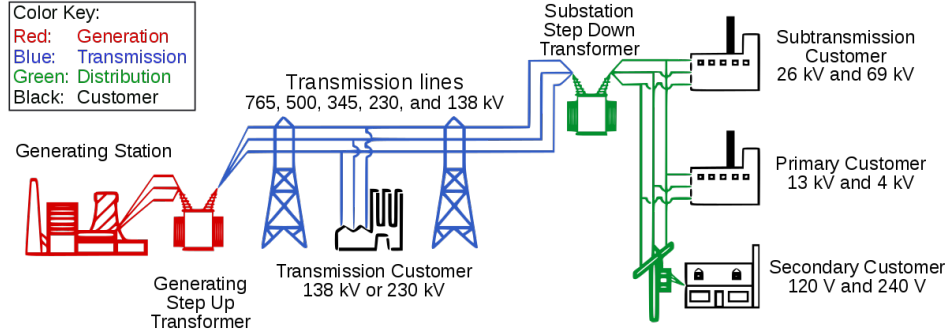


Figure 2.1: General layout of electric power grids[JMe08]

at a low current, the voltage can be increased proportionally according to Equation 2.2. This is the reason why high voltages are used in electrical power transmission.

With cost efficiency in mind, energy delivery in the power grid has two stages, namely energy transmission and energy distribution. Although the highest possible transmission voltage should be used for reasons of efficiency, the hardware, i.e. pylons, power lines, etc., is significantly more expensive and cannot be used everywhere since for safety reasons the clearances have to be larger. In the case of Austria, voltages of 220 kV and 380 kV are used for power transmission. Power lines and substations operating at these high voltage levels are the backbone of the power grid, to which the largest power plants are connected and with which the greatest distances are bridged. Power distribution is meant to bring the energy to the end consumer. Here, the power grid infrastructure becomes more fragmented, the distances bridged are shorter and the voltages used are 110 kV or below.

The process from energy generation to energy consumption is shown from left to right in figure 2.1. Electricity is generated in power plants and then stepped up to extra-high voltage for transmission that is low-loss over long distances. Reaching a substation, the transition from power transmission to power distribution takes place and the voltage is transformed down. From there, the energy is transported to the end consumers, each of which is connected to the distribution network. Finally, the voltage is reduced once again, depending on the end consumer. Normal households for example, use a voltage of 230 V. In order to better understand the electricity grid and its limitations, we take a closer look at the components and at the same time learn how the Austrian power grid is structured.

**Power generation** in this context is the conversion of energy sources or the use of other energy sources to generate electricity. Even though renewable energies, such as wind power and photovoltaics, have been on the rise for years, the majority of power generation is currently produced by large centralized power plants that burn coal, for example, to generate energy. In contrast to wind power and photovoltaics, power generation from power plants is easier to plan and therefore more reliable because it does not depend on external influences such as the weather or the sun. However, that power generation takes place in a few large power plants and therefore inevitably has to be transported

over long distances to the consumers. Most power plants generate alternating current with a voltage of 2.3 kV to 30 kV - depending on the size of the power plant. Before power transmission, the voltage is transformed up to at least 110kV to enable low-loss transmission.

Austria has an annual electricity consumption of approx. 71 GWh, of which 70% is self-generated and 30% is imported. There are a total of 1,329 power generators in Austria with a capacity of more than 1 MW. Just under half of these are wind turbines, but they only contribute 10% of total generation. At just under 57%, the majority of power is generated by hydropower plants[Ene23]. Power generation through the burning of mostly fossil fuels follows in second place with 27.5%. The maximum capacity of all Austrian electricity generators is 28,307 MW and the most powerful power plant is the Simmering thermal power plant with a capacity of 1,328 MW[Ene24, Ene23].

The **power transmission** network in Austria is operated by Austrian Power Grid AG (APG)<sup>1</sup>. The extra high voltage 380 kV and the high voltages 220 kV and 110 kV are used for power transmission in Austria. The backbone of the power grid consists of 3,121 km of 220 kV lines and 2,577 km of 380 kV lines[AP]. Since there can run multiple lines between two transmission towers, there are in total approximately 3500 km of power line infrastructure in Austria, which are connected by nine large grid nodes[(AP24]. These grid nodes consist of powerful transformers, with which the voltage can be changed, and switchgear, with which the power flow, i.e. loads and generation, can be controlled. In addition, there are 67 substations where the **power distribution** networks are connected to the power transmission network by mostly 110 kV lines. Note that while most power lines run above ground, there are also underground cables where space is limited, for instance in urban areas. At the transition from the transmission grid to the distribution grid, the voltage is stepped down because only short distances need to be bridged to the consumers. The connected distribution grids are not operated by APG, but by the Austrian provinces themselves, and are not the subject of this work. It is also important to note that the Austrian grid is part of the European interconnected grid and interacts with it. This is discussed later.

In Austria, only 3-phase alternating current (AC) is used to transmit electricity via these lines, i.e. one line consists of three cables, whereby the voltages on the cables are shifted by 120 degrees in relation to each other. A major advantage of 3-phase power is that, compared to 1-phase power, smaller conductors are needed to transmit the same amount of energy. When dimensioning power cables, a trade-off has to be made between the electrical resistance, which is lower with larger cables, and the material costs and the weight that causes the cables to sag.

## 2.2 Power flow analysis

In order to better analyse an electrical system and its properties, a mathematical description of the system is required that reflects not only the topology of the network but

---

<sup>1</sup><https://www.apg.at/>

## 2 Background

also the electrotechnical relationships. One application for such a description is power flow analysis, in this case for alternating current (AC). The aim of power flow analysis is the precise determination of power flows between energy producers and consumers in an electrical system. The results can be used to examine the power grid for overloaded lines or supply bottlenecks. The power flow analysis consists of two steps: Firstly, the power flow problem is set up from the known variables of the electrical system that results in a non-linear system of equations, which is solved numerically in the second step. There are various approaches and conventions for formulating the power flow problem. Because we solve the power flow problems in this work with Matpower, we roughly follow the definition of the power flow problem that is used in the Matpower manual[RDZ20].

### 2.2.1 System components

A power system consisting of power lines, power plants, transformers, etc. is represented in our definition of the power flow problem with the components: buses, branches, generators, consumers, and shunt elements.

#### Buses

The substations in the power grid correspond to so-called **buses** in power flow problems. However, a substation can include several buses as a bus can only have one specific voltage. If a substation connects to multiple lines with several voltages, there would be a bus for each voltage level in the power flow problem. The voltage at a bus is described by two values, namely the voltage magnitude  $V_m$  and the voltage angle  $\Theta$ , which describe the phase shift with regard to a reference sinusoidal curve. We depict the vector of all bus voltages as  $V$ . Furthermore, each bus has a value for the active power  $P$  and for the reactive power  $Q$  that is feed resp. taken from the network.

#### Branches

Buses are connected to each other via so-called **branches** in the power flow problem. In Matpower's branch model, branches not only represent electrical lines, but also include transformers and phase shifters. While the details are described in section 2.4, note that Ohm's law applies to current flows through branches.

The series impedance for branches is  $z_s = r_s + jx_s$ , where  $r_s$  is the resistance and  $x_s$  is the reactance. Additionally, the reciprocal of the impedance  $z_s$  is the admittance  $y_s = 1/z_s$  that consists of the conductance  $g_s$  and the susceptance  $b$  and is defined as  $y_s = g_s + jb_s$ . Because of the voltage drop between the two ends of a branch, the complex current injections at the two ends of the branch are different and are defined as  $i_t$  for the *to* end and  $i_f$  for the *from* end. Applying Ohm's law, the currents at both ends of a branch  $i$  between two buses can be stated by the equation

$$\begin{bmatrix} i_f \\ i_t \end{bmatrix} = Y_{br}^i \begin{bmatrix} v_f \\ v_t \end{bmatrix} \quad (2.3)$$

Here,  $v_f$  and  $v_t$  are the voltages at the respective buses that are connected by the branch.  $Y_{br}$  is the  $2 \times 2$  shaped branch admittance matrix

$$Y_{br}^i = \begin{bmatrix} y_{ff}^i & y_{ft}^i \\ y_{tf}^i & y_{tt}^i \end{bmatrix}, \quad (2.4)$$

which results from the branch impedance  $z_s$  and other factors. The matrix is of shape  $2 \times 2$  because in Matpower transformers are part of the branch model that are modeled to be at the *from* end of the branches. Therefore, both the current direction and the differences between the branch ends must be taken into account. How exactly  $Y_{br}^i$  is constructed in Matpower is examined in section 2.3. It can be seen that the power flow in the network depends primarily on the bus voltages, as the impedance and thus the admittance is constant. To gather the admittances for all branches of the system in one place, we construct the  $n_l \times 1$  vectors  $Y_{ff}$ ,  $Y_{ft}$ ,  $Y_{tf}$ , and  $Y_{tt}$  where  $n_l$  is the total number of branches. The entries of those vectors are the respective elements from  $Y_{br}^i$  in equation 2.4, e.g. the 1st element of  $Y_{tf}$  equals  $y_{tf}^1$ .

### Power generators

Power generators and consumers are modeled as complex power injections at specific buses. For **generator**  $i$ , the injection is

$$s_g^i = p_g^i + jq_g^i \quad (2.5)$$

With  $n_g$  being the total number of generators, let us define

$$S_g = P_g + jQ_g \quad (2.6)$$

as the  $n_g \times 1$  vector containing all generator injections. Because several generators can be connected to one bus, a helper matrix  $C_g$  is required to assign the generators to the buses.  $C_g$  is of size  $n_b \times n_g$  where  $n_b$  is the total number buses. Now,  $C_g$  is defined such that the  $(i, j)^{th}$  element is 1 if generator  $j$  is connected to bus  $i$  and 0 otherwise[RDZ20]. Combining  $S_g$  and  $C_g$ , the resulting  $n_b \times 1$  vector

$$S_{g,bus} = C_g \cdot S_g \quad (2.7)$$

gives the generators' complex power injections at all buses.

### Power consumers

Power consumers, also referred to as loads or demands, are modelled similarly to generators. The **load** at a bus  $i$  is defined as the power injection

$$s_d^i = p_d^i + jq_d^i \quad (2.8)$$

consisting of the real power demand  $p_d^i$  and the reactive power demand  $q_d^i$ . The  $n_b \times 1$  vector

## 2 Background

$$S_d = P_d + jQ_d \quad (2.9)$$

gives the complex loads at all buses. Note that there is no helper matrix required in this case because there can only be one load for every bus. Also note that dispatchable loads are modeled as negative power generators and appear with a negative sign in equation 2.6.

### Shunt elements

The last basic power grid component in this problem formulation are the shunt elements, short **shunts**. Shunts are fixed impedances attached to buses and are usually used to dissipate current to prevent overloads. Examples of shunts are capacitors and inductors[RDZ20]. The admittance of a shunt at a bus  $i$  is defined as  $y_{sh}^i = g_{sh}^i + jb_{sh}^i$ . Analogous to the loads, we define the  $n_b \times 1$  vector

$$Y_{sh} = G_{sh} + jB_{sh} \quad (2.10)$$

that gives the shunt admittances at all buses.

### 2.2.2 Formulating the power flow problem

Now that we have defined all the components, we can put them together and set up the power flow problem as defined in [RDZ20]. First, we start by constructing the bus admittance matrix  $Y_{bus}$  which consists of both the branches admittances as well as admittances from attached shunt elements. For that, we define the operator  $[\cdot]$  that creates an  $n \times n$  diagonal matrix from an  $n \times 1$  vector by putting the vector entries on the diagonal and being zero elsewhere. Furthermore, we need the  $n_l \times n_b$  connector matrices  $C_f$  and  $C_t$ , that store which branches are connected to which buses. They are defined in such a way that for branch  $i$  the  $(i, j)^{th}$  element of  $C_f$  and the  $(i, k)^{th}$  element of  $C_t$  are 1 if the branch connects from bus  $j$  to bus  $k$  and 0 elsewhere.

As a result, the branch admittances can be set up at the buses.  $Y_f$  is the admittance matrix for the *from* ends of the branches and  $Y_t$  is the admittance matrix for the *to* ends of the matrix:

$$Y_f = [Y_{ff}]C_f + [Y_{ft}]C_t \quad (2.11)$$

$$Y_t = [Y_{tf}]C_f + [Y_{tt}]C_t \quad (2.12)$$

Taking the shunt elements into account, the final  $n_b \times n_b$  bus admittance matrix  $Y_{bus}$  is defined as

$$Y_{bus} = C_f^\top Y_f + C_t^\top Y_t + [Y_{sh}] \quad (2.13)$$

Given the bus admittance matrix  $Y_{bus}$  we are now able to relate the complex current injections  $I_{bus}$  at every bus to the vector of bus voltages  $V$ :



$$I_{bus} = Y_{bus}V \quad (2.14)$$

Now we can go one step further and express the complex power injection at every bus  $S_{bus}$  as a function of the bus voltages  $V$ :

$$S_{bus}(V) = [V]I_{bus}^* \stackrel{2.14}{=} [V]Y_{bus}^*V^* \quad (2.15)$$

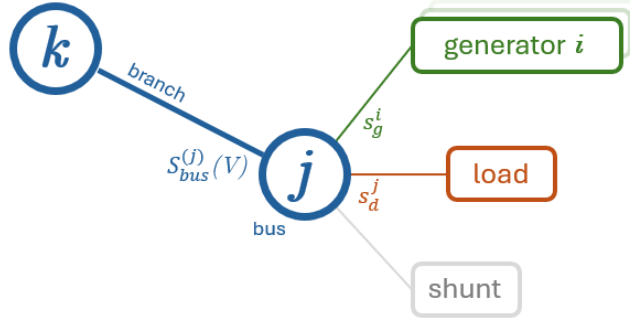


Figure 2.2: Different power injections at bus  $j$ . Balance should be given, i.e.  $S_{bus}^j + s_d^j - s_g^i = 0$ .

Furthermore, we require that for every bus the sum of all incoming power flows is equal to the sum of all outgoing power flows. In addition to the complex bus injections  $S_{bus}$ , we must also take into account the power injections from the generators  $S_g$  and the loads  $S_d$ . Combined with each other, these nodal power equations can be expressed as a function

$$g_S(V, S_g) = S_{bus}(V) + S_d - C_g S_g \stackrel{!}{=} 0 \quad (2.16)$$

that depends on the bus voltages  $V$  and generator power injections  $S_g$ . Keeping  $S_g$  variable allows for solving the problem for different power generation profiles.

### 2.2.3 Solving power flow problems in theory

For solving the power flow problem, we first split up the nodal power equations (2.16) including the corresponding complex power injections into their real and reactive components, yielding the function  $g_P$  for the real power balance

$$g_P(\Theta, V_m, P_g) = P_{bus}(\Theta, V_m) + P_d - C_g P_g \stackrel{!}{=} 0 \quad (2.17)$$

and  $g_Q$  for the reactive power balance

$$g_Q(\Theta, V_m, P_g) = Q_{bus}(\Theta, V_m) + Q_d - C_g Q_g \stackrel{!}{=} 0 \quad (2.18)$$

Also note that  $V$  is now depicted as the combination of both the voltage magnitude  $V_m$  and the voltage angle  $\Theta$  at the respective bus. Furthermore, we introduce a distinction

## 2 Background

Bus type	Known variables	Unknown variables	Index set
Reference bus	$V_m, \Theta$	$P_g, Q_g$	$B_{ref}$
PV bus	$P_g, V_m$	$Q_g, \Theta$	$B_{PV}$
PQ bus	$P_d, Q_d$	$V_m, \Theta$	$B_{PQ}$

Table 2.1: Different bus types with known and unknown variables

between different buses, namely between PV and PQ buses as well as a single reference bus (also called slack bus), depending on which of the four variables  $V_m$ ,  $\Theta$ ,  $P_g$ , and  $Q_g$  are considered to be given.

**PV buses** are generator buses to which at least one generator, e.g. a power plant, is connected. Both the amount of power generated by the generator and the voltage level at the feed-in are given by the power plant and can be regarded as constant in the power flow problem. The management of the power plant output and feed-in is carried out elsewhere and is not part of the power flow problem. Therefore,  $P_g$  and  $V_m$  are given for PV buses (hence the name), while  $\Theta$  and  $Q_g$  are to be determined.

**PQ buses** are pure consumer buses, i.e. there is no generator connected to them. Here, both the real power demand  $P_d$  and the reactive power demand  $Q_d$  are known (hence the name PQ bus) and assumed to be constant. Thus, the unknown variables for PQ buses are the voltage magnitude  $V_m$  and the voltage angle  $\Theta$ .

The **reference bus** is a special bus in power flow problems and while you could theoretically define several reference buses, usually only one is used. It is a bus that is connected to a generator, but in comparison to normal PV buses the reference bus has a fixed voltage angle and voltage magnitude. The reason behind this is that the reference bus serves as a reference point for the other buses in the system. Furthermore, the real and reactive power injections  $P_g$  and  $Q_g$  at the reference bus are kept variable. In this way, imbalances in the system, such as overloads or power bottlenecks, can be balanced out and collected at a central location. In order to be able to handle this additional load, the bus to the most powerful generator in the system is usually selected as the reference bus.

Finally, we also define the sets  $B_{ref}$ ,  $B_{PV}$ , and  $B_{PQ}$  which contain the bus indices for the respective bus type. An overview over the different bus types is given in table 2.1.

Given the different bus types, we know the known and unknown variables for the given system and are now able to set up a system of equations that is to be solved. In the first step, we want to determine the voltage angles and magnitudes for the PV and PQ buses. To do this, we take the functions  $g_P$  and  $g_Q$  from Equation 2.17 and 2.18 and only use a certain subset of the power equations. More precisely, we consider the real power equations for all PV and PQ buses as well as the reactive power equations for only the PQ buses. Written as a function this yields

$$g(x) = \begin{bmatrix} g_P^{\{i\}}(\Theta, V_m, P_g) \\ g_Q^{\{j\}}(\Theta, V_m, Q_g) \end{bmatrix} \stackrel{!}{=} 0 \quad \begin{array}{l} \forall i \in B_{PV} \cup B_{PQ}, \\ \forall j \in B_{PQ}. \end{array} \quad (2.19)$$

with  $x$  being the the vector of unknown variables

$$x = \begin{cases} \Theta_{\{i\}} & \forall i \notin B_{ref} \\ V_m & \forall j \in B_{PQ} \end{cases} \quad (2.20)$$

that is the voltage angle for all PV and PQ buses as well as the voltage magnitude for all PQ buses. The result is a system of  $n_{pq} + 2n_{pv}$  nonlinear equations with  $n_{pq}$  being the total number of PQ buses and  $n_{pv}$  being the number of PV buses. Consequently, the system has  $n_{pq} + 2n_{pv}$  unknown variables since for every PQ bus both the voltage magnitude and the voltage angle are unknown, while for PV buses only the real power balance equations are looked at in this step so only the voltage angle is unknown. Obtaining the voltage magnitudes and voltage angles by solving equation 2.19 for  $x$  makes it possible to solve the real and reactive power balance equations of the reference bus for  $P_g$  and  $Q_g$  respectively. Furthermore, the reactive power balance equations for all PV buses can be solved for  $Q_g$ . However, since the power balance equations are of nonlinear nature, solving them is difficult and a non-trivial problem. For that reason, iterative solving algorithms are deployed that might not give the perfect solution but compute solutions that are reasonable close. Since solving the power balance equations for the power flow problem equals finding the zeros of the function, the Newton-Raphson-algorithm has been a popular algorithm of choice.

## 2.3 Introduction to solving power flow problems with Matpower

In Matpower, electrical systems that are to be analysed are called Matpower cases. A Matpower case is saved in a case file, which is an .M or .MAT file. These case files can be edited with any conventional text editor. To use the Matpower functions and to perform simulations or analyses for a case, best possibly Matlab or Octave are used. We use Octave almost exclusively in this work. If Matlab is used, e.g. for a few functions that Octave does not have, this will be explicitly stated. In the following, we will first look at how to solve power flow problems with a few Matpower commands. Afterwards, we will take a closer look at the Matpower case format.

### 2.3.1 Solving power flow problems with simple Matpower commands

The first step of performing a power flow analysis with Matpower is loading the case file into Octave:

```
>> mpc = loadcase(test_case)
```

Here, *test\_case* is the name of an .m-file and *mpc* is a struct data structure that contains the case data. Information about the loaded case can be printed to the terminal by:

```
>> case_info(mpc)
```

Before running any solving algorithms on the case, an options object can be constructed to configure the algorithm. Although Matpower has default values for the parameters of most functions, it is recommended to set values which best fit to the problem at hand. In Matpower, the options are managed in a specific option struct, which can be changed and

## 2 Background

extended as required to be used in different functions<sup>2</sup>. For example, a different solving algorithm than the default one can be set for solving our first power flow problem:

```
>> mpopt = mpoption("pf.alg", "FDXB")
```

would set the algorithm to the fast-decoupled power flow method (instead of the default Newton-Raphson method). Running a simple power flow solver is done by calling *runpf*<sup>3</sup>:

```
>> results = runpf(mpc, mpoption)
```

*results* is a struct that contains both the solved Matpower case as well as information about whether a solution could be found. Optionally, the results can be returned already decomposed into the different components:

```
>> [baseMVA, bus, gen, branch, success, et] =  
runpf(mpc, mpoption)
```

Here, *baseMVA*, *bus*, *gen*, and *branch* is the solved case data, but with the difference that before unknown quantities (e.g. voltages angles and magnitudes) are now known and filled in. *success* indicates whether a solution could be found (1 = successful, 0 = unsuccessful) and *et* is the elapsed time in seconds. Similar to loading a case from a file, you can also save a case back to a file. With our solved case this would be done by

```
>> savecase(file_name, baseMVA, bus, gen, branch)
```

Furthermore, running *runpf* prints an overview of the both case and the results to the terminal. This output can also be saved to a file with

```
>> results_file = fopen(''+save_path_results+'.m',  
"w")  
>> print_opt = mpoption("out.all", 1)  
>> printpf(results, results_file, print_opt)  
>> fclose(results_file)
```

Another useful function is *runcpf*<sup>4</sup> that runs a continuation power flow algorithm. Here, the goal is to determine how much load the given system can handle. For that, both a starting load configuration and a target load configuration are given. To illustrate this, we take our example Matpower case and zero out the loads as well as the generation

```
>> define_constants  
>> mpbase = mpc  
>> mpbase.bus(:, PD) = 0  
>> mpbase.bus(:, QD) = 0  
>> mpbase.gen(:, PG) = 0
```

Afterwards, we run the continuation power flow with *mpbase* as base configuration and *mpc* as target configuration:

```
>> results = runcpf(mpbase, mpctarget, mpopt)
```

The results struct now contains, among other things, the maximum load scaling factor *results.cpf.max\_lam*. A factor of 1 corresponds to the load configuration of the target configuration, here the one of *mpc*.

Another class of problems that can be solved with Matpower are the optimal power flow

---

<sup>2</sup><https://matpower.org/docs/ref/matpower7.1/lib/mpoption.html>

<sup>3</sup><https://matpower.org/docs/ref/matpower7.1/lib/runpf.html>

<sup>4</sup><https://matpower.org/docs/ref/matpower7.1/lib/runcpf.html>

### 2.3 Introduction to solving power flow problems with Matpower

problems. In general, optimal power flows are an extension of the power flow problem with the addition of a constrained optimization function. For the standard AC optimal power flow problem, typical constraints are branch flow limits and the optimization function is about the costs of power generation done by the generators. For solving such problems, Matpower provides the function *runopf*<sup>5</sup>, which can be configured as required. Similarly to *runpf*, it is called by:

```
>> results = runopf(mpc, mpooption)
```

#### 2.3.2 The Matpower case format in detail

A Matpower case consists of a struct data structure that by convention is named *mpc*. First of all, this struct contains the version of the case format used. Currently, there are versions 1 and 2 available with version 2 being the latest one. Upon loading a case file into Octave, cases of case format version 1 are automatically converted to version 2. In this work only version 2 is used, i.e. *mpc.version* = '2' for every case.

Next, *mpc* contains information about the electrical system. This is *mpc.baseMVA*, which is a scalar, and then the matrices *mpc.bus*, *mpc.gen*, *mpc.branch* and optionally *mpc.gencost*. The matrices respectively contain the information about the buses, generators and branches from the power flow problem in Section 2.2. However, they not only contain absolute numbers as entries, but also values that are expressed in the per-unit system (abbreviated p.u.). For this reason, we briefly introduce the p.u. system before looking at the structure of the matrices.

In the p.u. system, quantities are not represented in absolute numbers, but in relation to a base value. For example, a voltage  $U$  could have an absolute value of 165,000 V. With a base voltage  $P_{base} = 110,000V$ , the voltage per-unit would be  $V_{pu} = \frac{165,000V}{P_{base}} = 1.5$ . Getting from voltage p.u. to the absolute voltage would be done by multiplying the p.u. value with the base value, e.g.  $V = 1.5 \cdot 110,000V = 165,000V$ . In power systems, the p.u. system is used for voltage and power quantities. One of the advantages is that p.u. values on different sides of transformers can be compared with each other despite different voltage levels. In Matpower cases, the p.u. base value for the reactive power is the same everywhere. That is the reason for it to have an extra field in the *mpc* struct, namely *mpc.baseMVA*. In this work, the value of this field is set to 100 MVA (Megavolt Ampere), i.e. *mpc.baseMVA* = 100, for every Matpower case.

We proceed by describing the matrices for the system components starting with the bus matrix.

#### **mpc.bus**

The bus matrix *mpc.bus* contains the buses that were introduced in Subsection 2.2.1. Additionally, shunt elements that are connected to buses are also part of the bus matrix. *mpc.bus* has a row for every bus in the power system, whereby the information for bus  $i$  is stored in the  $i^{th}$  row of *mpc.bus*. Consequently, *mpc.bus* has  $n_b$  rows and 13 columns. An example for a row of *mpc.bus* is shown in Figure 2.3.

<sup>5</sup><https://matpower.org/docs/ref/matpower7.1/lib/runopf.html>

## 2 Background

```
%% bus data
% bus_i type Pd Qd Gs Bs area Vm Va baseKV zone Vmax Vmin
19 1 10.3036309 1.03036309 0 0 1 1 0 110 1 1.11 0.95;
```

Figure 2.3: Example row from *mpc.bus*

The columns of *mpc.bus* are described below, with the column number in brackets after the column name.

**bus\_i (1)** Bus number that is a positive integer and ascending starting from 1. Also corresponds to the row number.

**type (2)** Defines the bus type introduced in Subsection 2.2.3: 1 = PQ, 2 = PV, 3 = reference, 4 = isolated. Isolated buses are defined as buses without connected branches.

**Pd (3)** The real power demand injected at the given bus in MW.

**Qd (4)** The reactive power demand injected at the given bus in MVar. Note that this value is not represented in p.u.

**Gs (5)** The conductance of the attached shunt element. The given value is to be interpreted as the real power (in MW) that is deduced from the bus by the shunt at a voltage of  $V_m = 1.0$  p.u. and voltage angle  $\Theta = 0$ . The base value for the bus voltage is the *baseKV* given in column 10.

**Bs (6)** The susceptance of the attached shunt element. The given value is to be interpreted as the reactive power (in MVar) that is injected at that bus by the shunt at a voltage magnitude of  $V_m = 1.0$  p.u. and voltage angle of  $\Theta = 0$ . The base value for the bus voltage is the *baseKV* given in column 10.

**area (7)** Positive integer that acts as area number of the given bus. Buses in a power system can be assigned to areas for managing load and generation area-wise.

**Vm (8)** The voltage magnitude  $V_m$  at the given bus. This value is expressed in the p.u. system with the *baseKV* in column 10 as base value.

**Va (9)** The voltage angle  $\Theta$  at the given bus. The unit is degrees, thus ranging from -360 to 360.

**baseKV (10)** The base voltage in kV at the given bus.

**zone (11)** Positive integer that acts as zone identifier for the given bus. This value is not used in Matpower, but is included for compatibility of the case format with other formats.

**Vmax (12)** Upper bound for the voltage magnitude  $V_m$  at the given bus. The value is expressed in the p.u. system with the *baseKV* in column 10 as base value. This field is mostly used for optimal power flow analysis.

**Vmin (13)** Lower bound for the voltage magnitude  $V_m$  at the given bus. The value is expressed in the p.u. system with the *baseKV* in column 10 as base value. This field is mostly used for optimal power flow analysis.

### mpc.gen

The generator matrix *mpc.gen* contains the power generators that were introduced in Subsection 2.2.1. It has a row for every generator in the power system. With a number of  $n_g$  generators, this makes  $n_g$  rows and 21 columns. Also note that there are no unique generator identifiers. Generators are assigned to the bus to which they are connected.

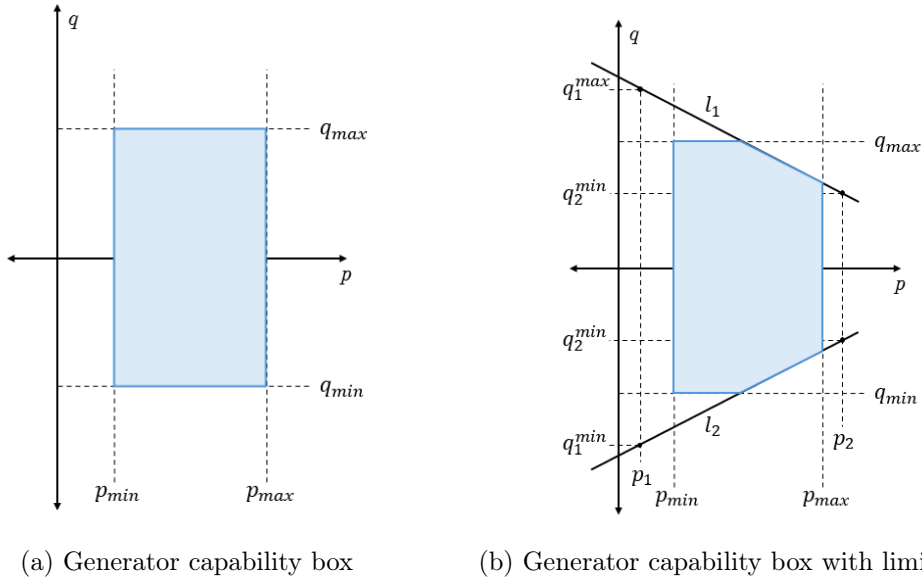


Figure 2.4: The space of possible combinations of an example generator's real and reactive power output depicted as blue area based on [RDZ20].

The columns 11-16 are used for including generator capability curves into the model. Without those columns, Matpower would model the generator capability as a box which shown in 2.4a. The points contained in the blue area are possible combinations of real power and reactive power output. In reality however, generators cannot deliver the maximum real and the maximum reactive power at the same time. Capability curves

## 2 Background

are used to model such limitations. In Matpower, such curves are modeled as lines that narrow the space of possible generator power outputs, shown in 2.4b. These restrictive lines  $l_1$  and  $l_2$  can be clearly defined by the values  $p_1$ ,  $p_2$ ,  $q_1^{min}$ ,  $q_1^{max}$ ,  $q_2^{min}$ , and  $q_2^{max}$  provided in *mpc.gen*. The columns 17-20 are used by the Matpower Optimal Scheduling Tool (MOST)<sup>6</sup> and are not described in detail here. MOST is used to solve steady-state electric power scheduling problems and builds on Matpower and its case format. However, incorporating power generation scheduling exceeds the scope of this work and can be subject to future work. An example for a row in *mpc.gen* is given in Figure 2.5 and 2.6.

Figure 2.5: Columns 1-10 of an example row from *mpc.gen*.

```
%% generator data
% bus Pg Qg Qmax Qmin Vg mBase status Pmax Pmin
211 15.3 1.7 1.7 -1.7 1 17.0847886 1 17 0

% Pc1 Pc2 Qc1min Qc1max Qc2min Qc2max ramp_agc ramp_l0 ramp_30 ramp_q apf
0 0 0 0 0 0 0 0 0 0 0;
```

Figure 2.6: Columns 11-21 of an example row from *mpc.gen*.

The columns of *mpc.gen* are described below, with the column number in brackets after the column name.

**bus (1)** Gives the corresponding *bus\_i* from *mpc.bus* for the bus to which the generator is connected. Note that multiple generators can be connected to the same bus.

**Pg (2)** The effective real power output of the generator given in MW.

**Qg (3)** The effective reactive power output of the generator given in MVar.

**Qmax (4)** The maximum reactive power output of the generator given in MVar. Corresponds to  $q_{max}$  in Figure 2.4a and 2.4b. This parameter is only used in the optimal power flow analysis.

**Qmin (5)** The minimum reactive power output of the generator given in MVar. Corresponds to  $q_{min}$  in Figure 2.4a and 2.4b. This parameter is only used in the optimal power flow analysis.

**Vg (6)** The voltage magnitude at the given bus. This value is expressed in the p.u. system with the *baseKV*(10) from the connected bus as base value. The voltage magnitude of generators is assumed to be given and can be set by setting the baseKV at the bus (since those buses are by definition PV buses).

<sup>6</sup><https://github.com/MATPOWER/most>



### 2.3 Introduction to solving power flow problems with Matpower

**mBase (7)** The overall capacity rating for the given generator given in MVA. It is defined as the amount of the vector of active and reactive power:  $mBase = \sqrt{P_{max}^2 + Q_{max}^2}$ . Among other things, this value is used in power flow simulations with dynamic power generation. The default value corresponds to *mpc.baseMVA*.

**status (8)** Status of the generator. If *status* > 0 then generator is in-service. For *status* ≤ 0 then generator is out-of-service.

**Pmax (9)** The maximum real power output of the generator given in MW. Corresponds to  $p_{max}$  in Figure 2.4a and 2.4b. This parameter is only used in the optimal power flow analysis.

**Pmin (10)** The minimum real power output of the generator given in MW. Corresponds to  $p_{min}$  in Figure 2.4a and 2.4b. This parameter is only used in the optimal power flow analysis.

**Pc1 (11)** Corresponds to  $p_1$  in Figure 2.4b. The unit is MW.

**Pc2 (12)** Corresponds to  $p_2$  in Figure 2.4b. The unit is MW.

**Qc1min (13)** Corresponds to  $q_1^{min}$  in 2.4b. The unit is MVar.

**Qc1max (14)** Corresponds to  $q_1^{max}$  in 2.4b. The unit is MVar.

**Qc2min (15)** Corresponds to  $q_2^{min}$  in 2.4b. The unit is MVar.

**Qc2max (16)** Corresponds to  $q_2^{max}$  in 2.4b. The unit is MVar.

**ramp\_agc (17)** Used in MOST.

**ramp\_10 (18)** Used in MOST.

**ramp\_30 (19)** Used in MOST.

**ramp\_q (20)** Used in MOST.

**apf (21)** The area participation factor *apf* for a generator is used to define how much of the total generation demand in the area is to be produced by the given generator. Here, the area of interest is the area which the bus connected to the generator is part of.

### mpc.branch

The branch matrix *mpc.branch* contains the branches that were introduced in Subsection 2.2.1. Before we take a closer look at *mpc.branch*, we need to expand the definition from 2.2.1. In Matpower, branches are used for modeling power transmission lines. As underlying transmission model the  $\pi$ -model is used, meaning that the impedance of a line is modeled as a series impedance in the middle of the line. Furthermore, the Matpower branch model not only includes transmission lines but also transformers and phase shifters. A circuit diagram is depicted in Figure 2.7. The from end of the branch is shown on the left-hand side of the diagram with the voltage  $v_f$  and the injected current  $i_f$ . Similarly, the to end of the branch is on the right-hand side of the diagram with the voltage  $v_t$  and the current  $i_t$ . The series branch impedance  $z_s$  is expressed as admittance  $y_s = 1/z_s$  (1). Transformers and phase shifters are assumed to always be at the *from* end of a branch (2). Here,  $\tau$  is the tap ratio that decides the voltage transformation.  $\Theta_{shift}$  is the difference in voltage angle introduced by the phase shifter. Furthermore, each end of the branch has half of the branch susceptance  $b_c$  (3).

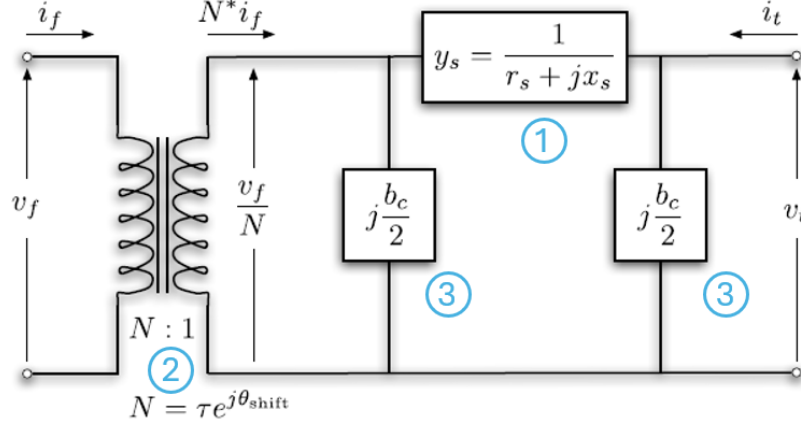


Figure 2.7: The  $\pi$  branch model as introduced in the Matpower manual [RDZ20].

As mentioned in Subsection 2.2.1, the current at the *from* and *to* ends of the branch can be computed as:

$$\begin{bmatrix} i_f \\ i_t \end{bmatrix} = Y_{br}^i \begin{bmatrix} v_f \\ v_t \end{bmatrix} \quad (2.21)$$

Having in mind the  $\pi$  branch model from Figure 2.7, we can fully define the branch admittance matrix  $Y_{br}$  as

$$Y_{br} = \begin{bmatrix} (y_s + j\frac{b_c}{2})\frac{1}{\tau^2} & -y_s\frac{1}{\tau e^{-j\Theta_{shift}}} \\ -y_s\frac{1}{\tau e^{j\Theta_{shift}}} & y_s + j\frac{b_c}{2} \end{bmatrix} \quad (2.22)$$

Before continuing with the branch matrix, we define the base impedance  $Z_B$ , as some

### 2.3 Introduction to solving power flow problems with Matpower

branch values are expressed in the p.u. system. Combining the base voltage  $V$  and the base power  $mpc.baseMVA$ , the base impedance is computed as

$$Z_B = \frac{V^2}{mpc.baseMVA} \Omega \quad (2.23)$$

Continuing with the branch matrix, *mpc.branch* now combines all information about the transmission lines as well as the transformers and phase shifters. It has a row for every branch in the power system. With a number of  $n_l$  branches, this makes  $n_l$  rows and 13 columns. Also note that there are no unique branch identifiers. Branches can be identified by the combination of the two buses that are connected by the branch, defined in the columns (1) and (2) of *mpc.branch*. An example for a row in *mpc.branch* is given in Figure 2.8.

```
%% branch data
% fbus tbus r x b rateA rateB rateC ratio angle status angmin angmax
 46 47 0.00045455 0.03 0 0 0 0 0.5 0 1 -360 360;
```

Figure 2.8: Example row from *mpc.branch*

The column definitions for *mpc.branch* are the following:

**fbus (1)** The bus number at the *from* end of the branch. Corresponds to the bus number from *mpc.bus*.

**tbus (2)** The bus number at the *to* end of the branch. Corresponds to the bus number from *mpc.bus*.

**r (3)** The branch resistance that is expressed in p.u. with  $Z_B$  from equation 2.23 as base value.

**x (4)** The branch reactance that is expressed in p.u. with  $Z_B$  as base value.

**b (5)** The branch susceptance that is expressed in p.u. with  $Z_B$  as base value.

**rateA (6)** *rateA* is a long term rating used for branch flow limits. This field is only used in optimal power flow analysis. By default, this limit is for reactive power and thus expressed in MVA. The meaning of this field can be switched to real power (MW) or to current ( $kA \cdot V_{basekV}$ ) by modifying the *opf.flow\_lim* in Matpower.

**rateB (7)** *rateB* is a short term rating used for branch flow limits. This field is only used in optimal power flow analysis. By default, this limit is for reactive power and thus expressed in MVA. The meaning of this field can be switched to real power (MW) or to current ( $kA \cdot V_{basekV}$ ) by modifying the *opf.flow\_lim* in Matpower.

## 2 Background

**rateC (8)** *rateC* is a emergency rating used for branch flow limits. This field is only used in optimal power flow analysis. By default, this limit is for reactive power and thus expressed in MVA. The meaning of this field can be switched to real power (MW) or to current ( $kA \cdot V_{basekV}$ ) by modifying the *opf.flow\_lim* in Matpower.

**ratio (9)** States the ratio between the nominal voltages  $v_f$  and  $v_t$ . A transformer that is included in this branch would step up/down the voltage according to *ratio*. For pure transmission lines without any transformer, *ratio* is 0 by definition (which mathematically would be equal to a transformer with *ratio* = 1).

**angle (10)** The angle phase shift  $\Theta_{shift}$  performed by the transformer in degrees. Here, a positive value stands for a delay.

**status (11)** The branch status. If *status* is 1 then the branch is in-service. If *status* is 0 then the branch is out-of-service.

**angmin (12)** The minimum difference between the voltage angles at the *from* and *to* ends of the branch in degrees. If *angmin*  $\leq -360$  degrees then there is no lower bound for the angle difference. If *angmin* = *angmax* = 0 then the voltage angle difference is unconstrained. This field is only used for optimal power flow analysis.

**angmax (13)** The maximum difference between the voltage angles at the *from* and *to* ends of the branch in degrees. Similarly to *angmin*, if *angmax*  $\geq 360$  then there is no upper bound for the angle difference. If *angmin* = *angmax* = 0 then the voltage angle difference is unconstrained. This field is only used for optimal power flow analysis.

### mpc.gencost (optional)

Adding *mpc.gencost* gives the option of adding the costs for power generation to the model. This includes the start-up and shutdown costs as well as the costs for ongoing operation. *mpc.gencost* has either  $n_g$  or  $2n_g$  rows. In the first case, only the costs for real power generation are considered where the  $i^{th}$  row of *mpc.gencost* corresponds to the  $i^{th}$  generator in *mpc.gen*. In the second case, in addition to the costs for real power generation, the last  $n_g$  rows give the costs for reactive power generation. The number of columns is variable, but at least 5. Column 5 and the following columns state the parameters for the cost function. There are two options for the cost functions. The first option is a piecewise linear (or segmented) cost function. The given parameters are  $p_1, f_1, p_2, f_2, \dots, p_n, f_n$  from which the points  $(p_1, f_1), (p_2, f_2), \dots, (p_n, f_n)$  are formed. These points are connected by straight lines, constructing the graph of a function  $f(p)$  where the  $p$ -coordinate gives the generated power in MW (or MVar) and the  $f(p)$ -coordinate gives the operating cost in US\$/hour. The second option is a polynomial cost function. Here, the operation cost for a given power generation is returned by the polynomial  $f(p) = c_n p^n + \dots + c_1 p + c_0$ . An example for a row in *mpc.gencost* is given in Figure 2.9.

```

%% generator cost data
% 1 startup shutdown n x1 y1 ... xn yn
% 2 startup shutdown n c(n-1) ... c0
mpc.gencost = [
    2 0 0 3 0.02 2 0;

```

Figure 2.9: Example row from *mpc.gencost*

The columns definitions for *mpc.gencost* are the following:

**model (1)** States the type of cost function. If *model* = 1 then the cost function is piecewise linear. If *model* = 2 then the cost function is polynomial.

**startup (2)** The generator startup cost in US \$.

**shutdown (3)** The generator shutdown cost in US \$.

**n (4)** Positive integer that states the number of parameters for the cost function. In case of *model* = 1, *n* corresponds to *n* + 1 data points (totaling to  $2(n + 1)$  parameters) for the piecewise linear function with *n* segments. In case of *model* = 2, *n* corresponds to the number of *n* + 1 coefficients  $c_n, \dots, c_0$  for the polynomial.

**parameters (5)** For *model* = 1, column 5 and the following columns contain the parameters in the order  $p_1 f_1 \dots p_n f_n$ . For *model* = 2, column 5 contains  $c_n$  and the following columns contain  $c_{n-1} \dots c_0$ .

## 2.4 Related Work

The idea of using open source intelligence (OSINT) for research into energy infrastructure is not new. For instance, the *SciGRID* project funded by the *German Federal Ministry of Education and Research* leverages OSM data to build an open source model of the European electrical transmission grid [Med14]. It does so, in an automated fashion, by filtering the OSM database and extracting objects and relations with a *power* tag. A similar approach is taken in the *osmTGmod* project [Sch17]. However, in contrast to *SciGRID*, an attempt is made here to add missing power lines and substations that cannot be captured by OSM data (e.g. underground cables) to the power grid model using heuristics. This approach is also used in *GridKit* [Wie16], which is based on *SciGRID*. However, according to Klauzer et al. [KMAHU24], these three approaches of converting OSM data into a realistic power grid model all lack the consideration of power generation and consumption in the grid. Although Müller et al. extend the *osmTGmod* approach to include power generation and consumption [MWK<sup>+</sup>18], they do not validate the underlying power grid topology. Now, using the example of the Austrian power grid, Klauzer et al. close this gap in research by combining the derivation of a

## 2 Background

power grid model from OSM data with the development of a procedure to validate the correctness of said model [KMAHU24]. The first step is similar to the other approaches, as it consists of filtering the OSM database for objects with the *power* attribute. The second step brings something new to the table as Klauzer et al. combine multiple data sources to construct a ground truth that the power grid model can be validated against. This includes Google Street View<sup>7</sup>, coarse-grained data on the electricity infrastructure provided by the transmission and distribution grid operators (e.g. [(AP24], research results from other projects<sup>8</sup> and other publicly accessible information (e.g. [Ene24]. The results of their validation are impressive. For extra-high voltage (380 kV) power lines, they achieve a 100% match between their power grid model and the ground truth. For high voltage (110 kV and 220 kV) power lines it is 97%. For substations, there is a match of 97% for extra-high voltage and a match of 87% for high voltage. High values are also achieved for power plants, with 81% of power plants and thus 87% of national Austrian generation being represented by the model. For power consumption, they achieve with their model that in 97% of cases only 15 km or less lie between the center of the consuming municipality and the nearest substation. Due to the combination of a high-quality model and the existing validation of the same, we decided to use this model as the basis for our work. The same decision was made by A. Hubbermann, whose master's thesis, starting from the model constructed by Klauzer et al., examines the Austrian power grid's robustness against random failures and targeted attacks [Hub23]. In general, it was limited to the graph-theoretical properties of the model and, for the development of various attack scenarios, has primarily focused on the connectivity of the graph and its nodes. In contrast, we take it a step further and perform a power flow analysis, that is considering both the power generation and consumption in the grid as well as the electrical properties of the power system. Our tool of choice for that is Matpower, an open source Matlab extension that offers a wide variety of utilities and algorithms. Matpower itself comes with several data sets from the Polish power grid from the period 1999 - 2008 [MAT15a, MAT15b, MAT15c, MAT15d, MAT15e, MAT15f, MAT15g, MAT15h]. Furthermore, Josz et al. publish and analyse several cases for the French extra-high voltage grid [JFMP16]. When it comes to solving AC power flow problems, as we do in this work, multiple methods have been developed over the years. The Newton-Raphson method [TH67] was found to be the most practical in terms of convergence and robustness [Sto74]. It comes in various types that are compared by Sereeter et al. in [SVW19]. However, Matpower offers further alternative solving methods that were proposed in literature. That includes the Fast Decoupled Load Flow algorithm [SA74] and the Gauss-Seidel method [Ten02].

---

<sup>7</sup><https://www.google.com/streetview/>

<sup>8</sup><https://www.energiemosaik.at/intro>

### 3 Methodology

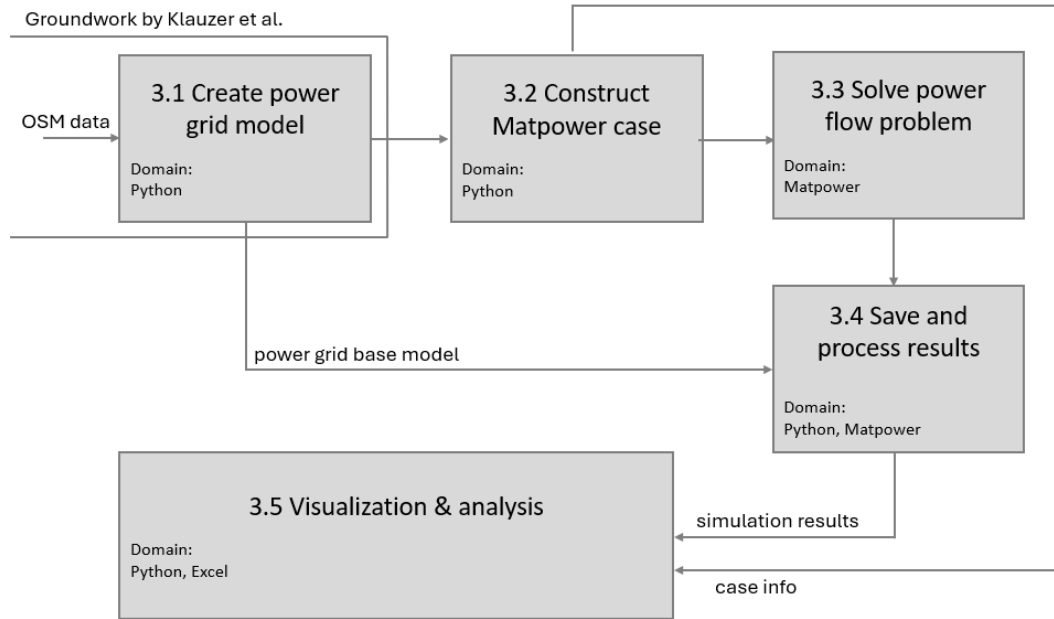


Figure 3.1: Overview of the methodology, how the power grid base model is transformed into a power flow problem, how it is solved and how the results are processed.

In this chapter, we describe our approach to get from a model of the Austrian power grid generated from OSM data to a power flow analysis in Matpower. We also describe how to evaluate and visualize the results of the power flow analysis. An overview of the workflow is given in Figure 3.1. In addition to the data basis used in each section, we also discuss the tools and data structures used.

The overall aim is to investigate whether it is possible to create an accurate model of the Austrian power grid from the data in the freely available OpenStreetMaps database. For that purpose, we implement a framework which automates that process to ease further research. This approach is based on the work done by Klauzer et al.[KMAHU24] by which the power grid base model is constructed for Austria. While in related work, Hubbermann uses that model to examine the power grid for its weaknesses and resistance to disruptions and attacks by means of graph-theoretical methods [Hub23], we want to go one step further and improve the accuracy of the model. In a power grid, not every power line or substation is equally important. Different power lines, for example, can

### 3 Methodology

have different capacities and different loads. The stability of the power grid is subject to a wide range of complex interrelationships. Not only the physical laws underlying electrical energy transmission, but also the quickly changing demand for electricity in the grid make operating the power grid a challenge. To reflect this aspect, we expand the power grid base model so that it incorporates the power demand and generation in the grid. In addition, electrotechnical relationships need to be modeled. To do this, we choose the well-known power flow analysis method, the result of which is a comprehensive description of the system. As tools we choose Python as well as Octave and Matpower. We use Python to convert the existing model of the Austrian power grid into a format that can be understood by Matpower. We can then use Matpower to perform different types of analyses (e.g. power flow/continuation power flow/optimal power flow analysis), for instance for different parameters and simulation time frames. We are interested in how the electrical system or rather the Matpower simulation behaves for different parameters. The aim here is to better understand the process from a model generated from OSM data to an accurate Matpower power flow simulation. We want to identify difficulties and find the best possible parameter configuration that allows the procedure to be used for other data bases and for the power grids of other countries.

The remainder of this chapter is structured as follows: In Section 3.1 we get familiar with the graph representation of the power grid base model as introduced by Klauzer et al. In doing so, we look at the Python code and the data structures used. At the same time, the extensions and improvements we make are described. Then, in Section 3.2, we describe the procedure for constructing a Matpower case from the graph representation. In particular, we explain how the individual electrotechnical parameters are calculated and which data basis we use for this. The result of that section is the Matpower base case which is later evaluated. Section 3.3 deals with the selection of the solving algorithm and its parameters. Section 3.4 deals with the variety of results that can be obtained from the simulation runs. We describe in detail which characteristic values from the results we continue to work with and why. In addition to the results from 3.5, the original power grid model from 3.1 and the case information generated by Matpower from Section 3.2 is also considered here. In doing so, we operate in both the Python and Matpower domains. We also describe how results are saved. Finally, analysis and visualization is discussed in Section 3.5. Following this chapter, we put our framework to work by running simulations for the months of January and July 2019. The results are then presented in the next chapter, namely Chapter 4

#### 3.1 Create power model

The basic model of the Austrian electricity grid is generated with Python from OSM data. This step is the subject of the preliminary work that Klauzer et al. We improve the data basis for the electricity generators included in the model, see Section 3.2. In addition, we use the format introduced by Klauzer et al. for the power grid base model and extend it so that it can also include the results of the Matpower simulations calculated later. The power grid is represented in the model with the help of four different base classes:



### 3.1 Create power model

*Line*, *Substation*, *PowerGenerator*, and *PowerConsumer*. Objects of these classes model the various resources of a power grid and create a model that takes into account both the power transmission itself and the power demand and power generation in the grid. The base classes are described below. A simplified class diagram containing the relevant attributes is shown for each class. The attributes are always sorted in such a way that the normal attributes are at the top, which contain, for example, the electrical parameters or the position in the model. These are followed by attributes that contain one or more objects from one of the other base classes. These attributes are later used to form a coherent graph of the power grid. At the bottom of the attributes are those that are later used to connect the graph of the power grid base model with the information from the (solved) Matpower case. The methods of the base classes are not of importance for understanding the model and are not explained further. The full source code can be viewed at<sup>1</sup>.

#### Line

Power lines are modeled with the *Line* class shown in Figure 3.2. It should be noted that in a real power grid, several electrical systems can be combined in one power line. Each of these systems has its own *Line*-object in this model. The first attribute is the *fid*, which serves as an identifier. This identifier identifies only a power line and can therefore be ambiguous. To distinguish between the different systems of a power line, the *ref* attribute is added. *wires* indicates how many wires the electrical system in question consists of and *voltage* indicates the nominal voltage of the power line. The length attribute specifies the length of the power line in meters. This value is important for calculating the resistance and reactance. Next, the *osm\_id* is the object identifier from the OpenStreetMaps database. This is used later in the visualization to reconstruct the geographical route of power lines. The end points of the power line are given by *substation\_start* and *substation\_end*. These values are to be understood as substation identifier *substation.id*.

The connection to the start and end point of the power line is also made via the attributes *connected\_substation\_1*, which is the substation object of the start point, and *connected\_substation\_2*, which is the substation object of the end point. The last attribute to be mentioned is *result\_branch*, which is of the list type. After the power flow analysis, the branch belonging to the power line from the resolved Matpower case is saved here. The list saved here is identical to the corresponding line from *mpc.branch*. As there is exactly one branch in the Matpower case for each electrical system of a power line, i.e. for each *Line* object, no further differentiation is required. Also, since there are

Line
+ fid: int + osm_id: str + name: str + ref: str + substation_start: int + substation_end: int + voltage: int + length: float + wires: int ----- + connected_substation_1: Substation + connected_substation_2: Substation ----- + result_branch: list ... ...

Figure 3.2: *Line* class

<sup>1</sup><https://gitlab.sba-research.org/johanna/nurzu-load-flow-analysis>

no identifiers for Matpower branches, they cannot be saved here.

## Substation

Substations in the power grid are modelled by the *Substation* class shown in Figure 3.3. The objects of this class form the nodes in the graph representation of the power grid base model. Each substation has a unique identifier with the *id* attribute. The type of *id* is specified here as *str*, but it is effectively of type *int* since only positive integers are taken as value. The second attribute *name* is the name of the substation. This originally comes from OSM and provides information about which substation is given. Next, *connected\_to\_line* gives information about whether or not the given substation is connected to a power line. This attribute can be used to filter for (in the model) isolated substations. The *wkt* attribute is important because it provides information about the geographical position of the substation. It contains coordinates as string, for example *POINT (747049.801890088 5237594.45747121)*. The reference coordinate system used is UTM zone 32N EPSG:32632<sup>2</sup>. Continuing, the *voltage\_specification* attribute is a list of integers that specify which voltage levels occur at a substation. In this model, only the power transmission network is considered and therefore only the voltages 110 kV, 220 kV and 380 kV are encountered. A substation is connected to the other elements of the base model in such a way that it has a list of the corresponding objects in the *connected\_lines*, *connected\_power\_consumers* and *connected\_power\_generators* attributes. In Section 3.2, substations are turned into Matpower buses. A separate bus is created for each voltage level of a substation. The list of all buses of a substation is saved in the *bus\_ids* attribute. The *bus\_ids\_for\_simulation* attribute is now of the *dict* type, which establishes the connection between the voltage levels of a substation and the buses generated for it. In this dictionary, the voltage is the key and the bus id is the value. It can happen that the generated system in Matpower is split into several segments. The attribute *island\_from\_simulation* is used to note which segment the buses of the current substation belong to. The *result\_buses* attribute is used to merge the results of the Matpower simulation back into the base power model. As several buses can belong to a substation, a dictionary is used here. The key is the Matpower *bus\_i* and the value is the corresponding row from *mpc.bus* of the solved Matpower case.

Substation
+ id: str
+ name: str
+ connected_to_line: bool
+ wkt: str
+ voltage_specification: list [int]
-----
+ connected_lines: list [Line]
+ connected_power_consumers: list [PowerConsumer]
+ connected_power_generators: list [PowerGenerator]
-----
+ bus_ids: list [int]
+ bus_ids_for_simulation: dict
+ island_from_simulation: int
+ result_buses: dict
...
...

Figure 3.3: *Substation* class

<sup>2</sup><https://epsg.io/32632>

## PowerGenerator

Power generators are represented in the model by the *PowerGenerator* class which is shown in 3.4. Similar to the *Substation* class, *PowerGenerator* has an *id* attribute of type *str* for unique identification, which is effectively of type *int*. It also has a *name* attribute and, as with the *Line* class, an *osm\_id*, which is the link to the OSM database. As with substations, power generators also have an *wkt* attribute for describing the geographical location. The format and the reference coordinate system are the same as for substations. The first generator-specific attribute is *power*, which defines the real power output capacity of the given generator in MW. Next, the *substation\_id* gives the substation identifier which equals the *substation.id* of the substation to which the generator is attached to. For generators, the attribute *generator\_number\_in\_simulation* saves what generator in *mpc.gen* belongs to the *PowerGenerator* object. Since generators do not have dedicated identifiers in Matpower cases, this value corresponds to the row number. The attribute *connected\_bus\_in\_simulation* corresponds to the *bus\_i* of the bus to which the generator is attached to in the Matpower case. Similar to *Line* objects and branches, the generator from the solved Matpower case is saved in *result\_gen*. Again, the saved list corresponds to the whole row of *mpc.gen*.

PowerGenerator
+ id: str + osm_id: str + name: str + power: float + wkt: str + substation_id: int ----- + generator_number_in_simulation: int + connected_bus_in_simulation: int + result_gen: list ...
...

Figure 3.4: *PowerGenerator* class

## PowerConsumer

Power consumers are represented in the model by the *PowerConsumer* class which is shown in Figure 3.5. They correspond to demands in the power flow problem and are attached to a dedicated substation. Similar to substations and generators, *PowerConsumer* objects have a *name* attribute that makes the connection from model to the real world. The *code* attribute is the Austrian community code that uniquely identifies every community in Austria. Again, a *wkt* attribute gives the geographic location in UTM zone 32N EPSG:32632 format. Next, there are 21 attributes starting from *cons3* up to *cons23*. These attributes represent the annual consumption in the different sectors of power consumption. For example, *cons3* is iron and steel production, private households is *cons22* and agriculture is *cons23*. Calculating the current demand of a power consumer is done by combining the annual consumption with a factor from the load profile (that is introduced in the next paragraph). The connection of a *PowerConsumer* object to its substation is done by the *connected\_substation* attribute.

PowerConsumer
+ name: str + code: str + wkt: str + cons3: float ... + cons23: float ----- + connected_substation: Substation ...
...

Figure 3.5: *PowerConsumer* class

### 3 Methodology

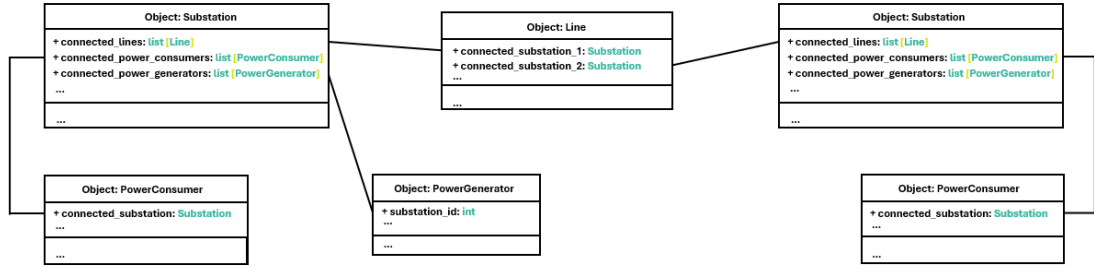


Figure 3.7: Example of interconnected base class objects

### PowerSimulation

The *PowerSimulation* class shown in Figure 3.6 combines the whole power grid base model into one object. It consists of a list for each of the four base classes *Line*, *Substation*, *PowerConsumer*, and *PowerGenerator*. The connections between the base class objects are as described above and are exemplified in Figure 3.7. Furthermore, the *PowerSimulation* class has an attribute *load\_profile* which is a data structure that holds information about the power consumption for every sector throughout the year. For each quarter-hour interval, the load profile indicates the percentage of the annual consumption that is consumed in the interval. The underlying data for the load profile is taken from Energiemosaik Österreich<sup>3</sup>.

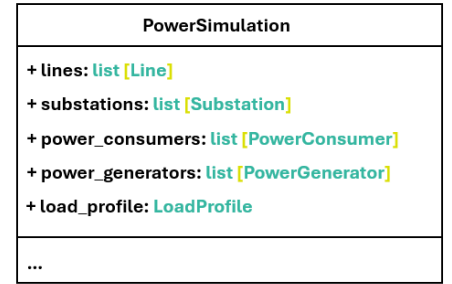


Figure 3.6: *PowerSimulation* class

<sup>3</sup><https://www.energiemosaik.at/>

## 3.2 Construct Matpower case

This section examines how a Matpower case is generated from the power grid base model from Section 3.1. The data structure is constructed in Python. Later, the Python package `oct2py`<sup>4</sup> is used as bridge between Python and the Octave runtime (and thus Matpower). Here, the lists of *Line*, *Substation*, *PowerGenerator* and *PowerConsumer* objects stored in a *PowerSimulation* object are run through and converted into Matpower buses, branches and generators and added to *mpc.bus*, *mpc.branch* or *mpc.gen* accordingly. Filling the Matpower case matrices, there are three types of values: i) values that can be taken directly from the power grid base model, ii) values that can be calculated using the base model or other data sources and iii) values that can only be guessed at first and found out experimentally. In the following, we revisit buses, branches and generators and discuss how values are chosen. At this stage, we derive a Matpower base case that is the basis for future power flow analyses. Also note that we always use version 2 for the Matpower case format and 100 MVA as *mpc.baseMVA*.

### Buses

Substations cannot be converted one-to-one into buses because substations have several voltage levels (i.e. 110 kv, 220 kv or 380 kv). That is the reason for separate buses being generated for each of these voltage levels. An example of the conversion between substation-line connections and bus-branch connections is shown in Figure 3.8.

The following describes what values are used for Matpower buses.

**bus\_i (1)** When iterating over the list of substations, the first *bus\_i* starts at one and is incremented with every new bus.

**type (2)** For substations with at least one power generator, the bus with the highest voltage level of all the buses generated for that substation is defined as PV bus and thus is assigned *type* = 2. Of all PV buses, one single bus is defined as reference bus and is assigned *type* = 3. For the base case, the reference bus belongs to substation *UW Wien Suedost*(178) since the most powerful generator *Kraftwerk Simmering* is attached to it in the power grid base model. This should ensure that the generator is capable of balancing out any load/demand imbalances. Every other bus that is neither PV bus nor reference bus is by definition a PQ bus and is assigned *type* = 1.

**Pd (3)** The real power demand is yielded by summing up the demands for every power consumer that is connected to the substation. In case there are multiple voltage levels and thus multiple buses given at the substation, the power demand is assigned to the bus with the lowest voltage level since those are typically connected to the power distribution network. Because in this work only the power transmission network is examined, the

---

<sup>4</sup><https://pypi.org/project/oct2py/>

### 3 Methodology

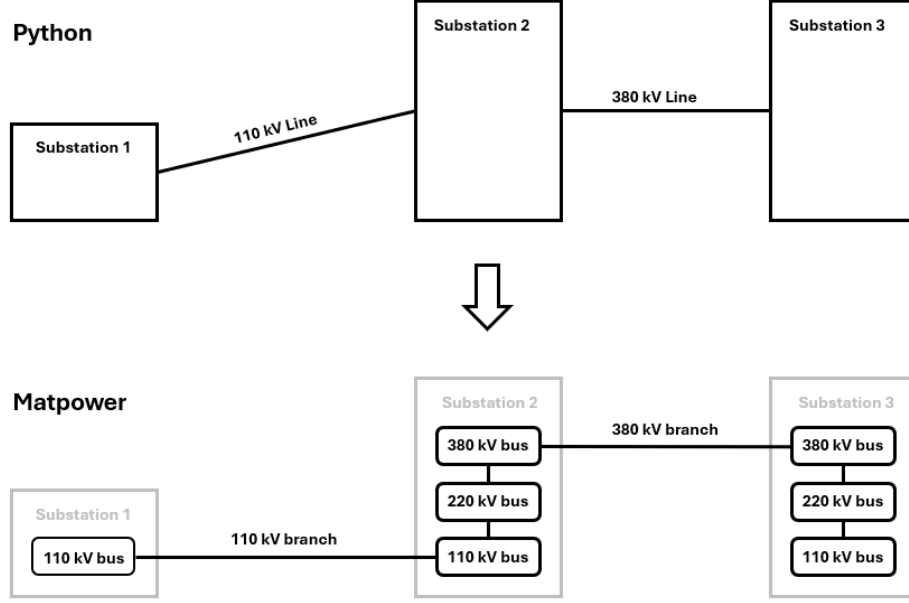


Figure 3.8: The connection of substations by lines is shown in the upper part of the graphic. The resulting Matpower buses and branches are shown in the lower part.

power demands are aggregated and included into the model at those points where the transmission network is connected to distribution networks.

**Qd (4)** The reactive power demand is not part of the power grid base model. We can only estimate this value and do so for the base case by assuming that the reactive power demand corresponds to 10% of the real power demand. Hence,  $Qd = \frac{Pd}{10}$ . Finding more accurate values of  $Qd$  can be the subject of future work.

**Gs (5)** For the base case, we set this value to zero. Considering shunt elements is subject to later experiments.

**Bs (6)** For the base case, we set this value to zero. Considering shunt elements is subject to later experiments.

**area (7)** Advanced load and power generation management exceeds the scope of this work, so every bus is assigned to the same area, i.e.  $area = 1$ .

**Vm (8)** For the voltage we pick  $Vm = 1.0 p.u.$ . For PV buses, this value is fixed. For PQ buses, the voltage is unknown and is to be determined by solving the power flow problem. Here, the given  $Vm$  value is to be understood as initial guess that is needed for

the Newton-Raphson method. For both cases, we pick a voltage of  $1.0 \text{ p.u.}$  for the base case.

**Va (9)** Similar to the voltage magnitude, the initial voltage angle  $Va$  needs to be guessed for all PQ and also all PV buses. In case of the reference bus, the voltage angle is fixed and is set here. For the base case, we pick an angle of zero degrees for all buses. Different starting angles are tried in our experiments.

**baseKV (10)** The base voltage is set to the corresponding voltage level of the substation, showcased in Figure 3.8.

**zone (11)** As mentioned in Subsection 2.3.2, this value is not yet used in Matpower and thus is set to 1.

**Vmax (12)** This value is only used for the optimal power flow analysis and has no impact on the standard power flow problem. For the base case,  $Vmax$  is set to zero.

**Vmin (13)** This value is only used for the optimal power flow analysis and has no impact on the standard power flow problem. For the base case,  $Vmin$  is set to zero.

## Branches

There are two kinds of branches. The first are transmission branches that are constructed from *Line* objects. The second are transformer branches that origin from substations with multiple voltage levels. In that case, buses from the same substation are connected with each other such that each bus is connected to the next lower or higher voltage bus, i.e. 110 kV bus with 220 kV bus and 220 kV bus with 380 kV bus. An example is shown in Figure 3.8. While most branch parameters are constructed in the same way for both type of branches, some parameters are chosen differently. For both transmission and transformer branches, the parameter setup is examined in the following.

**fbus (1)** Transmission branches: Every *Line* object in the base model is visited. With the voltage level at hand, the corresponding bus from *line.connected\_substation\_1* is taken as *fbus*. Transformer branches: When converting a substation with multiple voltage levels, connecting the resulting buses by a branch is done directly after the bus creation. The bus that is created first is set to *fbus* for the branch. The order is of no relevance since the transformation *ratio* (9) is chosen accordingly.

**tbus (2)** Transmission branches: Every *Line* object in the base model is visited. With the voltage level at hand, the corresponding bus from *line.connected\_substation\_2* is taken as *tbus*. Transformer branches: When converting a substation with multiple voltage levels, connecting the resulting buses by a branch is done directly after the bus creation.

### 3 Methodology

The bus that is created second is set to *tbus* for the branch. The order is of no relevance since the transformation *ratio* (9) is chosen accordingly.

**r (3)** For both the transmission branches and the transformer branches, the resistance  $r$  is computed by dividing a specific resistance  $r_s$  by the base resistance  $r_b$ , i.e.  $r = \frac{r_s}{r_b}$ , where  $r_b = \frac{voltage_1 \cdot voltage_2}{100,000,000}$ . For transmission branches  $voltage_1 = voltage_2$  while for transformer branches  $voltage_1$  and  $voltage_2$  correspond to the voltages at the ends of the transformer branches respectively. For transmission branches, the specific resistance  $r_s$  is provided by the power grid base model and can be retrieved from the corresponding *Line*-object. For transformer branches, the specific resistance is set to  $r_s = 0.215$  for the base case.

**x (4)** For both the transmission branches and the transformer branches, the reactance  $x$  is computed by dividing a specific reactance  $x_s$  by the base resistance  $r_b$ , i.e.  $x = \frac{x_s}{r_b}$ , where  $r_b = \frac{voltage_1 \cdot voltage_2}{100,000,000}$ . For transmission branches  $voltage_1 = voltage_2$  while for transformer branches  $voltage_1$  and  $voltage_2$  correspond to the voltages at the ends of the transformer branches respectively. For transmission branches, the specific reactance  $x_s$  is provided by the power grid base model and can be retrieved from the corresponding *Line*-object. For transformer branches, the specific reactance would be set to  $x_s = 16.283$  according to recommended values in literature. However, first experiments show that the resulting reactance  $x$  is unsuitable for our model as well as for its convergence behavior. For that reason, we set  $x = 0.18$  as initial value for the base case. This is further discussed in Chapter 4.

**b (5)** For both transmission and transformer branches the susceptance is set to zero for the base case.

**rateA (6)** This field is only used in optimal power flow problems to define branch flow limits. Such limits are not part of the base case, thus  $rateB = 0$  which means unlimited branch flows are possible.

**rateB (7)** This field is only used in optimal power flow problems to define branch flow limits. Such limits are not part of the base case, thus  $rateA = 0$  which means unlimited branch flows are possible.

**rateC (8)** This field is only used in optimal power flow problems to define branch flow limits. Such limits are not part of the base case, thus  $rateC = 0$  which means unlimited branch flows are possible.

**ratio (9)** Transmission branches: The *ratio* is set to zero, i.e. no transforming takes place. Transformer branches: The transformation ratio is defined as the ratio of the voltage at the *from* bus  $V_f$  to the voltage at the *to* bus  $V_t$ , i.e.  $ratio = \frac{V_f}{V_t}$ .



**angle (10)** Transmission branches: There is no phase shift performed, i.e.  $angle = 0$ . Transformer branches: For the base case, we assume that there is no phase shift introduced by transformer branches, i.e.  $angle = 0$ . This parameter can be subject to further experiments.

**status (11)** For the base case we assume that all branches are in-service, i.e.  $status = 1$ . When we examine the resilience of the power grid, branches can be specifically switched off at this point.

**angmin (12)** Limitations for the maximum voltage angle difference between the *from* and *to* ends of a branch are not used in the standard power flow problem, but are introduced in optimal power flow problems. Hence,  $angmin = 0$  (i.e. unconstrained) for the base case. When setting up an optimal power flow problem, this parameter is revisited.

**angmax (13)** Limitations for the maximum voltage angle difference between the *from* and *to* ends of a branch are not used in the standard power flow problem, but are introduced in optimal power flow problems. Hence,  $angmax = 0$  (i.e. unconstrained) for the base case. When setting up an optimal power flow problem, this parameter is revisited.

## Generators

The generator matrix *mpc.gen* is filled by iterating over all power generators in the power grid base model. How the parameters are set is explained in the following.

**bus (1)** For each visited generator, the corresponding substation is accessed by *power\_generator.connected\_substation*. In case the connected substation has multiple voltage levels, the bus with the highest voltage level is chosen to be connected to the generator.

**Pg (2)** The effective real power output of the generator is taken from the power grid base model where the power output is given in *power\_generator.power*. For the base case however, we only consider generators with a real power output of at least 10 MW. The reason for this is that we want to keep the power system in the base case simple and clear. Furthermore, in regards of generator power output the power grid base model is incomplete as for multiple large generators there is no power specification given. We fill these gaps by consulting the power plant map[Ene24] from *Oesterreichs Energie*, that shows every power plant in Austria with a maximum power capacity of >10 MW. We also received the contents of the map in form of a spreadsheet and the permission to use the data for our research. For generators without power specification we take the value for that generator from the spreadsheet and integrate into our model. For the Matpower base case, we consider 50% of the maximum real power output capacity. The amount

### 3 Methodology

of power generation in the system is of great importance and different values for this parameter are investigated in the experiments.

**Qg (3)** The power grid base model gives no information on the reactive power output of generators. For this reason we can only estimate this parameter and do so for the base case by setting  $Qg$  to 10% of the generator's maximum real power capacity.

**Qmax (4)** This parameter is only used for the optimal power flow analysis and could be set to zero for the base case. However, for the power generator capacities to show up in the case overview this value is set to 10% of the real power capacity  $Pmax$  (9).

**Qmin (5)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**Vg (6)** For the base case, we set the voltage magnitude at the generator to  $1.0 p.u.$ . This results in an electrical system that meets its definition from the power grid base case.

**mBase (7)** As introduced before, this value would be calculated as  $mBase = \sqrt{Pmax^2 + Qmax^2}$ , but since in this work we only perform power flow analyses with static power generation this value is not used and set to zero.

**status (8)** For the base case, we assume every generator to be in-service.

**Pmax (9)** This parameter is only used for the optimal power flow analysis and could be set to zero for the base case. However, for the power generator capacities to show up in the case overview this value is set to the power specification provided by the power grid base model.

**Pmin (10)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**Pc1 (11)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**Pc2 (12)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**Qc1min (13)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

### 3.3 Solve power flow problem

**Qc1max (14)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**Qc2min (15)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**Qc2max (16)** This parameter is only used for the optimal power flow analysis and thus is set to zero for the base case.

**ramp\_agc (17)** Used in MOST and is set to zero here.

**ramp\_10 (18)** Used in MOST and is set to zero here.

**ramp\_30 (19)** Used in MOST and is set to zero here.

**ramp\_q (20)** Used in MOST and is set to zero here.

**apf (21)** Since more complex power generation setups are not the focus of this work, we do not utilize the concept of areas and advanced load distribution. Thus, *apf* is set to zero, meaning that every generator is contributing equally proportionally to their capacity.

### 3.3 Solve power flow problem

This section explains how the power flow problem is solved for Matpower cases created in the previous section. The first step is to fulfill the condition that the power system must be connected. In our case, this is not the case because at least underground lines are not included in the power grid base model. Consequently, these are also missing here, resulting in 16 unconnected islands. In total, there are 597 buses, 1003 branches, 148 generators and 452 loads. Most of these are in island 1, namely 558 buses, 969 branches, 143 generators and 436 loads. At this point, we simplify the model by not considering islands 2-16 and concentrating exclusively on island 1 with the power flow analysis, which should nevertheless provide a sufficiently accurate representation of the Austrian power grid. An exception has already been made for the power plant Simmering, the Pfaffenau waste incineration plant and power plant Arsenal, which are connected to the grid via the substation Simmering (id 771). Further exceptions were made for the Freudenau power plant (substation Kaiserebersdorf, id 770) and the power plant Donaustadt (substation Donaustadt, id 772). The reason for this is that the power plant Simmering, with a nominal output of 1328.7 MW, is the most powerful power plant in Austria and should not be neglected. The same applies to the other power plants, which also provide a significant nominal capacity. Another special feature is that all these power plants are located in the south of Vienna in close geographical proximity to each other. However, as substations 770, 771 and 772 are not connected to the rest of the power grid, we have

### 3 Methodology

Number	Algorithm	Coordinate format	Approach	Matpower abrv.
1	Newton-Raphson	flexible	flexible	NR
2	Newton-Raphson	polar	power mismatch	NR-SP
3	Newton-Raphson	cartesian	power mismatch	NR-SC
4	Newton-Raphson	hybrid	power mismatch	NR-SH
5	Newton-Raphson	polar	current mismatch	NR-IP
6	Newton-Raphson	cartesian	current mismatch	NR-IC
7	Newton-Raphson	polar	current mismatch	NR-IH
8	Fast-Decoupled		XB version	FDXB
9	Fast-Decoupled		BX version	FDBX
10	Gauss-Seidel			GS

Table 3.1: Solving algorithms provided by Matpower

decided to connect the power plants in question to the power grid via substation Wien Suedost (id 178). This substation is well suited for this, as it is both geographically close to the power plants and is the largest substation in Austria. We have already made these changes earlier to the power grid base model and have therefore already taken them into account here.

Now that the power grid is connected, the algorithm for solving the power flow problem can be configured. Note that throughout this work we use the Matpower version 7.1 [Zim20]. In addition to the Newton-Raphson algorithm (NR) in various versions, Matpower also offers the Gauss-Seidel algorithm (GS) and two decoupled load flow-based algorithms (FD). To solve the bases case, we use the Newton-Raphson algorithm, which is also understood as the standard algorithm in the literature. At this point, you can choose between the polar, cartesian and hybrid representation forms for the voltage in the system. You can also choose whether the algorithm uses a power mismatch or a current mismatch approach for the system of equations. An overview is shown in Table 3.1. The advantages and disadvantages of the various combinations are discussed in [SVW19].

For the base case, we use the Newton-Raphson algorithm with polar coordinates and the power mismatch approach. We have already explained the latter in Subsection 2.2.3. We investigate the behavior and performance of other settings for the NR algorithm and the other algorithms in the experimental part of the work.

Another adjustment screw for the algorithm is the accepted error tolerance when solving each equation. As soon as the value determined after an iteration is within the tolerance, the algorithm terminates and continues with the next equation. This value is specified in the per-unit system and we choose it for the base case as  $10^{-7}$ .

Similar to this is the maximum number of iterations allowed for each equation. Here we choose a generous limit of 100. The reason for this is that in the case of the base case, we are interested in the solution found. The time performance is not yet the focus. A large number of allowed iterations is intended to achieve a higher convergence rate and at the same time provides greater independence from the guessed starting values for

voltage angle and magnitudes, as the algorithm is given time to arrive at a solution even with poor starting values. However, when terminating successfully, Matpower specifies the number of iterations required for a solution, which makes it possible to observe the convergence behavior and thus identify parameter configurations for Matpower cases for which the solution algorithm converges particularly quickly or particularly slowly.

It should be noted that the time in 2019 for which the state of the power grid is simulated using the power flow problem is not set here, but beforehand during the construction of the Matpower case. The reason for this is that the model differs in the location and scope of power consumption at different points in time. This is already included in the power grid base model.

## 3.4 Save and process results

Certain information and results are output and saved during and after each simulation run. For this purpose, a separate directory is created for each simulation run, which is named after the time of the simulation in the format *yyyy\_mm\_dd\_hh\_mm\_ss\_simulation*, for example *2024\_08\_09\_13\_43\_53\_simulation*. This allows all simulation runs to be clearly identified and evaluated at a later date. In a simulation run, the power flow problem can either be solved for just one point in time in 2019 or for several points in time, for example for an entire month. Solving the power flow problem for the power grid base model at a specific point of time in 2019 is called a simulation instance. Thus, a simulation run can either consist of a single simulation instance or of multiple simulation instances. There are three different types of output that our codes produces:

### Simulation instance independent information

This category includes information about the underlying power grid base model and its connections to the Matpower domain or the Matpower cases. The csv files *overview\_substations*, *overview\_generators*, and *overview\_lines* provide a list of the given *Substation*, *Generator* and *Line* objects in the base model. This is possible independently of the simulation instance, because only the consumption of the *powerConsumers* changes in this case due to the load profile. The ids, names, voltages and other specifications from the object attributes are then listed in the csv files. This enables a direct and quick view of the model of the power grid on which the simulation is based.

The csv files *sub\_bus\_mapping* and *power\_generators* build the bridge between the Python and Matpower domains. The former file specifies which bus in the Matpower case belongs to which substation in the power grid base model. Remember that several buses can belong to one substation due to the different voltage levels. The *power\_generators* file consists of four columns and works in a similar way: The column *gen\_id* gives the id from the *PowerGenerator* object and *gen\_number* gives the generator number in the Matpower case. In addition, *gen\_bus* is the bus in the Matpower case via which the generator is connected to the power system. Finally, *sub\_id* is the id of the substation to which said bus belongs.

#### Simulation instance specific information

This category consists of information that is different for each simulation instance. This primarily includes the Matpower case files and the results of the solved power flow problems. For this purpose, a separate directory is created for each simulation instance within the top directory of the simulation run. The naming is made up of the time in 2019 for which the power grid is simulated and the algorithm used. More precisely, the directory name has the format *mm\_dd\_hh\_mm\_alg\_alg-number*, e.g. *03\_17\_06\_15\_alg\_5*. The algorithm number *alg-number* is given in the first column of Table 3.1.

This simulation instance-specific directory contains an HTML file and four .M files. The HTML file contains a visualization of the power system, which contains both the power grid base model and the results of the power flow problem. The explanation of the visualization fits better in Section 3.5 and is given there. Three of the .M files are Matpower case files and are again named after the simulated time and the type of case. The file *case\_03\_17\_06\_15\_raw*, for example, contains the Matpower case as constructed in Section 3.2. The file *case\_03\_17\_06\_15\_ran* contains the Matpower case that was ultimately solved using the selected algorithm. In contrast to the raw case, this case can contain modifications, such as the removal of the disconnected islands in the power system. Finally, the case in *case\_03\_17\_06\_15\_solved* contains the solved case. The difference to the ran case is that the unknown quantities, such as voltage magnitudes and angles, have been given values and are no longer initial guessed values. In addition to the case files, there is an .M file that provides an overview of the analysed system and the results of the analysis. The file name contains the analysed time of the simulation instance as well as the prefix *result*, e.g. *results\_03\_17\_06\_15*. How this results file is processed across several simulation instances is explained in the next section.

#### Simulation instance overlapping information

This category includes information that is collected and utilized across multiple simulation instances. Firstly, there is the file for the simulation run statistics that is named after the date of the simulation run, e.g. *2024\_08\_09\_13\_43\_53\_simulation\_statistics*. This file contains a line for each simulation instance within a run, which contains the corresponding metadata. This includes information about the time in 2019 for which the power system was simulated, whether the solution algorithm converged for this instance and which options and which algorithm were used. Finally, an overview of the number of simulation instances performed and the number of successful simulation instances is provided. Secondly, a csv file is constructed that contains the interesting information and characteristic values from the respective results file for each simulation instance and collects them in one place across all simulation instances. The naming again contains the time of the simulation run and the suffix *simulation\_run\_results*, e.g. *2024\_08\_09\_13\_43\_53\_simulation\_run\_results*. A table is constructed in this file, which forms the basis in the analysis for carrying out and examining simulation runs for certain parameter configurations. In addition, certain parts of the table can be extracted from different simulation runs and brought together in order to compare simulation runs

with each other. The table is now structured in such a way that there is one column for each simulation instance. The first row contains the column names where each column is named after the date in 2019 for which the power system is simulated, for example *03\_17\_06\_15*. The second row contains the information as to whether the algorithm for the simulation instance has converged (converged = 1) or not (converged = 0). This information is important for the analysis as well as for what information is available about the simulation instance.

The rows 3-19 provide information about the system itself. Rows 3, 4 and 5 give the number of buses, branches and generators respectively in the system. For the following rows, we provide a brief explanation for the value given in each row, with the row number in brackets after the row name. The *total\_gen\_capacity\_P*(6) gives the total real power capacity across all generators, whereas *on\_line\_capacity\_P*(7) gives the total real power capacity of generators that are in-service. Furthermore, *generation\_P*(8) gives the actual total real power generation in the system. It works in the same way for the reactive power  $Q$ , although there is a lower limit as well as an upper limit for the capacity. Consequently, *total\_min\_gen\_capacity\_Q*(9) gives the lower limit for the reactive power capacity and *total\_max\_gen\_capacity\_Q*(10) gives the upper limit. *on\_line\_min\_capacity\_Q*(11) and *on\_line\_max\_capacity\_Q*(12) give the total reactive power capacity of in-service generators. The actual total reactive power generation is given by *generation\_Q*(13). Next comes the total load  $P$  *load\_P*(14) in the system, which we know as real power demand. This value is made up of the fixed load *fixed\_load\_P*(15) and the dispatchable load *dispatchable\_load\_P*(16). Dispatchable loads are loads that can be switched on as needed depending on the ratio of power generation and demand in the grid. They are not yet used in this work, but we will include them here for future work. Similarly, *load\_Q*(17), *fixed\_load\_Q*(18), and *dispatchable\_load\_Q*(19) give the information for the reactive power demand.

The remaining rows contain information that is specific to each simulation instance. Of interest here are the branch losses during transmission in the system as well as the calculated voltage magnitudes and angles on the buses. This information provides insight into how well the power system would function in reality. First comes the total amount of real power that is lost during transmission *total\_losses\_P*(20). This value is followed by *max\_loss\_P*(21) which is the highest amount of lost real power at a single branch. Now, the respective branch is identified by its *from*-bus *max\_loss\_P\_from\_bus*(22) and its *to*-bus *max\_loss\_P\_to\_bus*(23). High losses are generally undesirable and therefore it is of importance to not only look at the branch with the single highest loss but look at the branch losses from a bigger perspective. For that, we create loss ranges and categorize branches accordingly. That enables us to look at the distribution of branch losses as a whole and identify problematic branches in the grid across different simulation instances. For the real power losses  $loss_P$ , the first range is *loss\_P\_below\_50*(24) that contains every branch with a branch loss below 50 MW, i.e.  $loss_P < 50\text{MW}$ . After that, *loss\_P\_above\_50\_below\_100*(25) counts the branches with  $50\text{ MW} \leq loss_P < 100\text{ MW}$ . Similarly, *loss\_P\_above\_100\_below\_300*(26) counts the branches with  $100\text{ MW} \leq loss_P < 300\text{ MW}$  and *loss\_P\_above\_300*(27) counts the branches with

### 3 Methodology

Row Number	Row Name	Condition
24	loss_P_below_50	$loss_P < 50 \text{ MW}$
25	loss_P_above_50_below_100	$50 \text{ MW} \leq loss_P < 100 \text{ MW}$
26	loss_P_above_100_below_300	$100 \text{ MW} \leq loss_P < 300 \text{ MW}$
27	loss_P_above_300	$loss_P \geq 300 \text{ MW}$

Table 3.2: Real power loss ranges

$loss_P \geq 300 \text{ MW}$ . An overview is shown in Table 3.2.

For reactive power, the total losses are given by  $total\_losses\_Q$ (28) and the maximum loss at one branch is given by  $max\_loss\_Q$ (29). Again, the branch is given by its *from* and *to* buses  $max\_loss\_Q\_from\_bus$ (30) and  $max\_loss\_Q\_to\_bus$ (31). The loss ranges for the reactive power losses  $loss_Q$  are constructed in similar fashion as for the real power losses. Note that we choose to have one more loss range because we observe larger losses at single branches for the reactive power. An overview is given in Table 3.3.

Row Number	Row Name	Condition
32	loss_Q_below_50	$loss_Q < 50 \text{ MVar}$
33	loss_Q_above_50_below_100	$50 \text{ MVar} \leq loss_P < 100 \text{ MVar}$
34	loss_Q_above_100_below_300	$100 \text{ MVar} \leq loss_P < 300 \text{ MVar}$
35	loss_Q_above_300_below_1000	$300 \text{ MVar} \leq loss_P < 1000 \text{ MVar}$
36	loss_Q_above_1000	$loss_P \geq 1000 \text{ MVar}$

Table 3.3: Reactive power loss ranges

Continuing, we examine the bus voltage magnitudes and voltage angles that were determined by solving the power flow problem. Row 37  $min\_vol\_magnitude$ (37) gives the minimum voltage magnitude across all buses and  $min\_vol\_magnitude\_bus\_no$  (38) gives the corresponding bus number. The value for the voltage magnitude is given in the p.u. system with the nominal voltage as base value. Similarly, the maximum occurring voltage magnitude is given by  $max\_vol\_magnitude$ (39) and the bus is provided as  $max\_vol\_magnitude\_bus\_no$  (40). We construct voltage magnitude ranges that we can assign the buses to. The voltage magnitude at a bus is again denoted as  $V_m$ . An overview is given in Table 3.4. Also note that the most desirable voltage magnitude range is  $0.9 < V_m \leq 1.1$  since here the given voltages are closest to the nominal voltage, which would have the value 1.0.

Concluding the table, we save information about the bus voltage angles that were determined by the algorithm. First,  $min\_vol\_angle$ (46) gives the smallest voltage angle  $\theta$  (in degrees) across all buses and  $min\_vol\_angle\_bus\_no$ (47) provides the corresponding bus number to it. Similarly,  $max\_vol\_angle$ (48) provides the maximum voltage angle and  $max\_vol\_angle\_bus\_no$ (49) the bus number. Finally, for the voltage angles we also construct different groups for different angles which are listed in Table 3.5.



Row Number	Row Name	Condition
41	vol_mag_above_1_3	$V_m > 1.3$
42	vol_mag_above_1_1_below_1_3	$1.1 < V_m \leq 1.3$
43	vol_mag_above_0_9_below_1_1	$0.9 < V_m \leq 1.1$
44	vol_mag_above_0_7_below_0_9	$0.7 < V_m \leq 0.9$
45	vol_mag_0_7	$V_m \leq 0.7$

Table 3.4: Voltage magnitude ranges

Row Number	Row Name	Condition
50	vol_ang_above_150	$\theta > 150$
51	vol_ang_above_120_below_150	$120 < \theta \leq 150$
52	vol_ang_above_90_below_120	$90 < \theta \leq 120$
53	vol_ang_above_60_below_90	$60 < \theta \leq 90$
54	vol_ang_above_30_below_60	$30 < \theta \leq 60$
55	vol_ang_above_m_30_below_30	$-30 < \theta \leq 30$
56	vol_ang_above_m_60_below_m_30	$-60 < \theta \leq -30$
57	vol_ang_above_m_90_below_m_60	$-90 < \theta \leq -60$
58	vol_ang_above_m_120_below_m_90	$-120 < \theta \leq -90$
59	vol_ang_above_m_150_below_m_120	$-150 < \theta \leq -120$
60	vol_ang_below_m_150	$\theta \leq -150$

Table 3.5: Voltage angle ranges

### 3.5 Visualization and analysis

For the further evaluation and presentation of the results, we mainly rely on two pillars. The first is the visualization of the power grid, which we enrich with the results of the power flow problem. Secondly, we evaluate the *simulation\_run\_results* files and extract interesting characteristic values which we also present graphically.

#### pyvis-based visualization

The first type of visualization is a representation of the power grid as a graph - consisting of nodes and edges. The edges here are based on *Line*-objects from the power grid base model, while nodes can represent *Substation*-objects as well as *PowerGenerator*- or *PowerConsumer*-objects. With this type of visualization, the focus is not on the geographically correct representation of the elements in the national territory of Austria, but rather on the clear connections between the elements. The tool we use here is the Python library pyvis<sup>5</sup>, which has three main advantages:

<sup>5</sup><https://pyvis.readthedocs.io/en/latest/>

### 3 Methodology

1. Because it is used from Python, the visualization can be created and output in the same call as the rest of our algorithm.
2. The visualization is output as an HTML file and can be opened in any browser without any further steps.
3. The parameters and inputs for creating the visualization can be changed quickly and directly in the code.

These properties enable a smooth operation when using the simulation framework, for example when experimenting with different parameter configurations. It is also easy to highlight important things in the visualization and display unimportant things less prominently or hide them completely. Below we show how the power grid components are displayed, followed by example visualizations for highlighting specific aspects of the simulation results.

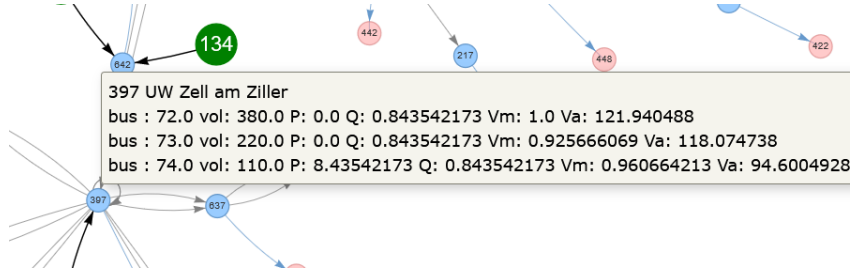


Figure 3.9: Example for information displayed when hovering over a substation node.

In the default visualization, *Substation*-objects are displayed as light blue circles with the corresponding *substation.id* in the middle. When hovering over the circle with the mouse, further information is displayed, as shown in Figure 3.9. The first line of the information contains the id and the name of the substation. A further line is displayed for each voltage level that is present at the substation. Each of these lines first contains the bus number of the bus that was created for this particular voltage level of this substation when the Matpower case was generated. Second, the voltage level is shown under *vol* in the unit kV. The P and Q values after this are to be understood as the real and reactive power demands. These values are already the sum of the demands across all power consumers connected to the substation. Following that, *Vm* gives the resulting voltage magnitude and *Va* the resulting voltage angle at that bus. Here it becomes clear how the power grid base model, the created Matpower case and the solution to the power flow problem come together in the visualization. Also note that the information displayed can be extended as required.

*Line*-objects are displayed in the visualization as light grey directed edges. They are equivalent to branches from the Matpower case, which connects two buses with each other. Note the special case of lines with an *fid* > 1000. There is no *Line*-object for these lines in the original power grid base model, because they are created at the same time as the Matpower case whenever a transformer branch is formed that connects two buses of a

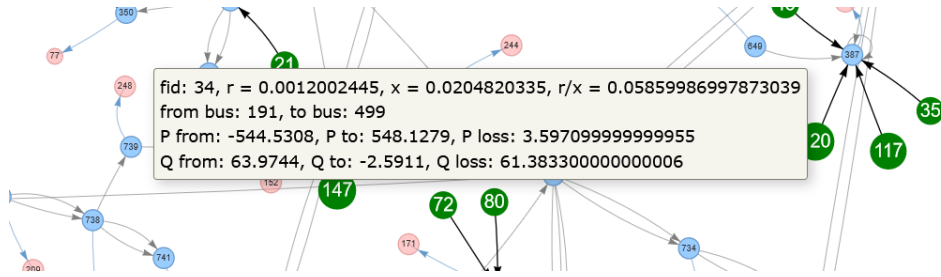


Figure 3.10: Example for information displayed when hovering over a line edge.

substation. In the visualization, these lines are represented as edges where the substation node at the start of the edge is identical to the substation node at the end of the edge. When hovering over an edge, detailed information is displayed, as shown in Figure 3.10. The first line contains the *fid* of the *Line*-object from the power grid base model. This is followed in the same line by the resistance  $r$  and the reactance  $x$  in p.u. The ratio of the two values to each other is then given under  $r/x$  at the end of the line. The second line gives the number of the *from* bus of the branch and the to bus. In the graph, the edge is also directed so that the end of the edge points to the substation to which the to bus belongs. The third line of information shows the real power injections of the buses at the ends of the branch in the first two places.  $P$  from is the injection on the *from* bus and  $P$  to is therefore the injection on the *to* bus. This is followed by the real power branch loss, which is the sum of the two real power branch injections. For example, the loss is zero if as much is taken from a line as is injected into it. The fourth line is constructed in a similar way for the reactive power injections and losses.

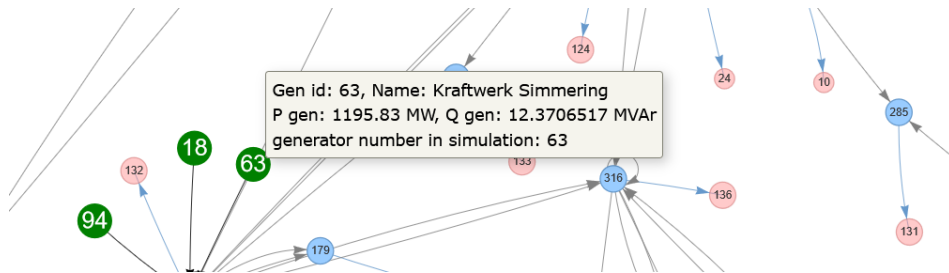


Figure 3.11: Example for information displayed when hovering over a power generator node.

*PowerGenerator*-objects are displayed as dark green nodes in the visualization. They are equivalent to generators in the Matpower case. The node itself contains the *id* given in the *PowerGenerator*-object and a box with further information is displayed when hovering over the node with the mouse, as shown in Figure 3.11. The first line contains the *PowerGenerator.id* again and then the name of the generator. The second line contains the real power generated by the generator under  $P$  gen and the reactive power provided by the generator under  $Q$  gen. These two values originate from the results of the solved

### 3 Methodology

power flow problem and thus form the connection from the power grid base model to the power flow model for the generators. Finally, the generator number, as used in the Matpower context, is specified in the third line. Note that generator nodes are always connected to substation nodes in the visualization. The connecting edges are not to be understood as power lines or branches, but simply symbolize the connection.

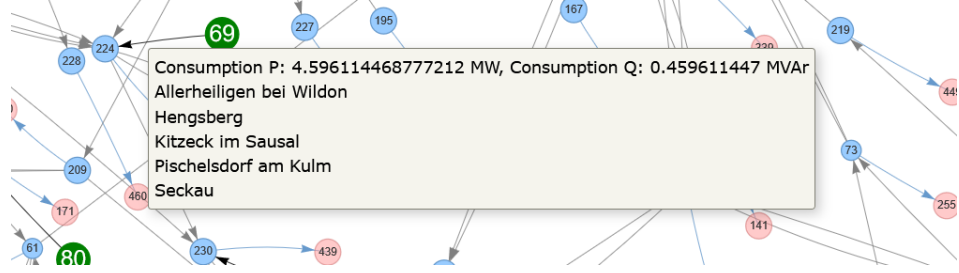


Figure 3.12: Example for information displayed when hovering over a power consumer node.

Power consumers are displayed as bright red nodes in the visualization. In this case, several *PowerConsumer*-objects that are connected to a single substation are combined into a single node for a better overview. This makes sense here because individual power consumers do not play a special role in the Matpower domain, but are combined into demands on a bus there. Also, consumers do not have their own number there, which is why the node here contains an single number that is incremented during the creation of such power consumer nodes. Further information is displayed when hovering over a power consumer node with the mouse. First and foremost this is the real power consumption under *Consumption P* and the reactive power consumption under *Consumption Q*. These values are to be understood as sums of the consumption of all *PowerConsumer*-objects belonging to the substation. All power consumers included in this way are listed in the second and all subsequent lines.

Building on the basic visualization that we built with pyvis, we can now extend and modify it so that certain relationships or characteristics can be easily read from it. Again, the branch losses as well as the bus voltage magnitudes and angles could be particularly interesting for analyzing the power flow results. For this purpose, we construct four variations of the basic visualization, three of which are shown below using the example of a solved power flow problem.

**Focus:**  
**Branch losses  $Q$**

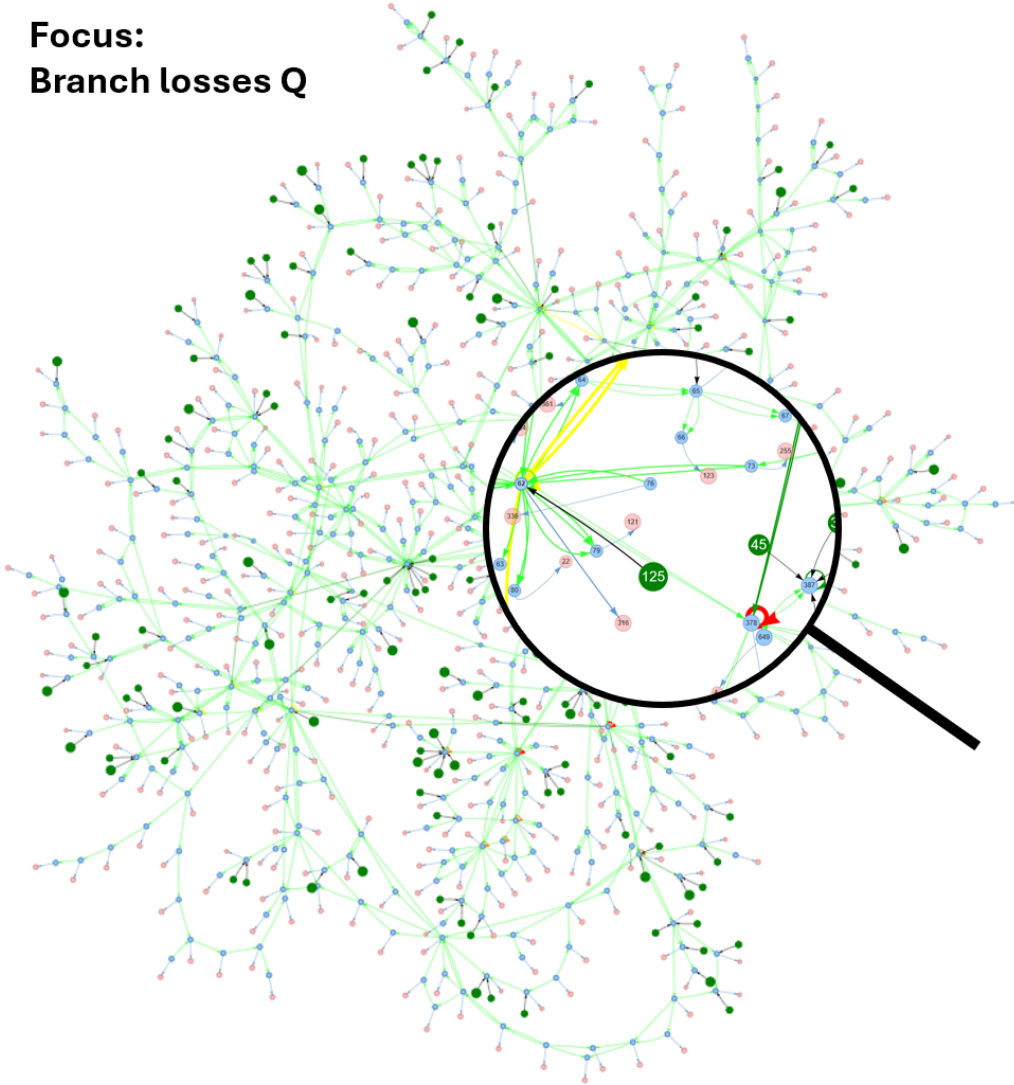





Figure 3.13: Visualization of the power system for the purpose of highlighting reactive power losses.

First, we create a visualization that shows the distribution and size of the reactive power losses in the individual branches. To do this, we use the reactive power loss ranges from Table 3.3 and define a separate edge color and edge width in the visualization for each of the ranges: the greater the reactive power losses in a branch, the thicker the edge is displayed. In addition, each loss range is given its own color, namely

- for  $loss\_Q\_below\_1000$
- for  $loss\_Q\_above\_50\_below\_100$

### 3 Methodology

-  for *loss\_Q\_above\_100\_below\_300*
-  for *loss\_Q\_above\_300\_below\_100*
-  for *loss\_Q\_above\_1000*

This type of visualization is shown for the base case with the simulation date *03\_17\_06\_15* in Figure 3.13. In this case, it can be seen at first glance that the largest reactive power losses occur exclusively on transformer branches. A similar visualization is created for the real power losses which is not shown here.

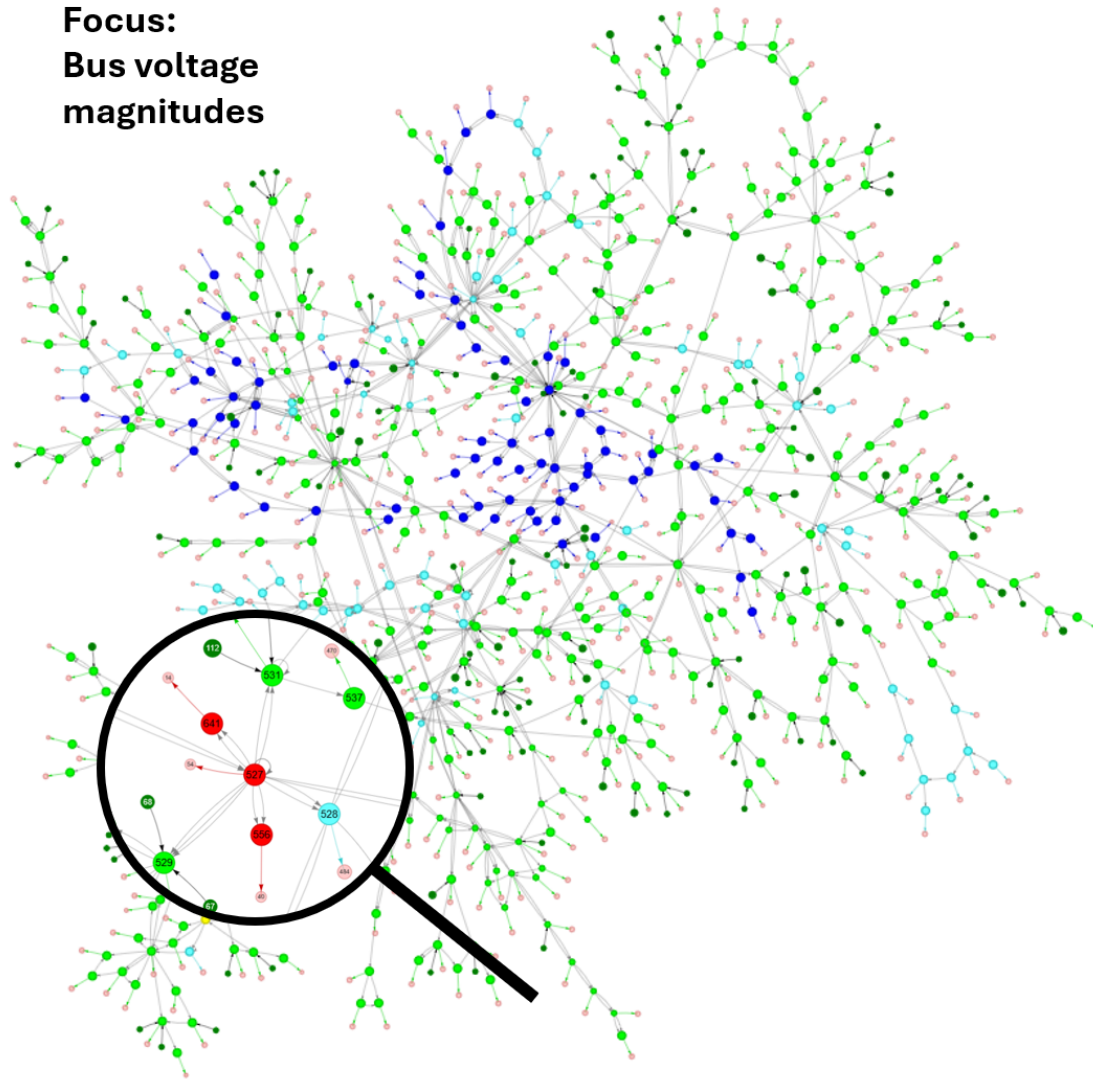





Figure 3.14: Visualization of the power system for the purpose of highlighting the highest bus voltage magnitude deviation at every substation.

Second, we want to highlight the voltage levels in the power system. For that purpose, we modify the base visualization such that the voltage magnitude ranges from Table 3.4 can directly be identified for every bus. Firstly, we increase the size of substation nodes and we also decrease the size of generator nodes since those have less relevance in illustrating the voltage levels in the grid. Secondly, we introduce specific colors for every given voltage magnitude range, namely

- for *vol\_mag\_above\_1\_3*
- for *vol\_mag\_above\_1\_1\_below\_1\_3*

### 3 Methodology

-  for *vol\_mag\_above\_0\_9\_below\_1\_1*
-  for *vol\_mag\_above\_0\_7\_below\_0\_9*
-  for *vol\_mag\_below\_0\_7*

Since substation nodes are given colors but every substation can consist of multiple buses, the bus voltage magnitude with the greatest absolute distance to 1.0 within a substation is deciding for the classification. This type of visualization is shown for the base case with the simulation date *03\_17\_06\_15* in Figure 3.14. In this example, one can directly identify areas in the power grid with low voltage magnitudes (dark blue nodes) and a few substations that consist of buses with a very high voltage magnitude (red nodes).



**Focus:**  
**Bus voltage angles**

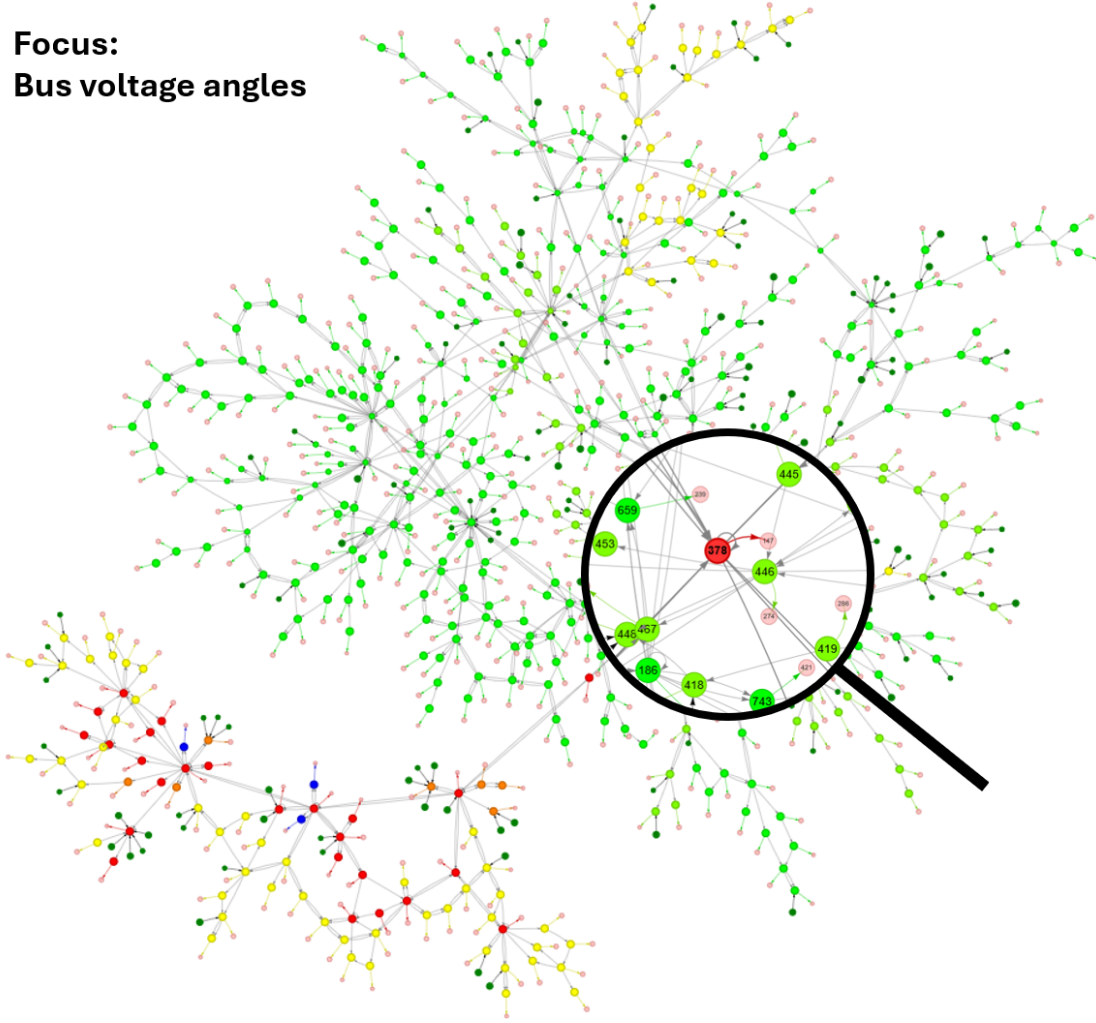













Figure 3.15: Visualization of the power system for the purpose of highlighting the highest bus voltage angle deviation at every substation.

Lastly, we want to highlight the voltage angles at the buses in the power grid. To do this, we proceed in a similar way to highlighting the voltage levels. Substations nodes are made bigger in comparison to the basic visualization. Furthermore, we make use of the voltage angle ranges from Table 3.5 and introduce a color scheme which aims to assign similar colors to substations with similar voltage angles and differing colors otherwise. To determine the voltage angle range of a substation, we identify the bus with the voltage angle furthest from zero degrees out of every bus belonging to that substation. This procedure results in the visualization showing the most extreme bus voltage angles, which is beneficial for analysis since those cases are most often interesting to investigate. This type of visualization is shown for the base case with the simulation date *03\_17\_06\_15* in Figure 3.14. Here you can see that in large parts of the network the voltage angles

### 3 Methodology

match except in western Austria where substation nodes are colored irregularly. The mapping from voltage angle ranges to node colors in the visualization is given as:

-  for *vol\_ang\_above\_150*
-  for *vol\_ang\_above\_120\_below\_150*
-  for *vol\_ang\_above\_90\_below\_120*
-  for *vol\_ang\_above\_60\_below\_90*
-  for *vol\_ang\_above\_30\_below\_60*
-  for *vol\_ang\_above\_m\_30\_below\_30*
-  for *vol\_ang\_above\_m\_60\_below\_m\_30*
-  for *vol\_ang\_above\_m\_90\_below\_m\_60*
-  for *vol\_ang\_above\_m\_120\_below\_m\_90*
-  for *vol\_ang\_above\_m\_150\_below\_m\_120*
-  for *vol\_ang\_below\_m\_150*

#### GeoJSON-based visualization

The second type of visualization is a geographically correct embedding of the core power grid, consisting of power lines and substations, into a map of Austria. For that, our algorithm generates a GeoJSON<sup>6</sup> file that is returned. This file contains the *Substation*- and *Line*-objects from the power grid base model enriched with the results of the simulation and could easily be extended for power generators and consumers. We have opted for the GeoJSON format here because it is widely used and can therefore be handled by most visualization tools, e.g. QGIS<sup>7</sup>. The great advantage of this type of visualization is that data points can also be assessed geographically at first glance. This can be valuable, for example, if there are regional differences or problems in the power grid that may have geographical causes. The disadvantages compared to pyvis are that it takes longer from the completion of a simulation run to the opened visualization. While pyvis directly outputs an HTML file that can be opened with any browser, no finished visualization is output here, but only a file that has to be processed to a visualization manually. A further disadvantage is that when building the power grid map, the substations and power lines are not connected to each other based on their relationship, but both types of elements are entered into the map according to their geolocation. This means that the connections are not clear and unambiguous in some places. For these reasons, we recommend using the pyvis visualization when doing simulation runs and quickly checking in on results.

---

<sup>6</sup><https://geojson.org/>

<sup>7</sup><https://www.qgis.org/>

## 4 Results and discussion

In this chapter we present and discuss the results of the experiments. In Section 4.1 we discuss the results of the simulation run for the month of January 2019. In Section 4.2 we present the results of the simulation run for July 2019 and compare them to those for January. In both cases, we mainly work with *the simulation\_run\_results* file from Section 3.4. The aim of this evaluation is to describe and better understand the system and its modeling for both simulation periods. Firstly, we want to create a good foundation for understanding the system for future researchers. Secondly, we want to identify points that need further investigation and improvement in order to make the model and the results more meaningful. Please also note that the results and conclusions presented are by no means exhaustive, but only a selection from many other analyses that can be carried out with the data generated.

### 4.1 January 2019

Because the load in the grid is the only parameter that differs between two simulation instances (for different points in time) when creating the Matpower cases, we start with an overview of the real power load at different points in time. The mean real power load in January is 9356.35 MW and the median is at 8500.33 MW. The minimum load occurs in the interval 2019/01/01 03:45 at 3285.61 MW, while the maximum loads occurs in the interval 2019/01/15 10:45 at 16643.38 MW.

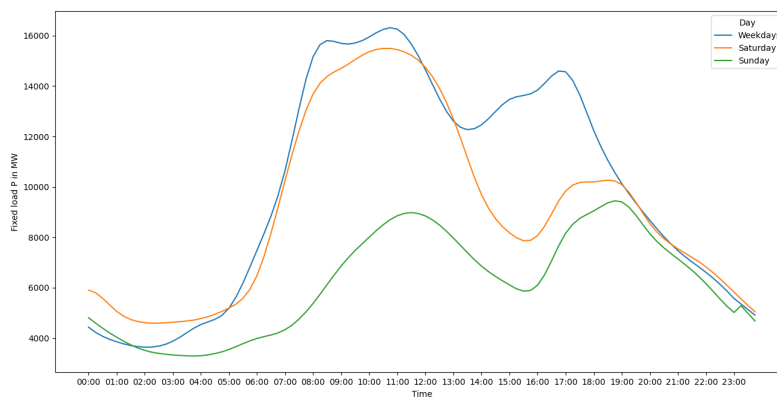


Figure 4.1: Average real power load at different times and days of the week in January

#### 4 Results and discussion

Due to the natural day-night rhythm of people and their working day, electricity consumption depends heavily on the time of day and the day of the week. Figure 4.1 reflects this by showing the average fixed real power load in the grid depending of the time of the day. The underlying values are from the line *fixed\_load\_P(15)* of the *simulation\_run\_results* file for January. The load on the weekdays Monday to Friday is shown here by the blue line. The calculation is based on the average of all relevant days for each simulated 15-minute interval. Similarly, the average for all Saturdays is represented by the orange line and the average for all Sundays by the green line. You can see that for all three groups the load is high during the day and low at night. During the day, there is a peak in the mornings, which is due to the fact that people are most active and consume the most electricity. Consumption drops at midday and in the early afternoon and then peaks again late in the afternoon.

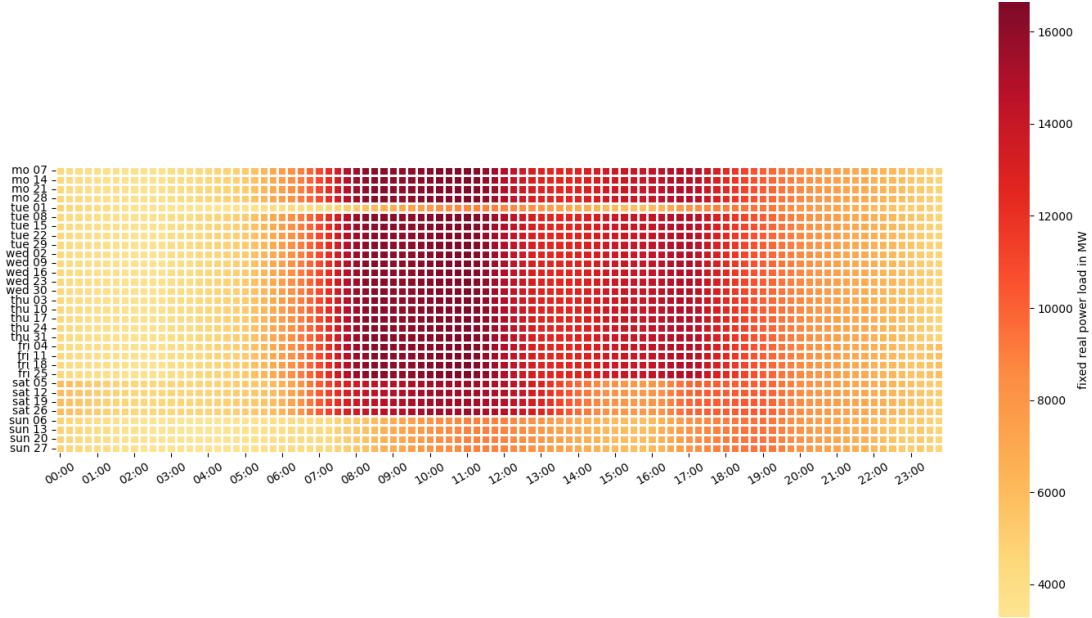


Figure 4.2: The level of real power load for every simulation interval in January.

The heat map in Figure 4.2 shows a similar picture, where the x-axis again shows the time, while each line on the y-axis represents a day in January 2019. The rows are arranged in such a way that the same days of the week are arranged below each other for better comparability. Now there is a box for each interval on each day, which is colored lightly when the load is small and increasingly darker as the load increases. Again, the pattern shows that on weekdays (and Saturdays) there is a lot of load in the grid in the morning and generally little load on Sundays at all times of the day. An exception to this is the first of January, which is an official non-working holiday in Austria, on which consequently little power is consumed.

It is particularly interesting to see for which simulation instances the solution algorithm

finds a solution and for which it does not (row *converged*(2) in *the simulation\_run\_results*). The convergence of the algorithm towards a solution can depend on various factors. For example, the guessed starting values for bus voltages and angles may be too far away from the actual values. The algorithm would then take too long to find a solution. There may also be constraints in the system of equations that cannot be met. For the 2976 15-minute intervals in January, the algorithm finds a solution in 2233 (75%) corresponding simulation instances and in 743 (25%) of them it finds no solution.



Figure 4.3: Overview of which simulation instances a solution can (green) and cannot (red) be found for.

We also visualize the convergence distribution as a heat map with the days on the y-axis and the time on the x-axis (see Figure 4.3). A successful simulation instance is indicated by a green box and an unsuccessful simulation instance by a red box. Comparing Figure 4.2 and Figure 4.3, it is easy to see that the solution algorithm often does not converge for the high-load times in the grid, while a solution can be found for all simulation instances at times with a low load, for example at night. The conclusion that there is a correlation between the load in the network and the convergence of the solution algorithm can be further supported by classifying the simulation instances into *converged* = 1 and *converged* = 0 and considering the load in the grid for both groups. In the group of converged simulation instances, the minimum real power load is 3285.61 MW and the maximum real power load is 15584.15 MW. The average load is 7773.98 MW and the median is 6856.16 MW. In the group of non-converged simulation instances, however, the minimum load of 9754.40 MW and the maximum load of 16643.38 MW are significantly higher. The same is the case for the average load at 14111.96 MW and the median at 14168.01 MW.

An important aspect of energy transmission and distribution is the amount of losses. For real power, these are normally between 8-15% from the power generator to the consumer [Gro23]. The level of losses is part of the solution to the power flow problem, which is why in our setup it is only possible to evaluate them for simulation instances for which the algorithm converges. Nevertheless, the losses are an elementary component of the behavior of the system that we want to investigate. The level of real power generation in our current model is the same for each simulation instance and is 90% of the maximum capacity for

#### 4 Results and discussion

the base case, namely 16761.41 MW. While the average load for the solved simulation instances is 7773.98 MW, the average real power loss is 2149.17 MW. In relation, this would be 12.8% of the power generation and 27.6% of the average load. While 12.8% looks plausible at first glance, it must be remembered that the power generation is always the same - even at times of low load. This distorts the picture, because in reality power generation is adjusted to power consumption. Looking at the minimum losses of 1677.96 MW and the maximum losses of 9229.08 MW, it becomes clear that the real power losses in our model do not only depend on the power generation and that the 12.8% are deceptive.

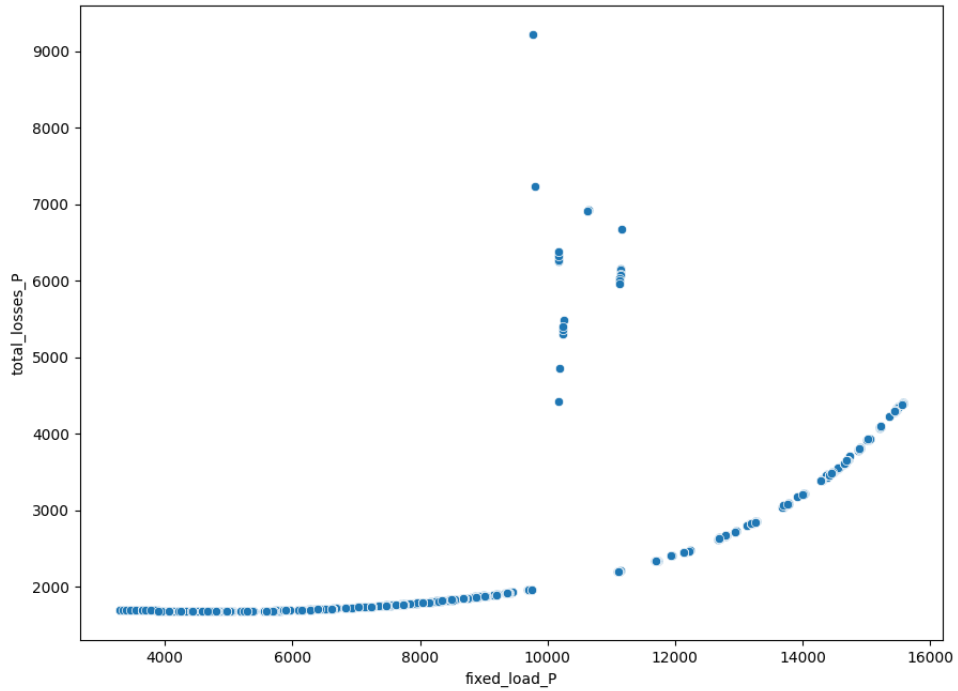


Figure 4.4: Simulation instances arranged according to the given real power load (x-axis) and the total real power losses (y-axis)

If we now look at Figure 4.4, we find that in our model the level of real power losses correlates with the level of load in the grid instead. Each point represents a simulation instance and the points are arranged so that their load is represented by their position on the x-axis and their losses on the y-axis. The trend that can be seen here is that the higher the load, the higher the losses. Interesting here are the 43 outliers in the range around  $fixed\_load\_P = 10000$ , where the highest losses across all simulations instances occur. It is also interesting to note that these outlier simulation instances mainly occur

at time intervals in the 17:45 - 19:15 range (with the exception of one outlier at 13:30). If you now take a look back at Figure 4.3, you can see that there is a very inconsistent convergence behavior for the simulation instances at this time of day. To examine this in more detail, we take a closer look at the period in terms of convergence for all weekdays and Saturdays. We do not consider Sundays (but keep January 1 for orientation) here because there solutions can be found for all simulation instances.

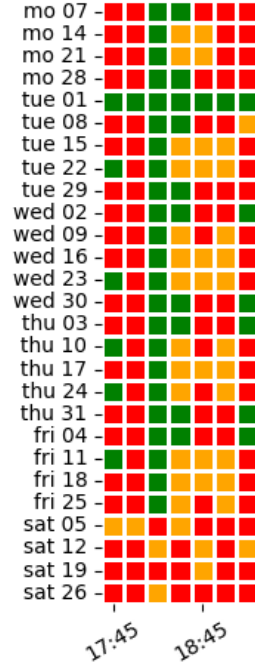


Figure 4.5: Overview of the convergence of simulation instances between 17:45 and 19:45. Red boxes correspond to *convergence* = 0 and green and orange boxes correspond to *convergence* = 1. Also, orange boxes correspond to the outlier simulation instances in Figure 4.4

Figure 4.5 now shows the convergence behavior for the times 17:45 - 19:15, where red again stands for *converged* = 0 and green for *converged* = 1. It becomes particularly interesting if the simulation instances that stand out as outliers in Figure 4.4 with regard to the losses are now colored orange instead of green. You can immediately see that, for example, the boxes for the times 18:00, 18:45 and 19:00 are all colored red or orange (apart from January 1). This means that either no solution can be found for these simulation instances or there are unusually high losses in the grid. Also, for every Saturday this time period seems to be problematic as there are only red and orange boxes. We further investigate the outlier simulation instances by examining the branches where the largest losses occur for each simulation instance. To do this, we evaluate the rows  $\max\_loss\_P(21)$ ,  $\max\_loss\_P\_from\_bus(22)$  and  $\max\_loss\_P\_to\_bus(23)$  from the

#### 4 Results and discussion

*simulation\_run\_results* file. The highest losses on a single branch are 683.21 MW in the simulation instance with the highest losses. This is the 110 kV branch between buses 399 and 428. The corresponding substations are *UW Hainfeld*(id 153) and *UW Neuhaus*(155). However, it is much more interesting to look at the *max\_loss\_P*branch for the other 42 outliers: For instance, the branch between buses 407 and 401 occurs nine times, always for the 18:45 interval. The 19:00 interval is also interesting, as here alone the branch 320-322 occurs four times, 323-307 four times and 320-316 three times. The situation is similar for 18:30, where exceptionally high losses occur in the whole grid and in the branch 164-596 thirteen times.

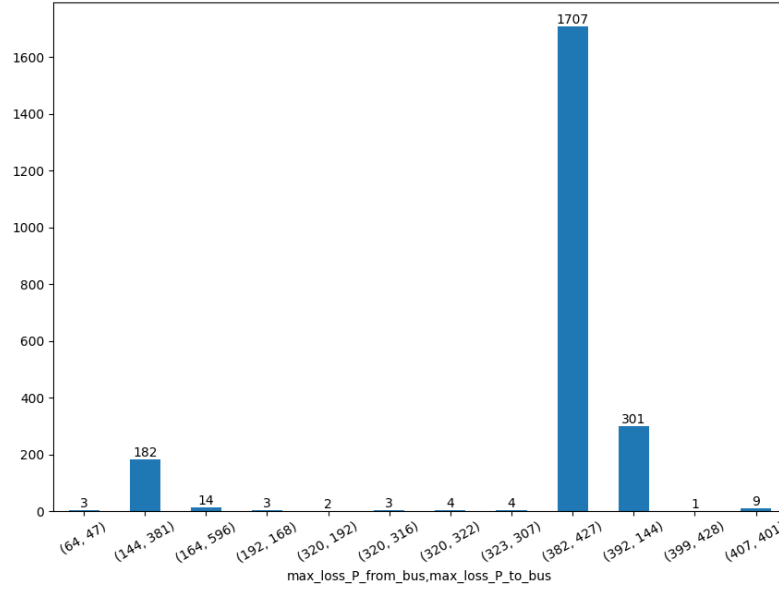


Figure 4.6: Distribution of how often different branches occur in the context of the single largest branch loss (*max\_loss\_P* across all simulation instances).

Figure 4.6 shows that these branches only stand out negatively in the outlier simulation instances. More precisely, it shows how often different branches occur in the context of the single largest branch loss (*max\_loss\_P* across all simulation instances). For the branches discussed, the 43 outlier simulation instances represent their only occurrence in this regard. Otherwise, branches 144-381, 382-427 and 392-144 are mainly represented. However, these are not so interesting because the maximum branch loss is moderate with an average across all simulation instances of 67.39 MW and the total loss for all instances outside the 43 outliers is also unremarkable and follows the correlation with the load in the grid. If we now return to Figure 4.5, it is easy to assume that the branches identified as problematic above could also be partly responsible for the non-convergence of the simulation instances in the period 17:45-19:15.

Figure 4.7 shows a section of the real power loss visualization for the simulation instance



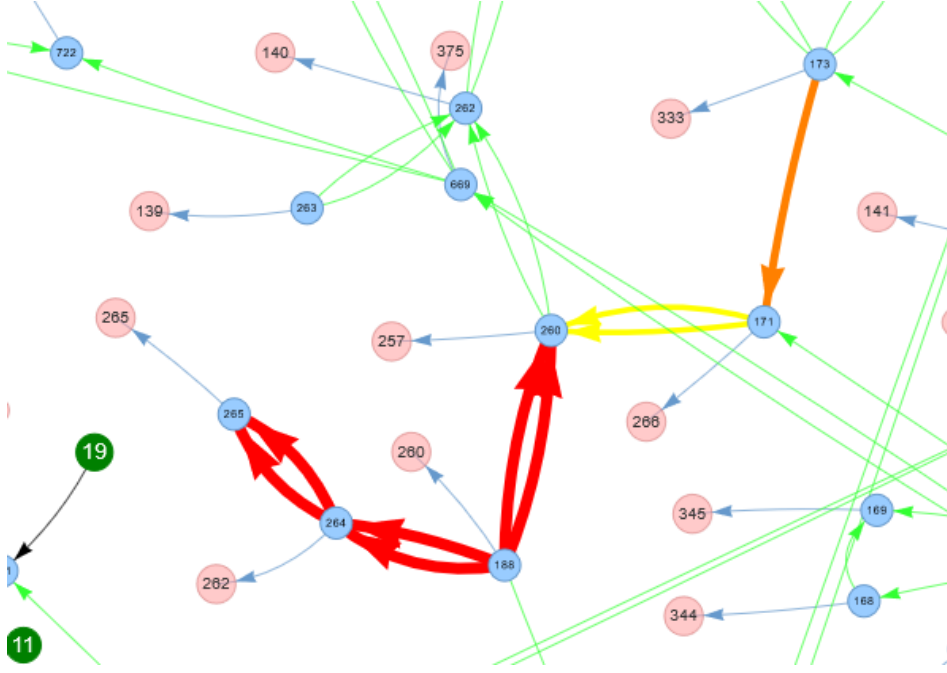


Figure 4.7: Cutout from the real power loss visualization for the 2019/01/09 19:00 simulation instance, showing the high loss branches.

2019/01/09 19:00. For this example, it can be seen that the 110 kV branches identified as problematic between buses 316-320-322-326 have high losses and are all in close proximity to each other. The corresponding substations in the power grid base model are *UW Mattersburg* (id 260), *UW Markt Sankt Martin* (id 188), *UW Oberpullendorf* (id 264) and *UW Deutschkreutz* (id 265). These substations are located in the Austrian province Burgenland. However, no other anomalies can be identified in this area. Due to the limited scope of this work, a deeper search for the causes of this behavior in the model can only be the subject of future work.

Another weakness of our model is the reactive power generation and consumption. We have no values for either of these in our underlying data and therefore have to estimate them. For the base case, the estimates are

$$Q_d = 0.1 \cdot P_d$$

$$Q_g = 0.1 \cdot P_g$$

The fact that the reactive power balance in the system is not yet well adjusted can be seen, for example, in the average reactive power losses, which are significantly too high at 27023.35 MVar. The maximum reactive power loss across all simulation instances is as high as 47145.40 MVar, while even the minimum of 25340.91 MVar is far too high. However, it is much more interesting to compare the branches resulting from *Line*-objects of the power grid base model with the transformer branches that are generated in order

#### 4 Results and discussion

to connect the voltage levels within a substation. While the reactive power losses of the normal branches ( $total\_loss\_Q\_real(37)$ ) are still very high at an average of 7779.28 MVar, the reactive power losses on the transformer branches ( $total\_loss\_Q\_virtual(38)$ ) are exorbitantly high at an average of 19236.91 MVar. This indicates that the parameters for the transformer branches are not yet well adjusted. Finally, we would like to discuss the voltage magnitudes and voltage angles in the network. For the former, we first look at the voltage magnitude range  $vol\_mag\_above\_1\_3$ , which counts the number of buses with a voltage magnitude of more than 1.3 p.u. for each simulation instance. Here it was found that in 2213 of a total of 2233 successful simulation instances, a group of the same five buses filled this voltage magnitude range. These are buses 17, 45, 58, 63 and 69, all of which should have a nominal voltage of 380 kV, but according to the simulation results would end up at 680 kV with 1.79 times the nominal voltage. Looking at the associated substations *Leupolz* (Germany, id 641), *Pradella* (Switzerland, id 556), *Obermooweiler* (Germany, id 585), *UW Buers* (id 544) and *UW Westtirol* (id 527), it shows that these substations are firstly in close proximity to each other and are partially even connected to each other (shown as an example in Figure 4.8, and secondly that they are located in the border triangle of Switzerland, Germany and Austria.

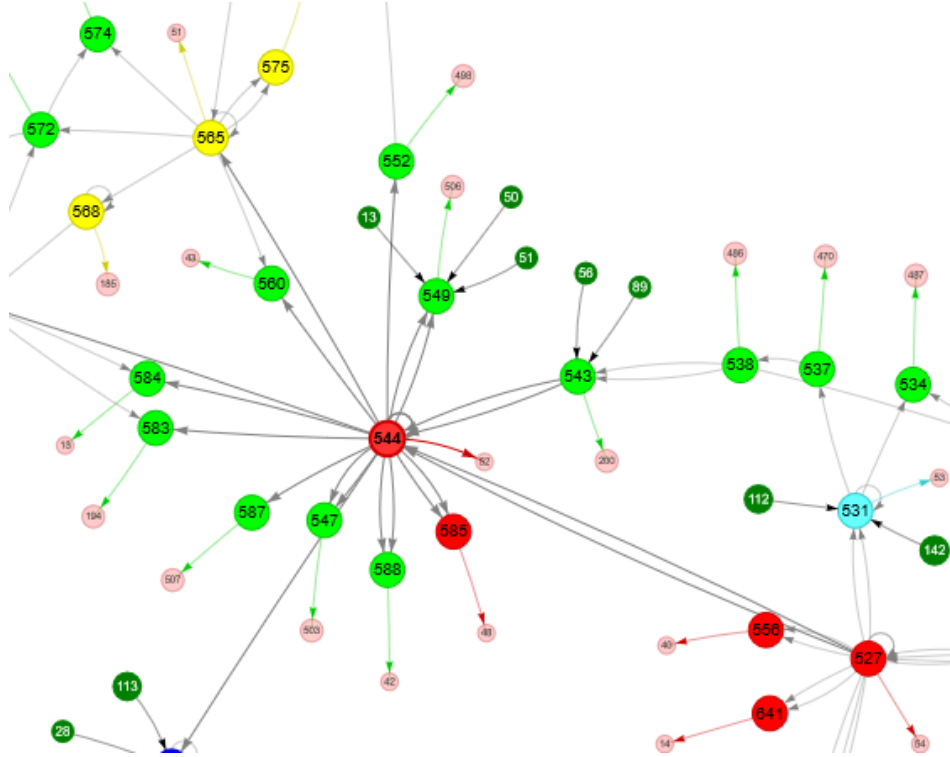


Figure 4.8: Cutout from the voltage magnitude visualization for the 2019/01/31 20:30 simulation instance, showing the high voltage magnitude buses in the border triangle of Switzerland, Germany and Austria.

This behavior could be due to the fact that the power grid base model mainly contains the Austrian grid infrastructure, but in reality the situation is more complex. In Europe, each country not only consumes the power it generates, but some countries produce less power than they consume, while some countries produce more power than they need. The countries' power grids are therefore interconnected to form a European grid. This fact is not yet taken into account in our model and could be the reason for the observation one can make here. A similar pattern shows for the voltage magnitude range *vol\_mag\_above\_1\_1\_below\_1\_3* which for 2130 out of 2233 successful simulation instances is populated by only the 220 kV buses 52, 61, 237. These buses belong to the substations 575, 565 and 568 that are located in the same area and are represented by the yellow nodes in Figure 4.8. The majority of the other buses are found in the voltage magnitude range *vol\_mag\_above\_0\_9\_below\_1\_1* and are therefore assessed by us as normal. We conclude with the voltage angles. There is only one group of outliers here, namely the voltage angle range *vol\_ang\_above\_m\_150\_below\_m\_120* that counts the buses with an voltage angle between -150 and -120 degrees. In 1301 out of 2233 successful simulation instances this range is filled with only the buses 17, 45, 58, 63 and 69. These buses correspond to the voltage magnitude outliers from above and stand out here because they have voltage angles of about -152 degrees according to the results.

## 4.2 July 2019

The analysis of the simulation run for July shows that it is similar in many respects to that for January. The load on the grid is lower in total than in January. Possible explanations for this could be that more energy is consumed in winter for heating and lighting, for example, while there is no demand for this in summer. Another explanation could be that July is during the summer vacations, which could result in the overall energy consumption being lower. The minimum load in July is 3360.75 MW and the maximum is 14244.63 MW. At 8228.14 MW, the average load is around 1000 MW lower than in January, while the median is 7202.68 MW.

Again, Figure 4.9 shows the average load at a given time for weekdays, Saturdays and Sundays. Apart from the slightly different load level, the plot looks very similar for January and July. One difference that stands out is that there is no as large of a peak between 17:00 and 19:00 in July. For weekdays, one possible explanation could be that people come home from work at this time and use electricity for heating and lighting in January, whereas this is not the case in July.

Figure 4.10 again shows the load in the grid in the form of a heat map. While the general pattern is similar, i.e. peak load phases mainly in the morning and afternoon, the difference between January and July is again mainly seen in the early evening.

The oftentimes lower load in the grid is also reflected in the better convergence behavior for the simulation instances. In comparison to January, the convergence rate improved by 10% to overall 85% (2527/2976) for July. Figure 4.11 shows the distribution of convergence across all simulation instances. In combination with Figure 4.10, it can again be seen that convergence and load are related: the lower the load, the more likely it is that a

#### 4 Results and discussion

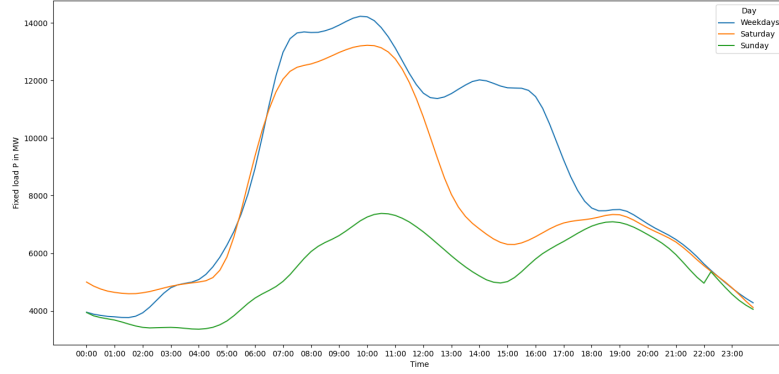


Figure 4.9: Average real power load at different times and days of the week in July

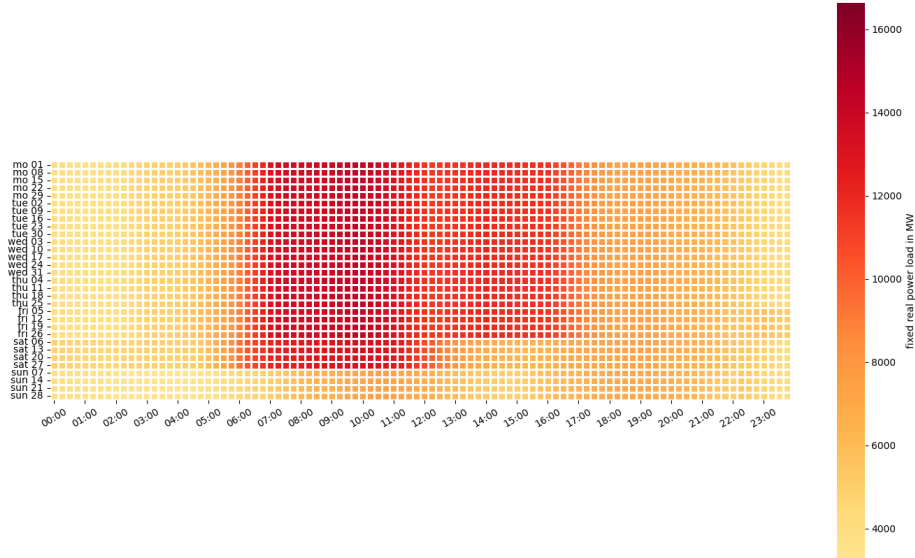


Figure 4.10: The level of real power load for every simulation interval in July.

solution can be found for the simulation instance. That is also indicated by the numbers, as the mean load for converging simulation instances is at 7431.82 MW (with a median of 6739.06 MW) while it is at 12709.89 MW (with a median of 13448.16 MW) for diverging simulation instances. Consequently, an improvement can be clearly seen in the late afternoon and early evening, as the load is lower here than in January. Especially for simulation instances in the 17:45-19:15 range, which is problematic in January, solutions can be found for July.

When it comes to power losses in the grid, once again the simulation runs compare well between January and July. The reactive power losses are very much similar, while the



Figure 4.11: Overview of which simulation instances a solution can (green) and cannot (red) be found for.

real power losses are slightly lower. For instance, the minimum real power loss across all simulation instances is at 1677.21 MW while the maximum is 10149.71 MW. The average loss with 1979.55 MW and the median with 1713.88 MW are slightly lower as well. Once again, relating the real power losses with the load, draws an overall coherent picture with the exception of a few outliers, as depicted in Figure 4.12.

And again the connection can be made that in the outlier simulation instances a few buses and branches in the border triangle of Switzerland, Germany and Austria stand out with large branch losses and low voltage angles. Figure 4.13 shows how often individual branches occur in  $max\_loss\_P(21)$ .

The branch 407-401 is particularly striking here, appearing 31 times in the outliers. As far as the bus voltage magnitudes are concerned, the same buses and substations are under suspicion again. The voltage magnitude range  $vol\_mag\_above\_1\_3$  is filled with buses of substations 527, 544, 556, 585, and 641 in 2491 of 2527 successful simulation instances, while the range  $vol\_mag\_above\_1\_1\_below\_1\_3$  is again populated with buses of substations 565, 568, and 575 in 2491 of 2527 simulation instances. Regarding the voltage angles, the same set of buses stands out as in the January simulation run.

#### 4 Results and discussion

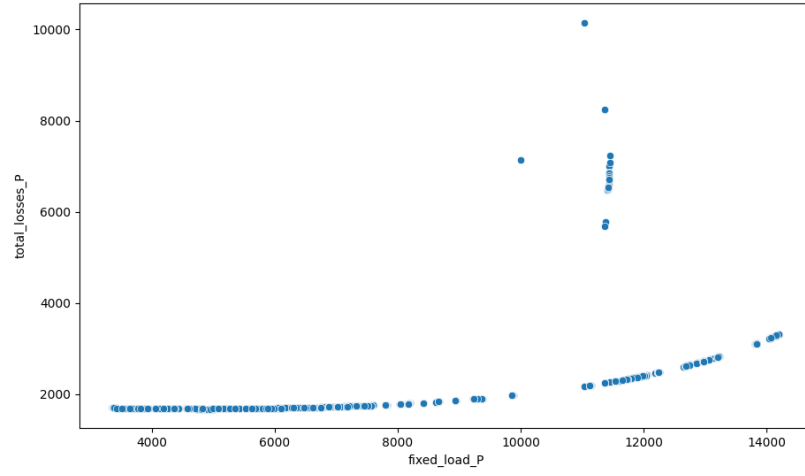


Figure 4.12: Simulation instances arranged according to the given real power load (x-axis) and the total real power losses (y-axis)

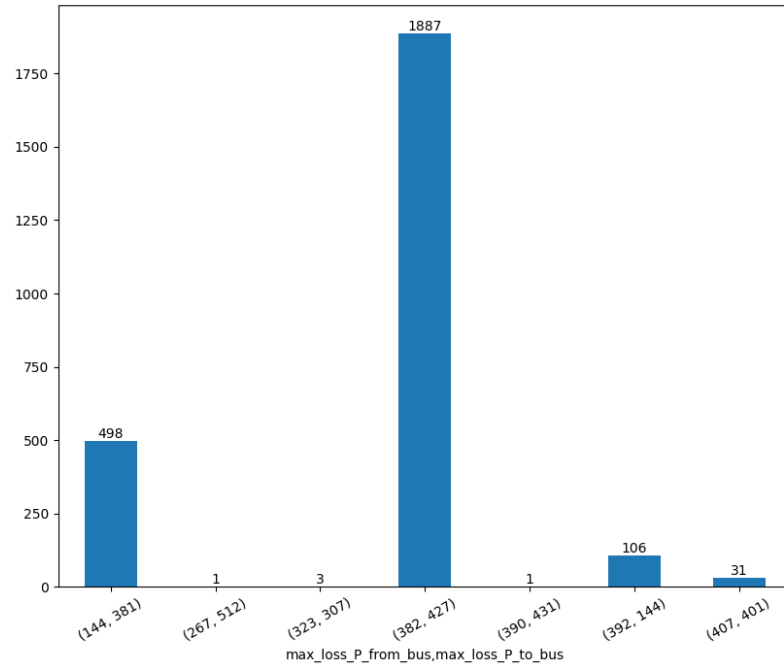


Figure 4.13: Distribution of how often different branches occur in the context of the single largest branch loss ( $max\_loss\_P$  across all simulation instances.)

## 5 Conclusion

In this work, we implemented, experimentally investigated and evaluated a framework, that offers a multitude of functionalities, ranging from constructing Matpower case to running power flow analyses to automatically building visualizations and result analysis. The overall goal was to create the possibility to transfer OSM-based, graphical models of power grids (in this work the Austrian one) into the domain of power flow analysis. This transition extends the scope of the analyses of the present system to include power generation and consumption as well as the general electrotechnical parameters of the power grid. In order to achieve this goal, we have implemented a framework that automates and connects the various steps required on the way from the simple power grid base model to the evaluation of the results of the power flow analysis. The starting point for our work is the OSM-based model of the Austrian power grid, developed by Klauzer et al. [KMAHU24]. We have extended the model in Python so that it can be linked to the data structures from the power flow analysis domain. In the next step, a Matpower case is created based on this. This is the data format used in the open source power flow analysis tool Matpower. The challenge here is to select a suitable parameter configuration, as the information required for a Matpower case goes beyond what is provided by the power grid base model. As initial parameter configuration, we proposed the base case that is later evaluated. Following that, the power flow problem is solved for the case by the means of algorithms provided by Matpower. Here, we proposed an initial configuration for the algorithm parameters. Furthermore, our framework comes with a wide range of utilities for processing and analyzing the results. Firstly, this contains detailed information about the power grid base model and the Matpower cases created. Furthermore, various visualizations are created and output for each simulation instance. These allow a view of the system and the results with a specific focus, for instance, where in the grid high real power losses mainly occur. For the analysis, a results file is created for each simulation run. A large number of characteristic values and statistics are stored in this file for each individual simulation instance. This makes it possible to view the bigger picture. For example, it is possible to observe how the model behaves and changes over the course of a month. We have placed particular emphasis here on the load in the grid, the branch losses and the bus voltage magnitudes and angles. To understand the model even better, we used our framework to create and evaluate a simulation instance for each 15-minute interval for the months of January and July. The results show a mixed picture, which leads us directly to the answer to the first two research questions:

1. How well can the given power grid base model constructed with OSM data be translated into the domain of power flow analysis?
2. What are suitable parameter configurations that make the model reflect reality well?

## 5 Conclusion

The base case, as we constructed it, was a great success in that we were able to achieve good convergence rates for both January (75%) and July (85%). The fact that a solution can be found at all for a simulation instance is important, as only then can you gain a deeper insight into the system. For example, the branch losses as well as the voltage magnitudes and angles provide information that makes further analysis possible. We were also able to observe correlations that suggest that our model reflects reality, at least up to a certain degree. For example, the load on the grid is higher at times when people are active and therefore more electricity is consumed. As the load increases, so do the transmission losses in the grid. At the same time, the convergence behavior of the simulation instances worsens the more load and losses there are in the grid. This is also logical because greater demands can make the power flow equation system more difficult to solve. However, since such large power grids are complex and sensitive systems, we deduce from the good convergence rate and the above-mentioned correlations that the parameter configuration of the base case is an acceptable starting point for further research. However, the results also revealed weaknesses and inconsistencies, which we attribute in part to the limitations of the underlying power grid base model and in part to parameters in the base case that could be improved. The power grid base model has two major limitations: The first is that it mainly depicts the Austrian electricity grid, ignoring the fact that this is part of the European interconnected grid. This means that at least the substations at the contact points with foreign power grids are influenced by these. In our results, this can be seen in particular at the substations *Leupolz*, *Pradella*, *Obermoowiler*, *UW Buers* and *UW Westtirol*. These are all located in the border triangle of Switzerland, Germany and Austria and some are even part of the Swiss or German power grid. In both the January simulation run and the July simulation run, these substations stand out due to excessive losses and voltage magnitudes as well as very low voltage angles. The second limitation that can be observed in our results is that the power grid base model does not contain any information about reactive power generation and reactive power consumption. In addition, unlike power consumption, there is no accurate modeling of real power generation. While in reality this is controlled in such a way that it matches the consumption in the grid, we must assume it to be constant. This is reflected in comparatively high real power losses and extremely high reactive power losses. Future research could focus on extending the model to include dynamic power generation and reactive power loads. Coming back to the base case parameter configuration, we revisit our third research question:

### 3. What parameters need further investigation and improvement?

One point of improvement could be made at the transformer branches. There, the reactive power losses were particularly prominent with extremely high values. Furthermore, we tried out different values for the reactance of the transformer branches in the search for a good parameter configuration for the base case. When doing that, we noticed that the convergence behavior and the losses in the grid reacted extremely sensitively to different reactance values. The behavior was similar with phase shifts in the transformer branches. When we introduced these, the convergence behavior and the losses became worse, which



is why we refrained from doing so for the time being. The reactance also has an influence on the phase shift and we were unable to investigate the relationships in this system further within the scope of this work, but it is certainly interesting for future work. In general, the possibilities of analyzing and finding correlations in the data gathered in the *simulation\_run\_results* file are not yet exhausted. This could be a starting point for further research, as could analyzing and evaluating the remaining months besides January and July. Also, the modeling could be brought even closer to reality by introducing branch flow capacities or dynamic power generation. It would also be interesting to bring the aspect of security and robustness back into focus. For example, one could investigate how the system would behave if one or more substations, power lines or generators were to fail.



# Bibliography

- [Age21] America’s Cyber Defense Agency. Cyber-attack against ukrainian critical infrastructure. Available at <https://www.cisa.gov/news-events/ics-alerts/ir-alert-h-16-056-01> (accessed: 2024/09/26), 2021.
- [(AP] Austrian Power Grid (APG). Apg-netz in zahlen. <https://www.apg.at/stromnetz/stromnetz-oesterreich/> (accessed: 2024/09/26).
- [(AP24] Austrian Power Grid (APG). Netzkarte Österreich. [https://pb1-medien.apg.at/im/dl/pboxx-pixelboxx-18712/APG\\_Netzgrafik\\_Legende\\_d\\_11-2022.pdf](https://pb1-medien.apg.at/im/dl/pboxx-pixelboxx-18712/APG_Netzgrafik_Legende_d_11-2022.pdf), (accessed: 2024/09/26), 2024.
- [Bun07] Bundesnetzagentur. Bericht der bundesnetzagentur für elektrizität, gas, telekommunikation, post und eisenbahnen über die systemstörung im deutschen und europäischen verbundsystem am 4. november 2006. Technical report, Bundesnetzagentur, 2007.
- [Ene23] Oesterreichs Energie. Stromstatistik. <https://oesterreichsenergie.at/stromstatistik-1> (accessed: 2024/09/26), 2023.
- [Ene24] Oesterreichs Energie. Kraftwerkskarte Österreich. <https://oesterreichsenergie.at/kraftwerkskarte> (accessed: 2024/09/26), 2024.
- [Gro23] CHINT Group. How much power loss in transmission lines. <https://chintglobal.com/blog/how-much-power-loss-in-transmission-lines/> (accessed: 2024/09/26), 2023.
- [Hub23] A. Hubbermann. Robustness analysis of the austrian power grid against intentional and non-intentional attacks. Master’s thesis, Vienna University of Economics and Business, 2023.
- [IHK<sup>+</sup>16] Pierre Ibisch, Monika Hoffmann, Stefan Kreft, Guy Pe’er, Vassiliki Kati, Lisa Biber-Freudenberger, Dominick Dellasala, Mariana Vale, Peter Hobson, and Nuria Selva. A global map of roadless areas and their conservation status. *Science*, 354:1423–1427, 12 2016.
- [JFMP16] Cédric Jozs, Stéphane Fliscounakis, Jean Maeght, and Patrick Panciatici. Ac power flow data in matpower and qcqp format: itesla, rte snapshots, and pegase, 2016.
- [JMe08] United States Energy Department; JMesserly. Electricity grid simple - north america, 2008.

## Bibliography

- [KMAHU24] Anja Klauzer, Markus Maier, Lore Abart-Heriszt, and Johanna Ullrich. Fostering security research in the energy sector: A validation of open source intelligence for power grid model data. *Computers Security*, 146:104042, 2024.
- [MAT15a] R. MATPOWER 5.0, Korab. Case2383wp power flow data for polish system - winter 1999-2000 peak. <https://matpower.org/docs/ref/matpower5.0/case2383wp.html> (accessed: 2024/09/26), 2015.
- [MAT15b] R. MATPOWER 5.0, Korab. Case2736sp power flow data for polish system - summer 2004 peak. <https://matpower.org/docs/ref/matpower5.0/case2736sp.html> (accessed: 2024/09/26), 2015.
- [MAT15c] R. MATPOWER 5.0, Korab. Case2737sop power flow data for polish system - summer 2004 off-peak. <https://matpower.org/docs/ref/matpower5.0/case2737sop.html> (accessed: 2024/09/26), 2015.
- [MAT15d] R. MATPOWER 5.0, Korab. Case2746wop power flow data for polish system - winter 2003-04 off-peak. <https://matpower.org/docs/ref/matpower5.0/case2746wop.html> (accessed: 2024/09/26), 2015.
- [MAT15e] R. MATPOWER 5.0, Korab. Case2746wp power flow data for polish system - winter 2003-04 evening peak. <https://matpower.org/docs/ref/matpower5.0/case2746wp.html> (accessed: 2024/09/26), 2015.
- [MAT15f] R. MATPOWER 5.0, Korab. Case3012wp power flow data for polish system - winter 2007-08 evening peak. <https://matpower.org/docs/ref/matpower5.0/case3012wp.html> (accessed: 2024/09/26), 2015.
- [MAT15g] R. MATPOWER 5.0, Korab. Case3120sp power flow data for polish system - summer 2008 morning peak. <https://matpower.org/docs/ref/matpower5.0/case3120sp.html> (accessed: 2024/09/26), 2015.
- [MAT15h] R. MATPOWER 5.0, Korab. Case3375wp power flow data for polish system - winter 2007-08 evening peak. <https://matpower.org/docs/ref/matpower5.0/case3375wp.html> (accessed: 2024/09/26), 2015.
- [Med14] Matke C. Kleinhans D. Medjroubi, W. Open source reference model of european transmission networks for scientific analysis (v0.2). <https://www.power.scigrid.de/> (accessed: 2024/09/26), 2014.
- [MWK<sup>+</sup>18] Ulf Philipp Mueller, Lukas Wienholt, David Kleinhans, Ilka Cusmann, Wolf-Dieter Bunke, Guido Pleßmann, and Jochen Wendiggensen. The ego grid model: An open source approach towards a model of german high and extra-high voltage power grids. *Journal of Physics: Conference Series*, 977(1):012003, feb 2018.

- [Ngu23] L. Nguyen. Understanding the energy crisis in south africa. <https://earth.org/energy-crisis-south-africa/> (accessed: 2024/09/26), 2023.
- [ONL21] ZEIT ONLINE. Stromausfall in dresden vermutlich durch ballon verursacht. Available at <https://www.zeit.de/gesellschaft/2021-09/stromausfall-dresden-ballon-ermittlungen-polizei> (accessed: 2024/09/26), 2021.
- [RDZ20] C. E. Murillo-Sanchez R. D. Zimmermann. *MATPOWER User's Manual, Version 7.1*, 2020.
- [SA74] B. Stott and O. Alsac. Fast decoupled load flow. *IEEE Transactions on Power Apparatus and Systems*, PAS-93(3):859–869, 1974.
- [Sch17] Nebel A. Scharf, M. Open source german transmission grid model based on openstreetmap. <https://github.com/wupperinst/osmTGmod> (accessed: 2024/09/26), 2017.
- [Sta23] Statista. *Net electricity consumption worldwide in select years from 1980 to 2022*, 2023.
- [Sto74] B. Stott. Review of load-flow calculation methods. *Proceedings of the IEEE*, 62(7):916–929, 1974.
- [SVW19] Baljinnyam Sereeter, Cornelis Vuik, and Cees Witteveen. On a comparison of newton–raphson solvers for power flow problems. *Journal of Computational and Applied Mathematics*, 360:157–169, 2019.
- [Tag24a] Tagesschau. Bundesanwaltschaft ermittelt zu tesla-anschlag. Available at <https://www.tagesschau.de/inland/tesla-anschlag-ermittlungen-100.html> (accessed: 2024/09/26), 2024.
- [Tag24b] Tagesschau. Großangriff auf ukrainische energieinfrastruktur. Available at <https://www.tagesschau.de/ausland/europa/ukraine-energieversorgung-angriffe-100.html> (accessed: 2024/09/26), 2024.
- [Tag24c] Tagesschau. Warum der schaden für tesla so hoch ist. Available at <https://www.tagesschau.de/wirtschaft/unternehmen/tesla-gruenheide-konflikte-100.html> (accessed: 2024/09/26), 2024.
- [Ten02] Jen-Hao Teng. A modified gauss–seidel algorithm of three-phase power flow analysis in distribution networks. *International Journal of Electrical Power Energy Systems*, 24(2):97–102, 2002.
- [TH67] William. F. Tinney and Clifford E. Hart. Power flow solution by newton’s method. *IEEE Transactions on Power Apparatus and Systems*, PAS-86(11):1449–1460, 1967.

## Bibliography

- [UCT07] UCTE. Final report system disturbance on 4 november 2006. Technical report, UCTE, 2007.
- [Wie16] B. Wiegman. Gridkit is an power grid extraction toolkit. <https://github.com/bdw/GridKit> (accessed: 2024/09/26), 2016.
- [Zim20] Murillo-Sánchez Carlos E. Zimmerman, Ray D. Matpower, oct 2020.
- [ZMST11] Ray Daniel Zimmerman, Carlos Edmundo Murillo-Sánchez, and Robert John Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.