Energy Transition: Modelling and Simulating Large Scale Multi-Carrier Energy Networks

Literature Review

by

Lucy Westerweel

Supervisors:
Prof.dr.ir. C. Vuik
B. Nguyen

December 19, 2025

TU Delft Delft University of Technology

Contents

List of Abbreviations

AC alternating current.
AS additive Schwarz.

Bi-CG bi-conjugate gradient.
Bi-CGSTAB bi-conjugate gradient stabilized.
BIMs basic iterative methods.

CG conjugate gradient.
CHP combined heat and power.
CPU central processing unit.

DD domain decomposition.
DHS disctrict heating system.

FDLF fast decoupled load flow.

GMRES generalized minimum residual.
GPU graphics processing unit.

ILU incomplete LU.

KCL Kirchhoff's current law.
KVL Kirchhoff's voltage law.

MCN multi-carrier energy network.
MES multi-energy system.
MSS master-slave splitting.

NR Newton-Raphson.

RAS restricted additive Schwarz.
RMS root mean squared.

SAP Schwarz alternating procedure.
SC single-carrier.
SPD symmetric positive definite.

VAR volt-ampere reactive.

Chapter 1

Introduction

Climate change is one of the biggest challenges of the twenty-first century. Since the industrial revolution, human actions have increased the atmospheric concentration of greenhouse gas emissions. The shifts in recent global climate patterns primarily driven by human activities are referred to as *contemporary climate change*. It involves shifts in long-term averages and variation around these averages, with extreme weather events occurring more frequently. By 2017, the average temperature had increased with 1.0 °C compared to pre-industrial levels [1]. To combat climate change, the Paris Agreement was signed and adopted by 195 parties at the UN Climate Change Conference (COP21) in Paris in 2015 [2]. Its overarching goal is to

> *'Hold the increase in the global average temperature to well below* 2.0 °C *above pre-industrial levels and to pursue efforts to limit the temperature increase to* 1.5 °C *above pre-industrial levels'.*
>
> —Paris Agreement, 2015

Additionally, the Dutch Climate Act states that The Netherlands must be climate neutral in 2050 with an energy system with greenhouse gas emissions that are 95% lower than 1990 [3].

1.1 Motivation

To achieve the climate goals of the Paris Agreement [2] and Dutch Climate Act [3], the emission of greenhouse gases has to be reduced. This requires fossil fuel burning to be significantly reduced, while rapidly growing the share of sustainable energy. This type of energy largely consists of *renewable energy*, which is derived from natural sources that are replenished on a human timescale, such as wind or solar power. It is projected that ultimately renewable energy will account for the vast majority of electricity generation around 2050 [4]. These renewable energy sources are often inherently intermittent and variable, meaning that they are not constantly available. Incorporation of renewables is therefore not straightforward. Energy networks that include more than one carrier of energy, such as electricity, heat or hydrogen gas, are referred to as multi-carrier energy networks (MCNs). In order to properly incorporate renewable energy in MCNs, it is important to have reliable and fast simulations that can show the effects of coupling various energy carriers [4, 5].

Modelling of energy networks is done by determining the flows of energy within the network, also referred to as 'load-flow analysis'. Mathematically, the load-flow analysis leads to the network's load-flow equations, whose solution determines the steady-state flow of energy within the network. These equations are often non-linear and lead to large-scale sparse systems. In recent studies, it has been shown that a steady-state load-flow analysis can be performed for MCNs [6, 7, 8, 9, 10, 11]. However, the methods employed showed convergence problems in terms of accuracy and/or speed for large scale MCNs. Therefore, scalability and solvability are important when investigating and modelling these large scale networks.

1.2 Research Context and Scope

This literature review aims to get an insight into recent developments related to modelling and numerical simulation of large scale multi-carrier energy networks to identify open problems and research gaps, focussing on the numerical solution methods used in the load-flow analysis. This is done by including and comparing studies on the modelling of MCNs as well as single-carrier (SC) networks in terms of solution methods and high-lighting computational challenges.

Dynamic simulations, market-based models, control strategies, and optimization problems are considered outside the scope for this review, and should be considered as applications of the models mentioned in this report. Additionally, the formulation and models used to represent energy networks are mentioned and discussed in this report, but are not part of the scope of the research.

1.3 Report Overview

The rest of this literature review is structured as follows:

- Chapter 2: 'Single-Carrier Energy Networks', provides an introduction to the relevant topics related to modelling electricity and gas networks. It first provides fundamental concepts needed to understand electricity and gas networks, after which the corresponding systems of equations are derived.
- Chapter 3: 'Multi-Carrier Energy Networks' describes how electricity and gas networks are coupled into one MCN. The chosen method of coupling is described and justified. The chapter concludes with the system of equations for a general MCN.
- Chapter 4: 'Solvers and Preconditioners' contains an overview of solution methods used in modelling and simulation energy networks. Various iterative solution methods and preconditioning techniques are discussed, concluding with domain decomposition methods.
- Chapter 5: 'State of the Art' provides an overview of research and studies on solving large scale electricity, gas and MCNs, focussing on the method of solving the corresponding system of equations.
- Chapter 6: 'Conclusion' provides the implications of the literature review and contains the research questions for the next phase of the thesis.

Chapter 2

Single-Carrier Energy Networks

The modelling of large scale multi-carrier energy networks is based on the coupling and interaction between different energy carriers, such as electricity, gas and heat. These carriers are based on different physical concepts and operate on different time-scales. In order to interpret and work with the models and equations of multi-carrier energy networks, it is important to have a good understanding of the energy networks of each carrier individually. The aim of this chapter is to get a basic understanding of the physical phenomena in each single-carrier network, with a focus on the mathematical derivation of the load-flow equations. The contents and aim of this chapter as a whole are based on the PhD thesis of Markensteijn [6].

This chapter gives the concepts and models used in the modelling of electricity and gas networks. Electricity networks are discussed in Section 2.1, and gas networks in Section 2.2. The modelling of gas and electricity energy networks in the context of graphs is explained in Section 2.3. Finally, Section 2.4 derives the steady-state load flow equations for both single-carrier energy networks.

2.1 Electricity Networks

Electricity plays a central role in many modern technologies as a carrier of electric energy. It is based on *electric charge*, a physical property of matter that can result in attraction or repulsion in the presence of other matter with charge. The movement of this electric charge is known as *electric current*. The charge can flow along a closed path, referred to as a *circuit*, and is used to power components, such as lamps, motors and other electronic devices. In practice, electricity is generated at a centralized source and transported over power lines to the end users.

This section provides an overview of electricity as an energy carrier and introduces the reader to some fundamental concepts. Additionally, the modelling of power transmission lines is presented. The contents of this section are based on [12, 13, 14, 15].

2.1.1 Electricity Network Fundamentals

Fundamentals of electricity network modelling include the concepts of voltage, current and power, as well as important physical laws and properties of circuit elements.

Voltage and Current in AC Circuits

A circuit with alternating current (AC) is one where the current alternates between positive and negative over time. The currents and voltages in an AC power system usually follow a time-dependent sinusoidal function. In a steady-state system, the behaviour of the system no longer varies over time. In an AC circuit this implies that the current and voltage still follow a time-dependent sinusoid, but with a fixed

frequency ω. The steady-state voltage is then as follows

$$v(t) = V_m \cos(\omega t + \delta_V),$$

with V_m the voltage amplitude in V, ω the angular frequency in Hz and δ_V the phase of the voltage. Similarly, for the current we have

$$i(t) = I_m \cos(\omega t + \delta_I),$$

with I_m the current amplitude in amps, ω the angular frequency in Hz and δ_I the phase of the current. The fixed angular frequency ω is usually 50 or 60 Hz for domestic use, depending on the country or region.

In AC power system modelling it is customary to work with the *root mean squared (RMS) value*, denoted with $|\cdot|$, which is the value of the equivalent direct voltage (or current) that dissipates the same amount of power in a given resistor during one time period of the alternating voltage (or current). They are computed as

$$|V| = \sqrt{\frac{1}{T}\int_0^T v(t)^2 \, dt} = \frac{1}{\sqrt{2}} V_m,$$
$$|I| = \sqrt{\frac{1}{T}\int_0^T i(t)^2 \, dt} = \frac{1}{\sqrt{2}} I_m.$$

Without loss of generality, we can write the sinusoidal expressions for voltage and current as

$$v(t) = \sqrt{2}|V| \cos(\omega t), \tag{2.1}$$
$$i(t) = \sqrt{2}|I| \cos(\omega t - \phi). \tag{2.2}$$

Here, *phase shift angle* ϕ represents the difference in phase of the voltage and current sinusoids.

Phasor Notation

The angular frequency ω is constant in a steady-state system. Therefore, we are only interested in the amplitude and phase shift of the voltages and currents, which is why *phasor notation* is widely used. A phasor is a complex number that represents the amplitude and phase of a sinusoid. The use of phasor notation allows to switch from the time domain to the phasor domain, so that time-dependency is dropped, simplifying steady-state calculations and analysis. In the phasor domain the time-dependency is not denoted, as the frequency is constant in a steady-state system. A phasor can be seen as an equivalent of a sinusoid or complex number. Using Euler's identity, we see the equivalence relation between a sinusoid, complex number and phasor as follows

$$v(t) = \sqrt{2}|V| \cos(\omega t + \delta_V) = \sqrt{2}\mathrm{Re}\left(|V| e^{\boldsymbol{i}(\omega t + \delta_V)}\right) = \sqrt{2}\mathrm{Re}\left(V e^{\boldsymbol{i}\omega t}\right),$$

with $V = |V| e^{\boldsymbol{i}\delta_V}$ the voltage phasor. In phasor notation we denote the amplitude and phase angle,

$$V = |V| e^{\boldsymbol{i}\delta_V} = |V| \angle \delta_V.$$

Resistance and Ohm's Law

Various materials are known to resist the flow of electric charge, where the amount of resistance to the flow is a material property. This *resistance* is measured in ohms (Ω). *Ohm's law* states that the voltage v across an element (resistor) is directly proportional to the current i flowing through. This proportionality is defined as the resistance R of an element, which in computed in the phasor domain as

$$R = \frac{V}{I}. \tag{2.3}$$

Kirchhoff's Voltage and Current Laws in AC Systems

Besides Ohm's law, there are two other important physical laws often used in circuit analysis. There are *Kirchhoff's current law (KCL)* and *Kirchhoff's voltage law (KVL)*.

KCL is based on conservation of charge and states that the sum of current phasors entering and leaving a node is always zero,

$$\sum_{n=1}^{N} I_n = 0. \tag{2.4}$$

This implies that the magnitude of currents entering the node equals the magnitude of currents leaving the node.

KVL is based on conservation of energy and states that the sum of all voltage phasors around a closed path is zero,

$$\sum_{m=1}^{M} V_m = 0.$$

This implies that the sum of all voltage drops within a closed loop equals the sum of all voltage rises.

Power

The concept of power is defined as the time rate of the expending and absorbing energy, and is measured in watts ($W = J/s$). The *instantaneous power* p, is the power absorbed or supplied by an element in a circuit at a specific time. It is computed as the product of the voltage across the element and the current through it,

$$p(t) = v(t) \cdot i(t). \tag{2.5}$$

Substitution of voltage and current definitions (2.1) and (2.2) into instantaneous power (2.5) results in

$$\begin{aligned} p(t) &= 2|V||I| \cos(\omega t) \cos(\omega t - \phi), \\ &= |V||I| \cos(\phi) (1 + \cos(2\omega t)) + |V||I| \sin(\phi) \sin(2\omega t). \end{aligned} \tag{2.6}$$

In an AC circuit, the voltage and current are constantly changing over time, and thus so is the power. Similarly to how RMS values of voltage and current are used in AC circuits, the *real power* P is defined as the average of the instantaneous power during one time period,

$$P = \frac{1}{T} \int_0^T p(t)\ dt = |V||I| \cos(\phi). \tag{2.7}$$

Note that the real power corresponds to the time-independent part of the first term in (2.6). The real power can be interpreted as the useful power that is actually dissipated by the load, and is measured in watts.

Besides the real power, in AC systems there is also a *reactive power*, denoted by Q. This is the power that oscillates back and forth between the source and the load, and does not do work. It is however a necessary part for the functioning of AC systems. The reactive power is measured in volt-ampere reactive (VAR) and corresponds to the time-independent part of the second term in (2.6), which has average zero. We thus have

$$Q = |V||I| \sin(\phi). \tag{2.8}$$

Both the real and reactive power depend on phase shift angle ϕ, which indicates how in phase the voltage and current are. If $\phi > 0$, the current is lagging behind the voltage, and when $\phi < 0$, the current is leading the voltage. The *power factor* is defined as

$$\text{pf} = \cos(\phi) = \cos(\delta_V - \delta_I),$$

the cosine of the phase shift angle.

The phase shift angle ϕ influences the magnitude of the real and reactive powers. If ϕ is zero, the power factor is 1 and the voltage and current sinusoids are in phase, resulting in a large true power P, and zero reactive power Q. This can be interpreted as a perfect system where all electricity from the source is used to do work. On the other hand, when ϕ is near $\pm\frac{1}{2}\pi$ and the power factor is near 0 the true power P becomes zero, and reactive power Q becomes large. This implies that no electricity is used to do work and is instead stored in magnetic or electric fields.

In order to simplify power relations and expressions, the *complex power* is used. It combines the real and reactive power into a single complex number, defined as

$$S = VI^*, \tag{2.9}$$

the product of the voltage phasor V and complex conjugate of current phasor I. The complex power can equivalently be written in polar coordinates, phasor notation and Cartesian coordinates as follows

$$\begin{aligned} S &= VI^*, \\ &= |V||I|\left(\cos\left(\phi\right) + \boldsymbol{i}\sin\left(\phi\right)\right), \\ &= |V||I|e^{\boldsymbol{i}\phi}, && \text{Polar} \\ &= |V||I|\angle\phi = |S|\angle\phi, && \text{Phasor} \\ &= P + \boldsymbol{i}Q. && \text{Cartesian} \end{aligned}$$

In the phasor notation, the complex power has magnitude $|S| = |V||I| = \sqrt{P^2 + Q^2}$, defined as the *apparent power*. This is a practical quantity often used as an electrical apparatus specification.

Impedance and Admittance

Besides resistors, there are also other passive circuit elements. The most relevant in power system analysis are the storage elements *capacitors* and *inductors*. A resistor dissipates energy, whereas capacitors and inductors store energy.

A capacitor stores energy in an electric field. The amount of energy stored is proportional to the applied voltage. The effect of a capacitor in an AC circuit is that the voltage sinusoid will lag behind the current sinusoid, making them out of phase. With a lagging voltage there is a decrease in the real power and increase in the reactive power. The energy stored in the electric field is indicated as the reactive power.

Similarly, an inductor stores energy, but in a magnetic field. The amount of energy stored is proportional to the current flowing through. The effect of an inductor in an AC circuit is that the current sinusoid will lag behind the voltage sinusoid, making them out of phase. With a lagging current there is a decrease in the real power and increase in the reactive power. The energy stored in the magnetic field is indicated as the reactive power.

Similar to the complex power notation, the resistance, capacitance and inductance as voltage-current relations in the phasor domain, can be combined into one complex quantity, the *impedance* Z. Impedance represents the opposition that the circuit exhibits to the flow of sinusoidal current. The general expression of the impedance is given as

$$Z = \frac{V}{I} = R + \boldsymbol{i}X, \tag{2.10}$$

a complex number measured in ohms. The real component R is the resistance and the imaginary component X is the *reactance*. If the reactance is positive, so $X > 0$, the component is inductive and results in a lagging current. If the reactance is negative, $X < 0$, the component is capacitive and results in a lagging voltage. Note that impedance is a ratio of two phasors, but is not itself a phasor.

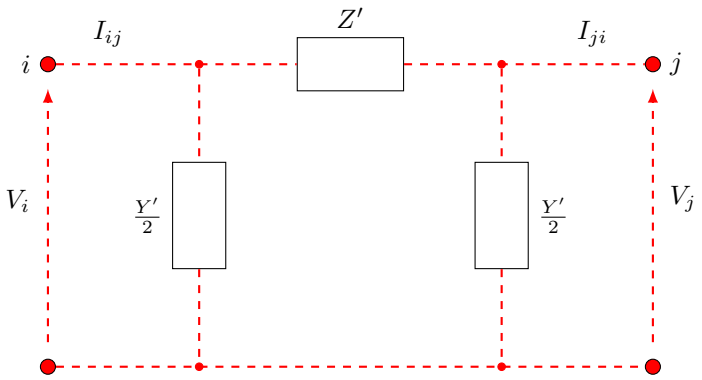


Figure 2.1: Equivalent lumped circuit of a long transmission line.

Conversely, *admittance* Y represents how easily current flows through a circuit. It is defined as the reciprocal of the impedance,

$$Y = \frac{1}{Z} = \frac{R}{R^2 + X^2} + \boldsymbol{i}\frac{X}{R^2 + X^2} = G + \boldsymbol{i}B. \tag{2.11}$$

The real component G is referred to as the *conductance*, while the imaginary component B is called the *susceptance*. Similar but opposite to the reactance, if $B > 0$ the component is capacitive, while it is inductive when $B < 0$.

2.1.2 Transmission Line Modelling

Transmission lines are an important part of power systems modelling, as they transport electricity to the end user. The derivation of different transmission line models can be found in more detail in Chapter 4 of [13] and Appendix E of [12]. This section discusses the properties of transmission lines and uses the derivation of the transmission and admittance matrices to relate sending currents, receiving currents and related voltages. Besides transmission lines, there are also other electricity network components that have to be modelled, most commonly transformers. More about the modelling of transformers can be found in Chapter 6 of [15], Chapter 3 of [12] or Chapter 13 of [14].

There are four different important parameters to consider for the modelling of a transmission line, which are *series resistance*, *inductance*, *shunt capacitance* and *conductance*. The series resistance and inductance refer to resistance and voltage lagging in line with the path of the current, whereas shunt capacitance and conductance refers to leakage current that is in parallel with the transmission line. The series impedance can be written as

$$z = r + \boldsymbol{i}\omega x,$$

with z the series impedance, r the series resistance and x the series inductance. Similarly, the shunt admittance is given by

$$y = g + \boldsymbol{i}\omega b,$$

with y the shunt admittance, g the shunt conductance and b the shunt capacitance. An equivalent circuit of a transmission line can be found in Figure 2.1. The left side of the circuit, with subscripts i is the sending side, while the right side corresponds to the receiving side, indicated with subscripts j. The shunt admittance is split into half, with one part on the sending end and one part on the receiving end of the line. Define the *total series impedance* and *total shunt capacitance* per phase as

$$Z = zL, \quad Y = yL,$$

where L is the transmission line length. The equivalent circuit is given in Figure 2.1 is a so-called *lumped equivalent circuit*. The term 'lumped' refers to the assumption that circuit components are

concentrated at discrete points, rather than being spread out along the entire line. Therefore, we define the equivalent lumped series impedance and shunt admittance as

$$Z' = Z \cdot \frac{\sinh(\gamma L)}{\gamma L}, \quad Y' = Y \cdot \frac{\tanh(\gamma L/2)}{\gamma L/2},$$

respectively. Here, $\gamma = \sqrt{zy}$ is the *propagation constant*. Notice the dependence of the lumped impedance and admittance on the transmission line length L.

Transmission parameters refer to the parameters that relate the input variables V_i, I_{ij} to the output variables V_j, $-I_{ji}$. The corresponding *transmission matrix* of the lumped equivalent circuit in Figure 2.1 follows from two-port theory and is given as

$$\begin{bmatrix} V_i \\ I_{ij} \end{bmatrix} = \begin{bmatrix} \frac{Z'Y'}{2} + 1 & Z' \\ \left(\frac{Z'Y'}{4} + 1\right) Y' & \frac{Z'Y'}{2} + 1 \end{bmatrix} \begin{bmatrix} V_j \\ -I_{ji} \end{bmatrix}. \tag{2.12}$$

Using this transmission matrix for the lumped equivalent circuit we can express sending current I_{ij} in terms of the sending and receiving voltages V_j, V_i,

$$I_{ij} = (V_i - V_j)\frac{1}{Z'} + \frac{1}{2}Y'V_i. \tag{2.13}$$

Note that impedance and admittance are related through (2.11). Therefore we will from now on denote the series elements with the superscript $\cdot^{\text{se}}$, and the shunt elements with the superscript $\cdot^{\text{sh}}$. Substituting Z' and Y' and working out the series and shunt admittances, (2.13) becomes

$$\begin{aligned} I_{ij} &= \left[Y'^{\text{se}} + \frac{1}{2}Y'^{\text{sh}}\right] V_i - Y'^{\text{sh}} V_j, \\ &= \left[\frac{1}{C_1}(G_{ij}^{\text{se}} + \boldsymbol{i}B_{ij}^{\text{se}}) + \frac{C_2}{2}(G_{ij}^{\text{sh}} + \boldsymbol{i}B_{ij}^{\text{sh}})\right] V_i - \left[\frac{1}{C_1}(G_{ij}^{\text{se}} + \boldsymbol{i}B_{ij}^{\text{se}})\right] V_j, \\ &= \Upsilon_{ii,j} V_i - \Upsilon_{ij} V_j, \end{aligned} \tag{2.14}$$

with $C_1 = \frac{\sinh(\gamma L)}{\gamma L}$, $C_2 = \frac{\tanh(\gamma L/2)}{\gamma L/2}$ and coefficients

$$\begin{aligned} \Upsilon_{ii,j} &= \frac{1}{C_1}(G_{ij}^{\text{se}} + \boldsymbol{i}B_{ij}^{\text{se}}) + \frac{C_2}{2}(G_{ij}^{\text{sh}} + \boldsymbol{i}B_{ij}^{\text{sh}}), \\ \Upsilon_{ij} &= \frac{1}{C_1}(G_{ij}^{\text{se}} + \boldsymbol{i}B_{ij}^{\text{se}}). \end{aligned}$$

These coefficients describe the relation between the sending current I_{ij} and related voltages V_i and V_j. The term $\Upsilon_{ii,j}$ is the sum of the shunt and series admittances connected to element i via the transmission line with sending current I_{ij}. The term Υ_{ij} is the series admittance of the direct transmission line between elements i and j. Since admittances are a property of the line itself, $\Upsilon_{ij} = \Upsilon_{ji}$ and $\Upsilon_{ii,j} = \Upsilon_{jj,i}$. The relations from the transmission matrix (2.13) can also be represented by the matrix

$$\begin{bmatrix} I_{ij} \\ -I_{ji} \end{bmatrix} = \begin{bmatrix} \Upsilon_{ii,j} & -\Upsilon_{ij} \\ -\Upsilon_{ij} & \Upsilon_{ii,j} \end{bmatrix} \begin{bmatrix} V_i \\ V_j \end{bmatrix}.$$

The *total current* at element i is defined as the sum of all sending currents leaving the element,

$$I_i = -\sum_{j,j\neq i} I_{ij} = -\sum_{j,j\neq i} \left[\Upsilon_{ii,j} V_i - \Upsilon_{ij} V_j\right]. \tag{2.15}$$

Substitution of the sending current expression in terms of voltages (2.14) into the total current leads to the *admittance matrix* $\boldsymbol{Y}$,

$$\begin{aligned}\begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{bmatrix} &= - \begin{bmatrix} \sum_{j,j\neq 1} \Upsilon_{11,j} & -\Upsilon_{12} & \cdots & -\Upsilon_{1N} \\ -\Upsilon_{12} & \sum_{j,j\neq 2} \Upsilon_{22,j} & \cdots & -\Upsilon_{2N} \\ \vdots & \cdots & \cdots & \vdots \\ -\Upsilon_{1N} & -\Upsilon_{2N} & \cdots & \sum_{j,j\neq N} \Upsilon_{NN,j} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix}, \\ &= - \begin{bmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \cdots & \cdots & \vdots \\ Y_{N1} & Y_{N2} & \cdots & Y_{NN} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_N \end{bmatrix},\end{aligned} \tag{2.16}$$

with $Y_{ii} = \sum_{j,j\neq i} \Upsilon_{ii,j}$ and $Y_{ij} = Y_{ji} = -\Upsilon_{ij}$. The admittance matrix of a circuit can be constructed by noticing that the elements are determined as

- The diagonal terms Y_{ii} are the sum of all admittances directly connected to element i.
- Y_{ij} equals the negative value of the admittance connected between elements i and j.

With the derived expression of sending current in terms of voltages (2.14), we can formulate the complex power S_{ij} at the sending end of the transmission line using (2.9)

$$\begin{aligned} S_{ij} &= V_i I_{ij}^*, \\ &= |V_i| e^{\delta_{V_i} \boldsymbol{i}} \left[\left(|V_i| e^{-\delta_{V_i} \boldsymbol{i}} - |V_j| e^{-\delta_{V_j} \boldsymbol{i}} \right) \frac{G^{\text{se}} - \boldsymbol{i} B^{\text{se}}}{C_1} + \frac{1}{2} \left(G^{\text{sh}} - \boldsymbol{i} B^{\text{sh}} \right) C_2 |V_i| e^{-\delta_{V_i} \boldsymbol{i}} \right], \\ &= P_{ij} + \boldsymbol{i} Q_{ij}, \end{aligned} \tag{2.17}$$

with sending real and reactive powers

$$P_{ij} = \frac{|V_i|^2}{A} G^{\text{se}} - \frac{|V_i||V_j|}{C_1} \left[G^{\text{se}} \cos(\phi_{ij}) + B^{\text{se}} \sin(\phi_{ij}) \right] + \frac{1}{2} |V_i|^2 G^{\text{sh}} C_2, \tag{2.18}$$

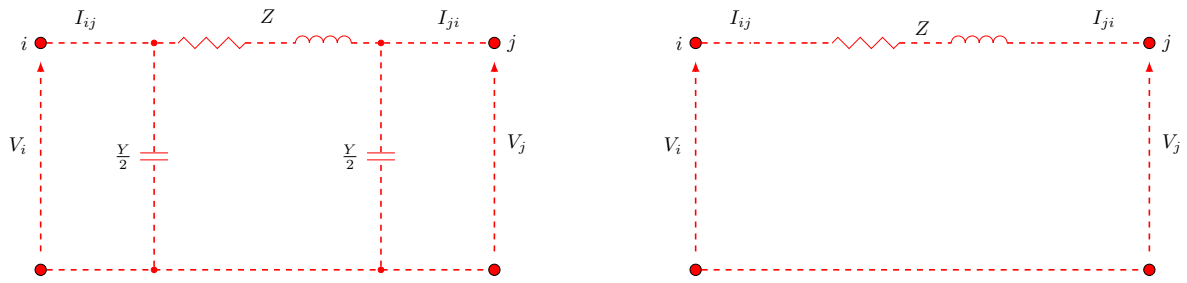
$$Q_{ij} = -\frac{|V_i|^2}{A} B^{\text{se}} - \frac{|V_i||V_j|}{C_1} \left[G^{\text{se}} \sin(\phi_{ij}) - B^{\text{se}} \cos(\phi_{ij}) \right] - \frac{1}{2} |V_i|^2 B^{\text{sh}} C_2. \tag{2.19}$$

Note that the length of the transmission line L appears in the definition of the equivalent lumped impedance and admittance. When modelling a transmission line, one therefore has to distinguish between long, medium and short length lines. In a long transmission line ($L > 240$ km) all components in the series impedance and shunt admittance are non-zero and thus have to be taken into account. For a medium length transmission line ($80 < L < 240$ km), simplifications can be made by assuming that $\sinh(\gamma L) \approx \gamma L$. Therefore $Z' \approx Z$ and $Y' \approx Y$ ($C_1, C_2 = 1$). Additionally, we can assume that the shunt conductance g^{sh} becomes negligible. For a short transmission line ($L < 80$ km), even more simplifications apply such that the shunt capacitance b^{sh} also becomes negligible. For a short transmission line the shunt admittance Y is therefore assumed to be zero. The equivalent circuits of medium- and short-length transmission lines can be found in Figure 2.2.

2.2 Gas Networks

The energy carrier gas refers to any gaseous substance that is transported or consumed for its energy content. Some common gas energy carriers are natural gas, hydrogen and biogas. The gas is transported and converted into other energy carriers, such as electricity and heat, via processes such as combustion. The modelling of a gas network focuses on the transport of gas through pipes based on mass flow and pressure.

This section gives a brief derivation of the mass flow of a gas in a pipe, based on [6], Chapter 4 of [16] and Chapter 2 of [17]. The section ends with a brief discussion on other common gas network elements.



(a) Medium transmission line.

(b) Short transmission line.

Figure 2.2: Equivalent circuits of medium and short transmission lines, taken from [12].

2.2.1 Mass Flow in a Pipe

According to [16], there are many different models and formulas used to describe steady-state flow of gas through a pipe. The gas dynamics within a single pipe are described by a set of PDEs, often called the Euler equations. For our use-case, assumptions have to be made, as we are interested in steady-state isothermal flow. This implies that there is no time-dependence and the gas temperatures are constant. Additionally, assumptions are made to simplify calculations and derivations. Thus we assume negligible kinetic energy change in a pipe, constant compressibility of gas in a pipe, constant friction coefficient along the pipe length, validity of Darcy friction loss relationship across the pipe and horizontal pipes.

Under steady-state isothermal conditions, the governing Euler equations reduce to

$$\frac{\partial p}{\partial x} + f\frac{|v|v}{2D}\rho = 0, \tag{2.20}$$

where we have gas pressure $p(x)$ in pascal, dimensionless constant friction factor f, gas velocity $v(x)$ in meters per second, gas density $\rho(x)$ in kilograms per cubic meter and constant inner pipe diameter D in meter. Denote by $q(p_1, p_2)$ the constant *gas mass flow* in a pipe ($kg \cdot s^{-1}$), depending on the pressure p_1 at the inlet of the pipe, and pressure p_2 at the outlet side of the pipe. Solving (2.20), as is done in detail in [16] and [17], results in the following expression of the mass flow,

$$q(p_1, p_2) = \text{sign}(\Delta p) C^g \sqrt{\frac{|\Delta p|}{f}}, \tag{2.21}$$

with C^g the pipe constant defined differently for high-pressure and low-pressure systems as

$$C^g = \begin{cases} \frac{\pi}{8}\sqrt{\frac{SD^5}{TR_{\text{air}}LZ}}, & \text{High pressure,} \\ \frac{\pi}{8}\sqrt{\frac{2p_nSD^5}{T_nR_{\text{air}}L}}, & \text{Low pressure.} \end{cases} \tag{2.22}$$

The pressure drop Δp in (2.21) is also defined differently for high- and low-pressure systems as

$$\Delta p = \begin{cases} p_1^2 - p_2^2, & \text{High pressure,} \\ p_1 - p_2, & \text{Low pressure.} \end{cases} \tag{2.23}$$

In the pipe constant expression (2.22) we have S the dimensionless specific gravity of the gas, T the constant gas temperature in Kelvin, R_{air} the gas constant of air in Joules per kilogram-Kelvin, L the length of the pipe in meters and Z the dimensionless compressibility factor. For low-pressure systems we were able to make the following additional assumptions:

1. $Z \approx 1$: In a low pressure system the real gas is close to ideal gas.
2. $T \approx T_n$: In the low-pressure system we can assume that the temperature T is close to the standard temperature T_n.

3. $p_1^2 - p_2^2 \approx 2p_n(p_1 - p_2)$: Linearization around p_n, the standard pressure.

There are various models that one can use for the friction factor. It can be assumed that the friction is a constant, dependent on the pipe parameters or dependent on the gas mass flow, $f = f(q)$. Several models are discussed in Chapter 2 of [6] and Chapter 4 of [16].

2.2.2 Gas Network Elements

Besides pipes, there are also other important network elements in gas networks.

- **Valves**: Valves are used to route or block gas flow, and can be either closed or open. If the valve is open, we have $p_i = p_j$, $T_i = T_j$ and $q_{ij} \neq 0$. If the valve is closed the gas is prevented from flowing, which results in $q_{ij} = 0$ and p_i, p_j are decoupled.
- **Control valves:** Also known as *pressure regulators*, control valves are used to interconnect network parts operating at high pressure with the low pressure parts. If the valve is active, the pressure at the outlet can be reduced to a controllable value. As a result of the pressure drop, the temperature of the gas also drops. If the control valve is closed, we have $q_{ij} = 0$ and p_i, p_j are decoupled. If the control valve is active, the pressure difference can be kept within a predetermined range around pressure p_{cv}, so we have $0 \leq p_{cv} - \epsilon \leq \Delta p_{ij} \leq p_{cv} + \epsilon$ and $q_{ij} \geq 0$.
- **Compressor machines**: Compressors are used to increase the pressure of the incoming gas to a higher outflow pressure. There are different types of compressor machines that can be some of the most complex parts of gas networks. A general compressor can be described by $p_j = rp_i$, with the constant r depending on the compressor used, and $q_{ij} \geq 0$.

2.3 Graph Representation of Energy Networks

Energy systems can be represented by graphs to visualize the connection between network components, producers of energy and consumers. This section discusses some general concepts from networks in graph theory to understand network modelling, based on Chapters 1 and 6 of [18]. It is also derived how electricity and gas networks can be interpreted as graphs, using notation, concepts and principles from [16] and [19].

2.3.1 Graphs and Networks

A *graph* $\mathcal{G}$ is a pair $(\mathcal{V}, \mathcal{E})$ of sets such that $\mathcal{E} \subseteq [\mathcal{V}]^2$. The set $\mathcal{V}$ is a set of *nodes or vertices* v_i and $\mathcal{E}$ is a set of pairs of nodes $e_k = \{v_i, v_j\}$, referred to as *links or edges*. A graph is *undirected* if the links $\mathcal{E}$ form an unordered set nodes. Conversely, a graph is *directed* if the links form an ordered set of nodes $e_k = (v_i, v_j)$, so that the link has a direction from node v_i to v_j. The *size or order* of a graph, denoted by $|\mathcal{V}|$, is the number of nodes of the graph. Denote with $|\mathcal{E}|$ the number of links.

The *incidence matrix* $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{E}|}$ describes the interconnection between nodes $\mathcal{V} = \{v_1, \ldots, v_n\}$ and links $\mathcal{E} = \{e_1, \ldots, v_m\}$ in a graph $\mathcal{G}$. If the graph is undirected, the incidence matrix is defined as

$$A_{ik} = \begin{cases} 1, & \text{link } e_k \text{ is connected to node } v_i \\ 0, & \text{otherwise.} \end{cases}$$

If the graph is directed, the incidence matrix is defined as

$$A_{ik} = \begin{cases} 1, & \text{link } e_k \text{ leaves node } v_i, \\ -1, & \text{link } e_k \text{ enters node } v_i, \\ 0, & \text{otherwise.} \end{cases} \tag{2.24}$$

A *terminal link* or *half link* is a special type of link that is only connected to one node and can be denoted by $t_l = \{v_i\}$. By definition, a terminal link can only be connected to one node, while a node

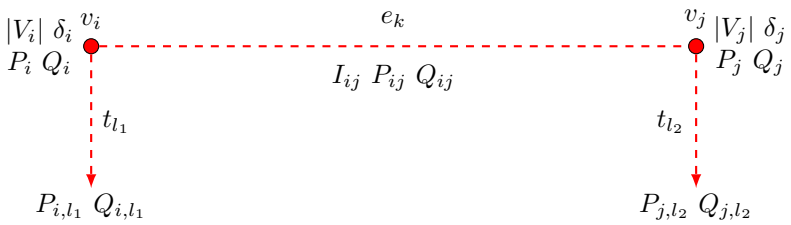


Figure 2.3: Electricity network with two nodes.

can have more than one terminal link connected to it. The set of terminal links t_l in a graph is denoted by $\mathcal{T}$ and the set of terminal links connected to node v_i is denoted by $\mathcal{T}(v_i)$.

A *network* is a graph with an associated physical model. The links of a network carry some kind of flow, such as water, electricity, data, etc. In this report, we will focus on a network that represents the flow/transport of energy along its links, called an *energy (transportation) network*. Denote the energy network by $\mathcal{N} = \{\mathcal{G}, \mathcal{T}\} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}\}$, a graph with a set of nodes and links, and a set of terminal links. Note that if graph $\mathcal{G}$ is (un)directed, then so is network $\mathcal{N}$.

The terminal links in an energy network are used to model inflow (source) and outflow (sinks) of energy. A node can represent such a source or sink, but since flow in a network is defined through links we associate the inflow and outflow of the sources with the terminal links. The terminal links thus represent the in- or outflow of the source or sink.

2.3.2 Electricity Network

A balanced AC power grid is represented by an undirected network, which we denote with $\mathcal{N}^e = \{\mathcal{V}^e, \mathcal{E}^e, \mathcal{T}^e\}$. In an electricity network a node, also called a *bus* in literature, is an electrical component in the circuit. A link represents a transmission line. An example of an electricity network is given in Figure 2.3. Generators and loads are represented by nodes and terminal links, and are modelled as *power injections*, which is the complex power supplied/consumed by the generator/load. It consists of the injected real power P and injected reactive power Q. The injected real and reactive power through terminal link t_l supplied to node v_i is denoted by $P_{i,l}$ and $Q_{i,l}$, respectively. Note that multiple loads or generators can be connected to a single node. Therefore, define the *total real and reactive powers* at node v_i as

$$P_i := \sum_l P_{i,l}, \quad Q_i := \sum_l Q_{i,l}. \tag{2.25}$$

Additionally, we denote the *real power flow* from node v_i to v_j with P_{ij}. Similarly, the *reactive power flow* Q_{ij} denotes the flow of reactive power. The convention is to denote consumed power as positive, and generated power as negative, following the direction of the terminal link arrows in Figure 2.3.

The additional variables of interest are the nodal voltage V and link current I. The voltage at node v_i is denoted as V_i. The current leaving node v_i and flowing to v_j is called I_{ij}. Conversely, the current leaving node v_j and flowing to v_i is denoted as I_{ji}. Note that if a transmission line does not contain any admittances, we have $I_{ij} = -I_{ji}$. Such a line is referred to as a lossless line. If there are series and/or shunt admittances present, we generally have $I_{ij} \neq -I_{ji}$ and the relation between I_{ij} and I_{ji} is determined via the transmission matrix (2.12), as described in Subsection 2.1.2. The *total injected current* at node v_i is given as

$$I_i := \sum_l I_{i,l}.$$

The admittance matrix $\boldsymbol{Y}$ (2.16) or transmission matrix $\boldsymbol{T}$ (2.12) of the network describes the relation between the link current injected at the nodes and the nodal voltages. Note that the admittance matrix describes the currents in terms of voltages, whereas the transmission matrix describes the sending

Table 2.1: Electricity network node types

Node type	Specified	Unknown	Size notation
PQ or load bus	P, Q	$\|V\|, \delta$	$\|\mathcal{PQ}\|$
PV or generator bus	$P, \|V\|$	Q, δ	$\|\mathcal{PV}\|$
Slack bus	$\|V\|, \delta$	P, Q	$\|\mathcal{S}\|$

current and voltage in terms of receiving current and voltage. They can be used interchangeably for the modelling of a single link that connects two nodes. Using the admittance matrix $\boldsymbol{Y}$ or transmission matrix $\boldsymbol{T}$, we can express the link currents I_{ij} in terms of the nodal voltages V_i. Therefore, we can fully describe the state of a electricity network by four nodal parameters. For each node we thus describe

- Voltage phasor magnitude $|V_i|$: The magnitude of the voltage phasor at node v_i,
- Voltage phasor angle δ_i: The angle of the voltage phasor at node v_i,
- Total real power P_i: The net generation or consumption of real power connected to node v_i,
- Total reactive power Q_i: The net generation or consumption of reactive power connected to node v_i.

There are different types of nodes/buses in a system. They represent different types of circuit elements and mathematically differ by which two of the four nodal parameters are known.

- **PQ bus**: Also known as a *load bus*, loads are modelled as constant power sinks. They do not have control over voltage levels, but their consumption is known through the total injected real and reactive powers P_i and Q_i. The unknown quantities are therefore the voltage magnitude $|V_i|$ and voltage phase angle δ_i. The amount of PQ-buses in a system is denoted by $|\mathcal{PQ}|$.
- **PV bus**: At a PV bus, also called a *generator bus*, the total real power P_i supplied by a generator and the voltage magnitude $|V_i|$ are known. In a real-life power system, the voltage at a generator is usually kept within a predetermined range, which is needed for the functioning and efficiency of the power system. The unknown quantities are therefore the reactive power Q_i and voltage phase angle δ_i. The amount of PV-buses in a system is denoted by $|\mathcal{PV}|$.
- **Slack bus**: A slack bus is a special type of bus that acts as a reference point for the entire network. The voltage magnitude $|V_i|$ and voltage phase angle δ_i are specified. Within a power network the total injected power supplied minus the total injected power consumed and total losses must balance to be zero. This balance is done by the slack node(s). The total injected real power P_i and injected reactive power Q_i are thus unknowns. The slack bus can be interpreted as a generator load that injects exactly the remaining power the system needs. The voltage magnitude $|V_i|$ and phase angle δ_i are known quantities and serve as a reference for the other voltages, meaning that we compute the voltage phase angles of the PQ and PV buses relative to the slack bus. There is typically only one slack bus in a system. However, in larger or specific systems there can be multiple slack buses. The amount of slack buses in a system is denoted by $|\mathcal{S}|$.

An overview of electricity network node types can be found in Table 2.1.

2.3.3 Gas Network

A gas system is represented by a directed network, which we denote with $\mathcal{N}^g = \{\mathcal{V}^g, \mathcal{E}^g, \mathcal{T}^g\}$. In a gas network a node can represent a sink, source or junction, which is an intersection of pipes where the gas is redistributed. A link can represent a pipe, compressor, valve and other gas network elements. An example of a gas network is given in Figure 2.4. The arrows in the directed graph show the direction of flow. The gas mass flow from a sink or source through terminal link t_l flowing to node v_i is denoted by

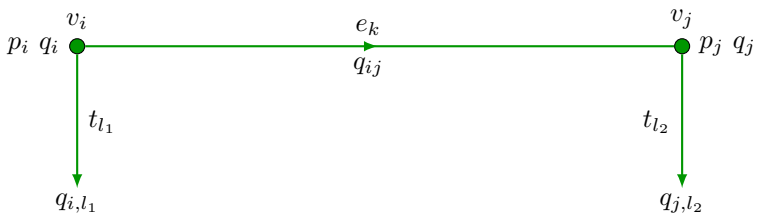


Figure 2.4: Gas network with two nodes.

Table 2.2: Gas network node types

Node type	Specified	Unknown	Size notation
Load node	q	p	$\lvert\mathcal{Q}\rvert$
Slack node	p	q	$\lvert\mathcal{P}\rvert$

$q_{i,l}$. Note that multiple sinks or sources can be connected to a single node. Therefore, the *total injected gas flow* at node v_i is defined as

$$q_i = \sum_l q_{i,l}. \tag{2.26}$$

The convention is to denote a supply of gas as negative and a sink as negative $q_{i,l}$, following the direction of the arrow in Figure 2.4.

We can describe a gas network with the following parameters

- Nodal pressure p_i: The pressure at node v_i,
- Link gas flow q_{ij}: The gas mass flow from node v_i to node v_j.

There are different types of nodes in a gas network system. They represent different network elements and mathematically differ by which of the two nodal parameters are known.

- **Load node**: Loads are modelled as gas flow sinks. Their consumption of gas flow is known through the total injected gas mass flow q_i. The unknown quantity is therefore the pressure p_i. The amount of load nodes in a system is denoted by $|\mathcal{Q}|$.
- **Slack node**: A slack node is a special type of node that acts as a reference point for the entire network. There is only one slack node in a system, and the pressure p_i is specified. Within a gas network, the total gas mass flow must balance to be zero. This balance is done by the slack node where the gas mass flow q_i is unknown. The slack bus can be interpreted as a generator or load that injects (consumes or supplies) exactly the remaining gas flow the system needs. The pressure p_i serves as a reference for the other nodal pressures. We therefore compute the pressure of the load nodes relative to the slack node. The amount of slack nodes in a system is denoted by $|\mathcal{P}|$.

An overview of gas network node types can be found in Table 2.2.

2.4 Load Flow Equations

This section mathematically derives the load flow equations used in simulations of energy networks for electricity and gas networks, based on the principles discussed in previous sections of this chapter.

2.4.1 Load Flow Equations: Electricity Networks

A steady-state AC power grid is completely described by KCL (2.4), an equation linking the current and voltages through admittance (based on Ohm's law), and the complex power equation (2.9). These lead

to the *load flow* or *power flow equations*. They relate the four quantities described in Subsection 2.3.2 at all nodes $v_i \in \mathcal{V}^e$, by expressing nodal real power P_i and nodal reactive power Q_i in terms of system voltage magnitudes and phase angles.

In every node $v_i \in \mathcal{V}^e$, KCL (see Subsection 2.1.1) holds, stating that the sum of all injected currents must equal the sum of all outgoing currents. Following the notation used in Subsection 2.3.2, we can write KCL as

$$\sum_l I_{i,l} = I_i = -\sum_{j,j\neq i} I_{ij}, \tag{2.27}$$

the sum of all injected currents in node v_i, $I_{i,l}$, equals the sum of all currents leaving the node, I_{ij}, where the minus sign follows from the convention of choosing the direction of flow.

Additionally, the complex power equation (see Subsection 2.1.1) holds in every node. It states that in every node v_i, the product of the nodal voltage V_i and complex conjugate of the nodal current I_i can be expressed as a complex number that relates the real and reactive powers P_i and Q_i, so

$$\sum_l S_{i,l} = S_i = V_i(I_i)^* = P_i + \boldsymbol{i}Q_i,$$

with S_i the total injected complex power at node v_i.

Using Ohm's law, we can denote the current I_{ij} from node v_i to node v_j in terms of the voltages and total admittance. The result of this substitution is the admittance matrix (2.16) or the transmission matrix (2.12). The derivation of both matrices can be found in Subsection 2.1.2. Using the transmission matrix we can substitute (2.14) into KCL (2.27), resulting in

$$\begin{aligned} I_i &= -\sum_{j,j\neq i} \left(\Upsilon_{ii,j}V_i + \Upsilon_{ij}V_j\right), \\ &= -\sum_j Y_{ij}V_j. \end{aligned}$$

Here, Y_{ij} is a component of the admittance matrix $\boldsymbol{Y}$. Therefore, one can alternatively directly use the admittance matrix (2.16) in place of substitution of the transmission matrix into KCL to obtain this same result. Denote the real and complex parts of the admittance term $\boldsymbol{Y}_{ij}$ as $\mathrm{Re}(\boldsymbol{Y}_{ij}) = G_{ij}$ and $\mathrm{Im}(\boldsymbol{Y}_{ij}) = B_{ij}$. As a result, we can compute the complex conjugate of the current, I_i^*,

$$I_i^* = -\sum_j Y_{ij}^* V_j* = -\sum_j \left(G_{ij} - \boldsymbol{i}B_{ij}\right)\left(|V_j|e^{-\delta_j \boldsymbol{i}}\right).$$

Using this expression, the nodal complex power equation can be expressed in nodal real and reactive powers,

$$\begin{aligned} S_i &= V_i(I_i)^*, \\ &= -|V_i|e^{\delta_i \boldsymbol{i}}\left[\sum_j \left(G_{ij} - \boldsymbol{i}B_{ij}\right)\left(|V_j|e^{-\delta_j \boldsymbol{i}}\right)\right], \\ &= -|V_i|^2\left(G_{ii} - \boldsymbol{i}B_{ii}\right) - \sum_{j,j\neq i}\left(G_{ij} - \boldsymbol{i}B_{ij}\right)|V_i||V_j|e^{(\delta_i-\delta_j)\boldsymbol{i}}, \\ &= -\left[|V_i|^2 G_{ii} + \sum_{j,j\neq i}|V_i||V_j|\left(G_{ij}\cos\delta_{ij} + B_{ij}\sin\delta_{ij}\right)\right] - \\ &\quad \boldsymbol{i}\left[-|V_i|^2 B_{ii} - \sum_{j,j\neq i}|V_i||V_j|\left(B_{ij}\cos\delta_{ij} - G_{ij}\sin\delta_{ij}\right)\right], \\ &= P_i + \boldsymbol{i}Q_i, \end{aligned} \tag{2.28}$$

where $\delta_{ij} = \delta_i - \delta_j$. This result for the complex power equation leads us to the formulation of the load for equations for electricity networks.

Load Flow Equations

The load flow equations for an electricity network can thus be written as the following non-linear system of equations

$$\boldsymbol{F}^e(\boldsymbol{x}^e) = \boldsymbol{0}. \tag{2.29}$$

The argument $\boldsymbol{x}^e$ is defined as

$$\boldsymbol{x}^e = \begin{bmatrix} \boldsymbol{\delta} \\ |\boldsymbol{V}| \end{bmatrix}.$$

The vectors $\boldsymbol{\delta}$ and $\boldsymbol{V}$ contain the unknown nodal voltage angles and magnitudes. The number of unknown voltage angles is $|\mathcal{V}^e| - |\mathcal{S}|$, as the slack nodes have a known voltage angle. Similarly, the number of unknown voltage magnitudes is $|\mathcal{V}^e| - |\mathcal{S}| - |\mathcal{PV}|$, as the slack nodes and PV nodes have prescribed voltage magnitude. The vector $\boldsymbol{x}^e$ thus has $2(|\mathcal{V}^e| - |\mathcal{S}|) - |\mathcal{PV}|$ entries.

The function $\boldsymbol{F}^e(\boldsymbol{x}^e)$ is defined as

$$\boldsymbol{F}^e(\boldsymbol{x}^e) = \begin{bmatrix} \boldsymbol{F}^P(\boldsymbol{x}^e) \\ \boldsymbol{F}^Q(\boldsymbol{x}^e) \end{bmatrix}. \tag{2.30}$$

The vectors $\boldsymbol{F}^P(\boldsymbol{x}^e)$ and $\boldsymbol{F}^Q(\boldsymbol{x}^e)$ contain the complex power equation for the nodes with known injected real and reactive power, respectively. The number of nodes with known total real power is $|\mathcal{PV}| + |\mathcal{PQ}|$, whereas the number of nodes with known total reactive power is $|\mathcal{PQ}|$. The vector $\boldsymbol{F}^e(\boldsymbol{x}^e)$ thus has $|\mathcal{PV}| + 2|\mathcal{PQ}|$ components. Suppose that the $\tilde{i}$-th entry of $\boldsymbol{F}^P(\boldsymbol{x}^e)$ corresponds to node v_{i_1}, which has known total real power, and that the $\hat{i}$-th entry of $\boldsymbol{F}^Q(\boldsymbol{x}^e)$ corresponds to node v_{i_2}, which has known total real power, then the entries of $\boldsymbol{F}^e(\boldsymbol{x}^e)$ are determined using the result in (2.28),

$$\begin{aligned} \boldsymbol{F}^P_{\tilde{i}}(\boldsymbol{x}^e) &= P_{i_1} + |V_{i_1}|^2 G_{i_1 i_1} + \sum_{j, j \neq i_1} |V_{i_1}||V_j| \left(G_{i_1 j} \cos \delta_{i_1 j} + B_{i_1 j} \sin \delta_{i_1 j}\right), \\ \boldsymbol{F}^Q_{\hat{i}}(\boldsymbol{x}^e) &= Q_{i_2} - |V_{i_2}|^2 B_{i_2 i_2} - \sum_{j, j \neq i_2} |V_{i_2}||V_j| \left(B_{i_2 j} \cos \delta_{i_2 j} - G_{i_2 j} \sin \delta_{i_2 j}\right). \end{aligned} \tag{2.31}$$

Recall that P_{i_1} and Q_{i_2} are defined as the sum of injected powers, (2.25).

Note that $|\mathcal{V}^e| = |\mathcal{PQ}| + |\mathcal{PV}| + |\mathcal{S}|$, so $|\mathcal{PV}| + 2|\mathcal{PQ}| = 2(|\mathcal{V}^e| - |\mathcal{S}|) - |\mathcal{PV}|$. The total system (2.29) thus contains $|\mathcal{PV}| + 2|\mathcal{PQ}|$ equations with $|\mathcal{PV}| + 2|\mathcal{PQ}|$ unknown variables.

2.4.2 Load Flow Equations: Gas Networks

A steady-state gas network can be completely described by conservation of mass and link equations that model the network elements the links represent. This leads to the *load flow equations* that relate the nodal pressures and total injected gas flow described in Subsection 2.3.3 at all nodes v_i.

In every node $v_i \in \mathcal{V}^g$, conservation of mass must hold. Therefore, the sum of all flows going into node v_i and all flows going out of node v_i must balance to be zero. This leads to

$$\sum_{j, j \neq i} q_{ji} - \sum_{j, j \neq i} q_{ij} - q_i = 0. \tag{2.32}$$

The first term corresponds to gas flowing into node v_i from other connected nodes, the second corresponds to the gas flowing out of node v_i to other connected nodes, and the final term q_i is the total injected gas flow (2.26). The signs follow from the convention of choosing the direction of flow.

Additionally, each link has a link equation, which differs per gas network element. If a link represents a pipe, then the results of Subsection 2.2.1 can be used, leading to the pipe link equation. For the pipe that allows for gas mass flow between nodes v_i and v_j, we have pipe link equation

$$q_{ij}(p_i, p_j) = \text{sign}(\Delta p_{ij}) C^g \sqrt{\frac{|\Delta p_{ij}|}{f}}, \tag{2.33}$$

where Δp_{ij} represents the pressure difference between the nodes v_i and v_j, defined in (2.23), and constant C^g is given by (2.22). Alternatively, instead of expressing the gas mass flow in terms of pressure difference as $q(\Delta p)$, one can express the pressure difference in terms of gas mass flow as $\Delta p(q)$. This leads to equivalent pipe link equation

$$\Delta p_{ij} = (C^g)^{-2} f |q_{ij}| q_{ij}. \tag{2.34}$$

Load Flow Equations

The load flow equations for a gas network can be written as the following system of equations

$$\boldsymbol{F}(\boldsymbol{x}^g) = \boldsymbol{0}. \tag{2.35}$$

There are various formulations used for the load flow equations of a gas network. All describe the same network but define $\boldsymbol{F}$ and $\boldsymbol{x}^g$ differently. The formulations used are the nodal, loop, nodal-loop and full formulations. In the nodal formulation there is a substitution of link equations (2.33) into the conservation of mass equations (2.32), as is done in [16]. The unknown quantities then consist of the unknown total injected gas flows and the unknown nodal pressures. However, one can only substitute link equations into conservation of mass if the link equations contain the variable q_{ij}. If the link represents a pipe, this is the case, see (2.33), but if the link represents different network element, such as a compressor, the gas mass flow term is not present in the equation, see Subsection 2.2.2. The nodal formulation is therefore only used where all links represent pipes. In the full formulation, there is no substitution. The unknown quantities consist of the unknown nodal pressures, the unknown total injected gas flows and gas mass flows along all links. This formulation is more widely applicable as it can include all gas network elements. Therefore, the full formulation is considered in the remainder of this report.

In the full formulation, the argument $\boldsymbol{x}^g$ is defined as

$$\boldsymbol{x}^g = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix}.$$

The vector $\boldsymbol{q}$ contains the unknown link flows. Since all link flows are unknown, this vector is of length $|\mathcal{E}^g|$. The vector $\boldsymbol{p}$ consists of all unknown nodal pressures. The number of unknown nodal pressures is $|\mathcal{V}^g| - |\mathcal{P}| = |\mathcal{Q}|$, the total number of nodes minus the number of slack nodes with specified nodal pressure. The vector $\boldsymbol{x}^g$ thus has $|\mathcal{E}^g| + |\mathcal{Q}|$ entries.

The function $\boldsymbol{F}^g(\boldsymbol{x}^g)$ is defined as

$$\boldsymbol{F}^g(\boldsymbol{x}^g) = \begin{bmatrix} \boldsymbol{F}^q(\boldsymbol{x}^g) \\ \boldsymbol{F}^L(\boldsymbol{x}^g) \end{bmatrix}. \tag{2.36}$$

The vector $\boldsymbol{F}^q(\boldsymbol{x}^g)$ is the vector of conservation of mass, and contains the components as in (2.32) for only the nodes with known total injected gas flow. Suppose that the $\tilde{i}$-th entry of $\boldsymbol{F}^q(\boldsymbol{x}^g)$ corresponds to node v_i, which has known total injected gas flow q_i, then we have

$$\boldsymbol{F}^q_{\tilde{i}}(\boldsymbol{x}^g) = \sum_{j, j \neq i} q_{ji} - \sum_{j, j \neq i} q_{ij} - q_i.$$

Note that this is a linear relation of components and we can therefore write conservation of mass as a linear system using the directed incidence matrix of the network as defined in (2.24). Since we are

only considering the conservation of mass equation for nodes with known total injected gas flow, the *reduced* incidence matrix $\boldsymbol{A}^{\boldsymbol{g}\prime}$ of the network is used, which only contains the rows of the incidence matrix corresponding to the nodes with known total injected gas flow. This result in the linear system

$$\boldsymbol{F}^q = \boldsymbol{A}^{\boldsymbol{g}\prime}\boldsymbol{q} - \boldsymbol{q}^{\text{inj}}, \tag{2.37}$$

where $\boldsymbol{q}^{\text{inj}}$ is the vector with all known total injected gas flows. The vector $\boldsymbol{F}^q$ thus has length $|\mathcal{Q}|$.

The vector $\boldsymbol{F}^L(\boldsymbol{x}^g)$ is the vector of unknown link flows. If the network consists of only pipes, the link equations are given by (2.33) or (2.34). In this derivation, we will use (2.34), as this formulation is generally more robust when linearising and solving the load flow equations. Additionally, for a low pressure system, (2.34) results in an equation that is linear in the pressures. Suppose that the k-th entry of $\boldsymbol{F}^L(\boldsymbol{x}^g)$ corresponds to a link (gas pipe) from node v_i to node v_j, then we have

$$\boldsymbol{F}^L_k(\boldsymbol{x}^g) = \Delta p_{ij} - (C^g)^{-2} f|q_{ij}|q_{ij}. \tag{2.38}$$

The vector $\boldsymbol{F}^L(\boldsymbol{x}^g)$ is of length $|\mathcal{E}^g|$, as we assume all link flows are unknown.

The total system (2.36) thus contains $|\mathcal{Q}| + |\mathcal{E}^g|$ equations with $|\mathcal{Q}| + |\mathcal{E}^g|$ unknowns.

Chapter 3

Multi-Carrier Energy Networks

Different energy carriers such as electricity, gas and heat, can interact with each other, leading to one combined system referred to as a multi-energy system (MES). Such a system can be represented by a multi-carrier energy network (MCN) by coupling nodes of different carriers. Such an integrated network can be used to more realistically model and analyse the energy grid.

This chapter discusses various methods of coupling that can be used for a multi-carrier energy network in Section 3.1 and includes the formulation introduced in [6]. The integrated load flow equations and node types of an integrated system can be found in Section 3.2.

3.1 Coupling

The coupling of single-carrier networks to form a multi-carrier network is a key element of integrated network modelling. This section first introduces types of coupling units that convert energy from one carrier to another. These coupling units can be included in the multi-carrier network in various ways, which is discussed next. Finally, the coupling method proposed in [6] is described in more detail.

3.1.1 Coupling Units

In an MES, the single-carrier systems interact with each other through various elements, called *coupling units*. These elements represent conversion units that transfer energy from one carrier to one or more other carriers. Table 3.1 shows an overview of conversion units in an MES that transform energy from one carrier to another. A combined heat and power plant (CHP) makes use of the residual heat that might occur in a coupling unit such that less energy is lost during conversion.

3.1.2 Methods for Coupling Single-Carrier Networks

As described in [6] there are three main methods for generally coupling different energy carriers:

Table 3.1: Conversion units and examples in multi-carrier energy networks

Input carrier	Output carriers	Abbreviation	Example
Gas	Electricity	G2P	Gas-fired generator
Gas	Heat	G2H	Gas boiler
Gas	Electricity, heat	CHP	Combined heat and power plant
Electricity	Gas	P2G	Electrolyser
Electricity	Heat	E2H	Electric boiler

1. Coupling link: Connect two nodes of a different energy carrier by a link. This method has no straightforward interpretation of a coupling link and it is less intuitive how more than two different energy carriers are coupled.
2. Merged node: Merge two nodes of a different carrier into one node. This method has the practical downside that nodal parameters might have to be doubly defined. Additionally, it has to be possible to differentiate between links of different carriers connected to the merged node in order to apply Kirchhoff's law in constructing the load flow equations.
3. Coupling nodes and dummy links: Introduce an additional coupling node and connect the two different energy carrier nodes with dummy links to the coupling node. With this method the models of the single-carrier parts are only slightly altered and the connection between the carriers is shown explicitly.

In the rest of this report the coupling of single-carrier parts is considered to be done with coupling nodes and dummy links.

3.1.3 Coupling Nodes and Dummy Links

The *coupling node* represents an MES element as described in Subsection 3.1.1 that couples at least two energy carriers. An example of a graph representation of an MES with one coupling node is given in Figure 3.1. By definition, no parameters are associated with the coupling node and it is not part of any of the single-carrier parts, but rather belongs to its own coupling part. If a coupling node couples nodes of the same carrier, it is referred to as *homogeneous*. Such a node can be introduced when for example coupling high-voltage and low-voltage electricity networks. A network with only homogeneous (coupling) nodes is a homogeneous network. Conversely, if a coupling node couples nodes of different carriers, it is referred to as *heterogeneous*. A network with at least one heterogeneous coupling node is a heterogeneous network. The total number of coupling nodes in an MES is denoted by $\mathcal{V}^c$.

The different single-carrier parts in an MES are connected to the coupling node via *dummy links*. They only represent connections between the single-carrier parts and therefore do not represent physical system components. However, they have the same parameter types associated with them as the links of the single-carrier network they are connected to and could therefore be seen as lossless links. Gas and electricity dummy links are interpreted as follows:

- Gas dummy link: A gas dummy link only has a gas mass flow q along the link but not a link equation.
- Electricity dummy link: An electricity dummy link has a current I and real and reactive powers P and Q associated with it. Since the dummy link can be interpreted as a lossless power line, it must therefore hold that $I_{ij} = -I_{ji}$ as well as that $P_{ij} = -P_{ji}$ and $Q_{ij} = -Q_{ji}$.

Following the introduction of dummy links, the MES can be easily separated in single-carrier parts by transforming the dummy links into terminal links. This flexibility allows for general application to MES.

3.2 Load Flow Equations

Using the coupling method described in Section 3.1, this section describes the formulation of a multi-carrier network. The newly introduced coupling equations and occurance of new node types lead to the load flow equations of a multi-carrier energy network.

3.2.1 Coupling Equations

In every heterogeneous coupling node one or more (non)linear coupling equations hold. The general form of the nodal coupling equations associated with coupling node $v_i \in \mathcal{V}^c$ connected to homogeneous

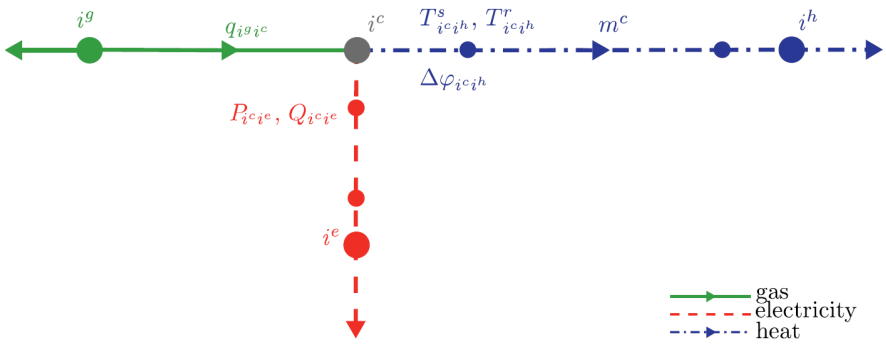


Figure 3.1: Taken from [6]: 'Network representation of a coupling node, showing quantities of interest for LF and the network elements they are associated with.'

node v_j is given as

$$\boldsymbol{F}_i^c(\boldsymbol{x}^c) = 0. \tag{3.1}$$

The coupling parameters are given as

$$\boldsymbol{x}^c = \begin{bmatrix} \boldsymbol{q}_c \\ \boldsymbol{P}_c \\ \boldsymbol{Q}_c \end{bmatrix}, \tag{3.2}$$

where $\boldsymbol{q}_c$ is the vector of gas mass flows of the gas dummy links, and $\boldsymbol{P}_c$ and $\boldsymbol{Q}_c$ are the real and reactive power of the electricity dummy links.

3.2.2 Additional Node Types

Generally, the coupling of single-carrier networks results in more added variables than added equations. A necessary condition for a well-posed linear system is to have as many variables as equations, called a square system. The load flow equations of the electricity and gas are generally non-linear. As a result, it is no longer necessary to have as many unknown variables as equations for the system to be well-posed. However, these linear systems are often linearized during the solution process, such as in the Newton-Raphson (NR) method. A necessary condition for the existence of the inverse of the Jacobian in NR is that it must be square. It is thus generally desirable for the non-linear system of equations for an MES to be square by having as many unknown variables as equations. In order for the multi-carrier load flow problem to have as many equations as unknown parameters, we require additional parameters to be known beforehand. More practically, the interpretations of the scenarios modelled using coupling nodes often naturally result in more known parameters in order to solve the load flow problem. For example, parameters of the slack node in one single-carrier network can be replaced by the coupling parameters, essentially using the other coupled single-carrier network as the slack. Consequently, new node types are introduced that only occur in the multi-carrier networks.

Possible gas and electricity node types in a multi-carrier network are given in Table 3.2 and 3.3, respectively. Mathematically, more electrical node types are possible by combining different known and unknown parameters. However, the node types mentioned in Table 3.2 and 3.3 are the node types that are generally physically possible in an MES with an electrical component. As a result of the additional node types and thus a possible transfer of the slack function from one carrier to another, it can occur that the load flow problem of one single-carrier network is overdetermined, while the load flow problem of the other single-carrier network is underdetermined. However, the final multi-carrier network is a square system.

Table 3.2: Gas node types in a multi-carrier network

Node type	Specified	Unknown
Load node	q	p
Slack node	p	q
Reference load node	p, q	-
Slack node	-	p, q

Table 3.3: Electricity node types in a multi-carrier network

Node type	Specified	Unknown
PQ or load bus	P, Q	$\|V\|, \delta$
PV or generator bus	$P, \|V\|$	Q, δ
Slack bus	$\|V\|, \delta$	P, Q
PQδ bus	P, Q, δ	$\|V\|$
QVδ bus	$Q, \|V\|, \delta$	P
PQVδ bus	$P, Q, \|V\|, \delta$	-

3.2.3 Steady-State Load Flow Problem of Multi-Energy Systems

To construct the load flow equations of a coupled network, one first constructs the load flow equations of the single-carrier parts according to carrier-specific formulation. The load flow equations for electricity and gas are described in Sections 2.4.1 and 2.4.2, respectively. During the construction of the single-carrier components the dummy link variables are taken into account, whereas the coupling nodes are not. The load flow equation(s) for the coupling nodes follow from the specific unit used for coupling and are of general form (3.1). Combining the load flow equations of the individual carriers and the coupling nodes results in the following MES load flow problem

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{F}^g(\boldsymbol{x}^g, \boldsymbol{x}^c) \\ \boldsymbol{F}^e(\boldsymbol{x}^e, \boldsymbol{x}^c) \\ \boldsymbol{F}^c(\boldsymbol{x}^c) \end{bmatrix} = \boldsymbol{0}, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^g \\ \boldsymbol{x}^e \\ \boldsymbol{x}^c \end{bmatrix}. \tag{3.3}$$

The load flow expressions for the gas network in the multi-carrier system are included in $\boldsymbol{F}^g$ that depend on the gas network parameters included in $\boldsymbol{x}^g$, the nodal pressures and link gas mass flows, as well as on the dummy link gas mass flows included in $\boldsymbol{x}^c$. Similarly, the load flow expressions corresponding to the electricity network are included in $\boldsymbol{F}^e$ that depend on the electricity network parameters in $\boldsymbol{x}^e$, which are the nodal voltage magnitudes and angles. Additionally, electricity load flow also depends on the real and reactive powers of the dummy links included in $\boldsymbol{x}^c$. Finally, the nodal coupling expressions are included in $\boldsymbol{F}^c$ that model the coupling units. The coupling equations only depend on the dummy link coupling variables $\boldsymbol{x}^c$.

Chapter 4

Solvers and Preconditioners

An energy network can be represented by a system of equations. The systems of equations for electricity, gas and heat networks are described in Chapter 2. The systems of equations for a multi-energy system where different energy carriers are combined are described in Chapter 3. In order to solve the energy flow problems they represent, these systems of generally non-linear equations have to be solved.

A general linear system of equations can be written as a matrix-vector product

$$A\boldsymbol{x} = \boldsymbol{f}, \tag{4.1}$$

with $A \in \mathbb{R}^{n \times n}$ a non-singular coefficient matrix, $\boldsymbol{f} \in \mathbb{R}^n$ a right-hand side or forcing vector and $\boldsymbol{x} \in \mathbb{R}^n$ the unknown solution vector. The goal is to solve this system reliably and efficiently. Generally, a direct computation of the inverse A^{-1} is computationally expensive and inefficient, especially for larger systems. Different solution methods have been developed to determine the solution vector $\boldsymbol{x}$ without directly computing the inverse. They are either direct methods or iterative methods. In a direct solution method the coefficient matrix A is factorized into a product of matrices in such a way that computation of solution vector $\boldsymbol{x}$ is simplified and less expensive. Direct solution methods that are used in solving multi-energy system problems can be found in Section 4.1. The second class of solution methods is based on an iterative process that aims to improve an approximation of the solution vector $\boldsymbol{x}$ or minimise errors in each iteration. Such iterative methods are often used for large systems due to their efficiency compared to direct solution methods. In [20] and [21], it is concluded that an iterative method is the optimal method for large-scale power systems. Iterative methods that are used in solving multi-energy system problems can be found in Section 4.2. Preconditioners play an important role in iterative solution methods, that improve the efficiency and robustness. They are discussed in Section 4.3. The next section in this chapter briefly introduces the Newton-Raphson (NR) method, which iteratively solved non-linear systems of equations, such as the load flow equations of energy networks. Finally, Section 4.5 discusses domain decomposition methods. The theory on direct and iterative solvers for linear systems is based on methods described in [22] and [23].

4.1 Direct Solution Methods

A commonly used direct solution method is *LU factorization*. It is based on the Gaussian elimination principle. In this method one factorizes the coefficient matrix A as

$$A = LU, \tag{4.2}$$

with L a lower-triangular matrix and U an upper-triangular matrix. This factorization is unique with the requirement that all diagonal elements of either L or U consist only of ones. The solution vector $\boldsymbol{x}$ is then found by solving the systems

$$L\boldsymbol{y} = \boldsymbol{f}, \quad U\boldsymbol{x} = \boldsymbol{y}.$$

The lower-triangular system with $L\boldsymbol{y} = \boldsymbol{f}$ is solved first using forward substitution. With the intermediate result $\boldsymbol{y}$, the upper-triangular system with $U\boldsymbol{y} = \boldsymbol{x}$ is solved using backward substitution, resulting in the solution vector $\boldsymbol{x}$.

Determining the solution vector $\boldsymbol{x}$ with LU factorization consists of computing the factorization (4.2), the forward substitution and backward substitution. For a dense coefficient matrix, the LU decomposition requires $\mathcal{O}(n^3)$ flops. The computational cost of both substitution steps is $\mathcal{O}(n^2)$ flops. The total complexity of determining the solution vector $\boldsymbol{x}$ is thus dominated by the factorization of the coefficient matrix.

LU factorization is used in solving load flow equations of single-carrier as well as multi-carrier networks. It is often used as the linear solution step in the non-linear NR solution method, as is mentioned in for example [24] and done in [25]. The NR method is discussed in more detail in 4.4.1.

4.2 Iterative Solvers for Linear Systems

In an iterative solution method one starts with an initial guess $\boldsymbol{x}_0$ of the solution vector and aims to iteratively improve the approximation of $\boldsymbol{x}$. When after k iterations, the iterate $\boldsymbol{x}_k$ is sufficiently close to the exact solution $\boldsymbol{x}$, $\boldsymbol{x}_k$ can be used as an approximation. The *error vector* of the k-th iteration is defined as the difference between the exact solution and the solution approximation,

$$\boldsymbol{e}_k = \boldsymbol{x} - \boldsymbol{x}_k.$$

In practice the exact solution $\boldsymbol{x}$ is unknown. Therefore the *residual vector* is used instead. The residual vector of the k-th iteration is the difference between the right-hand side and left-hand side of (4.1),

$$\boldsymbol{r}_k = \boldsymbol{f} - A\boldsymbol{x}_k.$$

The residual can be computed for every solution approximation $\boldsymbol{x}_k$ and is used as a measure of the quality of the approximation. Note that the residual vector can be computed from the error vector via the residual equation $A\boldsymbol{e}_k = \boldsymbol{r}_k$.

There are various classes and types of iterative methods. The *basic iterative methods (BIMs)* such as Jacobi, Gauss-Seidel and Successive Overrelaxation are methods based on relaxation steps. In each iteration one or more components of the approximation are modified in a certain order such that components in the residual vector reduce to zero. These methods are rarely used on their own in practice, but form the basis of more advanced methods. Since they are not often used in solving load-flow equations, they are not discussed in more detail in this report. More about BIMs can be found in Chapter 4 of [22].

Other classes of methods employed in energy system modelling are *Krylov subspace* methods. These methods are commonly used in practice in many different fields of research and are discussed in Subsection 4.2.1.

4.2.1 Krylov Subspace Methods

The class of *Krylov subspace* methods is based on generating solution vector approximations $\boldsymbol{x}_k$ of the form

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(A, \boldsymbol{r}_0), \tag{4.3}$$

where $\mathcal{K}_k(A, \boldsymbol{r}_0)$ is the Krylov subspace, defined as

$$\mathcal{K}_k(A, \boldsymbol{r}_0) = \text{span}\left\{\boldsymbol{r}_0, A\boldsymbol{r}_0, \ldots, A^{k-1}\boldsymbol{r}_0\right\}. \tag{4.4}$$

Krylov subspace methods can be interpreted as projection methods. In a general projection method the k-th solution approximation is based on the initial guess $\boldsymbol{x}_0$, updated with a vector from the search

subspace $\mathcal{K}_k$ such that the new residual has no components in the left subspace $\mathcal{L}_k$ (the new residual is orthogonal to all components in $\mathcal{L}_k$),

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{x}_0 + \boldsymbol{\delta}, \quad \boldsymbol{\delta} \in \mathcal{K}_k, \\ (\boldsymbol{r}_0 - A\boldsymbol{\delta}, \boldsymbol{\omega}) &= 0, \qquad\quad \forall \boldsymbol{\omega} \in \mathcal{L}_k. \end{aligned}$$

In Krylov subspace methods the search subspace $\mathcal{K}_k$ is the Krylov subspace $\mathcal{K}_k(A, \boldsymbol{r}_0)$ as defined in (4.4). The choice of subspace $\mathcal{L}_k$ in part determines the type of Krylov subspace method. In orthogonal projection methods one chooses $\mathcal{L}_k = \mathcal{K}_k$, whereas in oblique methods one takes $\mathcal{L}_k = A\mathcal{K}_k$.

The general procedure of a Krylov subspace method is described in Algorithm 1. As can be seen from the general Krylov subspace method algorithm, there are different methods used to compute the expanded basis vectors of the Krylov and left subspaces as well as computing the reduced systems. Two main methods to compute an orthogonal basis of the Krylov subspace are Arnoldi's orthogonalization method and Lanczos' biorthogonalization method. These methods give rise to different solvers.

Algorithm 1 General Krylov Subspace Method

```
Choose initial guess x_0 and compute initial residual r_0 = f − Ax_0
for k = 1, 2, ..., n − 1 do
    Expand the Krylov subspace K_k(A, r_0) = span{r_0, Ar_0, ..., A^{k−1} r_0}
    Generate/update basis vectors V_k of K_k(A, r_0) and possibly W_k of L_k (method-dependent)
    Form the projected matrix H_k = W_k^T A V_k
    Solve the reduced system (method-dependent) for y_k
    Update approximate solution x_k = x_0 + V_k y_k
    Compute residual r_k = f − Ax_k
    if converged then
        break
    end if
end for
```

Arnoldi's orthogonalization method computes an orthonormal basis of a Krylov subspace $\mathcal{K}_{k+1}(A, \boldsymbol{r}_0)$. It determines the basis vector $\boldsymbol{v}_{k+1}$ that contains the components of $A^k\boldsymbol{v}_0$ not spanned by span $\{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1}\} =$ span $\left\{\boldsymbol{v}_0, A\boldsymbol{v}_0, \ldots, A^{k-1}\boldsymbol{v}_0\right\}$. This is done by multiplying previous basis vector $\boldsymbol{v}_k$ by A and orthonormalising it against all previous basis vectors by a standard Gram-Schmidt procedure. As a result, the basis V_k satisfies $V_k^T V_k = I$. The Arnoldi method results in the Arnoldi relation,

$$AV_k = V_k H_k + \boldsymbol{w}_k e_k^T, \quad V_k^T A V_k = H_k, \tag{4.5}$$

where H_k is the *Hessenberg* matrix. It represents how the coefficient matrix A acts inside the Krylov subspace. The shape of H_k is called upper-Hessenberg, where all entries below the first sub-diagonal are zero. It thus follows that solving a system with the Hessenberg matrix H_k is generally cheaper than solving a system with the original coefficient matrix A.

Lanczos' bi-orthogonalization algorithm forms dual bases of the right subspace $\mathcal{K}_k(A, \boldsymbol{r}_0)$ and left subspace $\mathcal{K}_k(A^T, \boldsymbol{r}_0)$ that satisfy the bi-orthogonality condition where each left vector $\boldsymbol{w}_i$ is orthogonal to all right vectors $\boldsymbol{v}_j$, except $\boldsymbol{v}_i$. This is done by multiplying previous basis vectors $\boldsymbol{v}_k$ and $\boldsymbol{w}_k$ by A and A^T, respectively. As a result of the bi-orthogonality, $A\boldsymbol{v}_k$ and $A^T\boldsymbol{w}_k$ is spanned by a linear combination of the previous two basis vectors and a new residual component. These new vectors are computed and orthogonalized against the previous two vectors in each sequence. As a result, the dual bases V_k and W_k satisfy the relation $W_k^T V_k = I$. The bi-Lanczos algorithm results in the following relations,

$$AV_k = V_k T_k + \delta_{k+1}\boldsymbol{v}_k e_k^T, \quad A^T W_k = W_k T_k^T + \beta_{k+1}\boldsymbol{w}_k e_k^T, \quad W_k^T A V_k = T_k, \tag{4.6}$$

where T_k is a tri-diagonal matrix. The bases V_k and W_k are dual and as a result, the projection of the coefficient matrix A from the oblique projection process onto $\mathcal{K}_k(A^T, \boldsymbol{r}_0)$ and orthogonally to $\mathcal{K}_k(A, \boldsymbol{r}_0)$,

is a tri-diagonal matrix. It follows that solving systems with T_k is cheaper than solving systems with A.

The bi-Lanczos algorithm is cheaper to execute than the Arnoldi algorithm. In bi-Lanczos, basis vectors have to be orthogonalized against only the previous two, whereas for Arnoldi the new basis vector has to be orthogonalized against all previous basis vectors. Additionally, the matrix T_k requires less storage than the full Hessenberg matrix H_k. However, in the bi-Lanczos algorithm there are more opportunities for breakdown due to sensitivity to rounding errors.

In some scenarios the coefficient matrix A is symmetric positive definite (SPD), for example when the Poisson equation is solved using a finite difference method. When the coefficient matrix is SPD, the Arnoldi method reduces to Lanczos. The SPD property is an advantageous property of the coefficient matrix to have and there are therefore several methods that make use of this property, such as the well-known conjugate gradient (CG) method. However, the coefficient matrices that arise from linearization of load flow problems of energy systems are generally not symmetric. This class of methods will thus not be discussed in this report, but are described in detail in [22].

Generalized Minimum Residual Method (GMRES)

The *generalized minimum residual (GMRES) method* is a Krylov subspace method commonly used in practice for solving linear systems of equations [26]. Is it a projection method, taking $\mathcal{K} = \mathcal{K}_k$, the Krylov subspace, and $\mathcal{L} = A\mathcal{K}_k$. The iterations are based on Arnoldi's orthogonalization method. The procedure for computing the k-th iterate with GMRES is given in Algorithm 2. Each iteration of GMRES consists of the following steps:

1. Construct an orthonormal basis of the Krylov subspace $\mathcal{K}_k$ using Arnoldi's method by finding a vector $\boldsymbol{v}_k$ orthogonal to all vectors in V_{k-1}, an orthonormal basis of $\mathcal{K}_{k-1}$ (lines 3-14 in Algorithm 2).
2. Find the vector $\boldsymbol{y}_k$ that minimizes $||\boldsymbol{r}_k||_2$, the k-th residual (line 16 in Algorithm 2).
3. Compute iterate $\boldsymbol{x}_k = \boldsymbol{x}_0 + V_k\boldsymbol{y}_k$ (line 17 in Algorithm 2).

The second step of the GMRES iteration is to determine the vector $\boldsymbol{y}_k$ that minimizes the k-th residual. During this step, the Hessenberg matrix computed with the Arnoldi orthogonalization is used to define a minimization problem equivalent to minimizing the residual, but is less costly to compute. Using the Hessenberg matrix and Arnoldi relation (4.5)

$$\begin{aligned} \boldsymbol{f} - A\boldsymbol{x} &= \boldsymbol{f} - A(\boldsymbol{x}_0 + V_k\boldsymbol{y}), \\ &= \boldsymbol{r}_0 - AV_k\boldsymbol{y}, \\ &= \beta\boldsymbol{v}_1 - V_{k+1}\bar{H}_k\boldsymbol{y}, \\ &= V_{k+1}\left(\beta\boldsymbol{e}_1 - \bar{H}_k\boldsymbol{y}\right). \end{aligned} \tag{4.7}$$

Since V_{k+1} is an orthonormal basis, it follows that computing the $\boldsymbol{x}_{k+1}$ that minimizes $||\boldsymbol{f} - A\boldsymbol{x}||_2$ is equivalent to finding the $\boldsymbol{y}_k$ that minimizes $||\beta\boldsymbol{e}_1 - \bar{H}_k\boldsymbol{y}||_2$ and computing $\boldsymbol{x}_{k+1} = \boldsymbol{x}_0 + V_{k+1}\boldsymbol{y}_{k+1}$.

The GMRES method is a suitable method for solving large power systems as it is a robust method for non-symmetric coefficient matrices A that is well-suited for preconditioning.

Bi-conjugate Gradient method (Bi-CG)

The *bi-conjugate gradient (Bi-CG)* method is a Krylov subspace method used for solving linear systems of equations. The iterations are based on Lanczos' bi-orthogonalization method. The algorithm therefore not only solves the system $A\boldsymbol{x} = \boldsymbol{f}$, but also the dual system $A^T\boldsymbol{x}^* = \boldsymbol{f}^*$. It is a projection method, taking $\mathcal{K} = \mathcal{K}_k(A, \boldsymbol{v}_0)$, the Krylov subspace, and $\mathcal{L} = \mathcal{K}_k(A^T, \boldsymbol{w}_0)$. The procedure for computing the k-th iterate with Bi-CG is given in Algorithm 3. The algorithm makes use of the bi-orthogonality

Algorithm 2 Generalized minimum residual method (GMRES) algorithm

Choose initial guess $\boldsymbol{x}_0$
$\boldsymbol{r}_0 = \boldsymbol{f} - A\boldsymbol{x}_0$, $\beta = ||\boldsymbol{r}_0||_2$, $\boldsymbol{v}_1 = \frac{\boldsymbol{r}_0}{\beta}$
for $j = 1, 2, \ldots, k$ **do**
    $\boldsymbol{w}_j = A\boldsymbol{v}_j$
    for $i = 1, 2, \ldots, j$ **do**
        $h_{ij} = (\boldsymbol{w}_j, \boldsymbol{v}_j)$
        $\boldsymbol{w}_j = \boldsymbol{w}_j - h_{ij}\boldsymbol{v}_j$
    end for
    $h_{j+1,i} = ||\boldsymbol{w}_j||_2$
    if $h_{j+1,i} = 0$ **then**
        $k = j$
    end if
    $\boldsymbol{v}_{j+1} = \frac{\boldsymbol{w}_j}{h_{j+1,1}}$
end for
$\bar{H}_k = \{h_{ij}\}_{1 \leq i \leq k+1, 1 \leq j \leq k}$
$\boldsymbol{y}_k = \text{argmin}_{\ y} ||\beta \boldsymbol{e}_1 - \bar{H}_k \boldsymbol{y}||_2$
$\boldsymbol{x}_k = \boldsymbol{x}_0 + V_k \boldsymbol{y}_k$

properties in computing the next iterate. The relation in (4.7) also holds here and therefore the k-th iterate is computed as

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + V_k \boldsymbol{y}_k = \boldsymbol{x}_0 + V_k T_k^{-1}(\beta \boldsymbol{e}_1), \tag{4.8}$$

where T_k is tri-diagonal. Denote the LU factorization of T_k as $T_k = L_k U_k$. A substitution of this LU factorization into (4.8) leads to

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + P_k \boldsymbol{z}_k, \tag{4.9}$$

where $P_k = V_k U_k^{-1}$ and $\boldsymbol{z}_k = L_k^{-1}(\beta \boldsymbol{e}_1)$. Due to the short recurrences from the bi-orthogonality property, the matrix P_k is structured as

$$P_k = [P_{k-1}, \boldsymbol{p}_k],$$

where column $\boldsymbol{p}_k$ can be computed from the previous $\boldsymbol{p}_{k-1}$ and $\boldsymbol{v}_k$. Similarly, the vector $\boldsymbol{z}_k$ is structured as

$$\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{z}_{k-1} \\ \zeta_k \end{bmatrix},$$

where entry ζ_k can be computed from ζ_{k-1}. As a result, (4.9) is equivalent to

$$\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \zeta_k \boldsymbol{p}_k,$$

so the next iterate $\boldsymbol{x}_k$ can be computed with the previous iterate.

From the derivations above and the bi-orthogonality we thus have the original and dual identities

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \quad \boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k, \quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k, \\ \boldsymbol{x}^*_{k+1} &= \boldsymbol{x}^*_k + \alpha^*_k \boldsymbol{p}^*_k, \quad \boldsymbol{r}^*_{k+1} = \boldsymbol{r}^*_k - \alpha^*_k A\boldsymbol{p}^*_k, \quad \boldsymbol{p}^*_{k+1} = \boldsymbol{r}^*_{k+1} + \beta^*_k \boldsymbol{p}^*_k. \end{aligned} \tag{4.10}$$

Using the bi-orthogonality properties and substitution it is determined that the constants α and β are given as

$$\alpha_k = \alpha^*_k = \frac{(\boldsymbol{r}_k, \boldsymbol{r}^*_k)}{(A\boldsymbol{p}_k, \boldsymbol{p}^*_k)}, \quad \beta_k = \beta^*_k = \frac{(\boldsymbol{r}_{k+1}, \boldsymbol{r}^*_{k+1})}{(\boldsymbol{r}_k, \boldsymbol{r}^*_k)}.$$

In Algorithm 3 these coefficients are computed and used to update the previous iterate. The solution of the dual problem $\boldsymbol{x}^*_k$ is not explicitly computed in Algorithm 3 since it is not directly needed in the computation of $\boldsymbol{x}_k$. However, if one would want to simultaneously solve the dual problem, only the line $\boldsymbol{x}^*_{k+1} = \boldsymbol{x}^*_k + \alpha_k \boldsymbol{p}^*_k$ has to be added.

Algorithm 3 Bi-conjugate gradient method (Bi-CG) algorithm

Choose initial guess $\boldsymbol{x}_0$
$\boldsymbol{r}_0 = \boldsymbol{f} - A\boldsymbol{x}_0$, choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0, \boldsymbol{r}_0^*) \neq 0$
$\boldsymbol{p}_0 = \boldsymbol{r}_0, \boldsymbol{p}_0^* = \boldsymbol{r}_0^*$
for $j = 1, 2, \ldots, k$ **do**
    $\alpha_j = \frac{(\boldsymbol{r}_j, \boldsymbol{r}_j^*)}{(A\boldsymbol{p}_j, \boldsymbol{p}_j^*)}$
    $\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \alpha_j \boldsymbol{p}_j$
    $\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - \alpha_j A\boldsymbol{p}_j$
    $\boldsymbol{r}_{j+1}^* = \boldsymbol{r}_j^* - \alpha_j A^T \boldsymbol{p}_j^*$
    $\beta_j = \frac{(\boldsymbol{r}_{j+1}, \boldsymbol{r}_{j+1}^*)}{(\boldsymbol{r}_j, \boldsymbol{r}_j^*)}$
    $\boldsymbol{p}_{j+1} = \boldsymbol{r}_{j+1} + \beta_j \boldsymbol{p}_j$
    $\boldsymbol{p}_{j+1}^* = \boldsymbol{r}_{j+1}^* + \beta_j \boldsymbol{p}_j^*$
end for

Bi-conjugate Gradient stabilized method (Bi-CGSTAB)

In the Bi-CG method, the coefficient matrix transpose A^T occurs in the computation of the coefficients α_k and β_k. However, in some applications this transpose is not available, motivating to consider transpose-free methods. Additionally, these methods can converge faster for roughly the same computational costs. The bi-conjugate gradient stabilized (Bi-CGSTAB) method can be found in Algorithm 4 and is an iterative, transpose-free Krylov method that stabilizes the convergence behaviour seen in Bi-CG [27]. In Bi-CGSTAB, iterates are produced whose residual vectors are of the form

$$\boldsymbol{r}_k = \Upsilon_k(A)\phi(A)\boldsymbol{r}_0,$$

where $\phi_k(t)$ is the Bi-CG residual polynomial and $\Upsilon_k(t)$ the newly introduced stabilization polynomial. This stabilization polynomial aims to smooth the convergence behaviour of the Bi-CG algorithm and is defined recursively,

$$\Upsilon_k(t) = (1 - \omega_k t)\Upsilon_{k-1}(t).$$

Adding this stabilization polynomial to the identities of the original problem in Bi-CG as in (4.10), one gets the following residual and conjugate updates

$$\boldsymbol{r}_{k+1} = (I - \omega_k A)(\boldsymbol{r}_k - \alpha_k A\boldsymbol{p}_k), \quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k(I - \omega_k A)\boldsymbol{p}_k.$$

The computation of the coefficients α_k and β_k are rewritten algebraically such that direct application of A^T is not needed. The full derivation is given in [27] and in Chapter 7 of [22]. The parameter ω_k that occurs in the stabilization polynomial is chosen such that a steepest descent step is achieved in the residual direction.

The bi-conjugate gradient stabilized (Bi-CGSTAB) method has faster iteration times than GMRES and uses less memory. However, it is less robust than GMRES, especially when the (non-symmetric) coefficient matrix is ill-conditioned.

4.3 Preconditioning

Preconditioning is a technique used to improve the efficiency and robustness of iterative solution methods. The principle is based on transforming the original linear system into a system that is easier to solve, but has the same solution vector. The quality of the preconditioner that is used is influential on the performance of the iterative method. Preconditioning is done by incorporating a non-singular preconditioning matrix M, such that the linear system $M\boldsymbol{x} = \boldsymbol{f}$ is cheap and easy to solve. This section first discusses how iterative solution methods can be preconditioned. Next, different types of preconditioners are highlighted.

Algorithm 4 Bi-conjugate gradient stabilized method (Bi-CGSTAB) algorithm

Choose initial guess $\boldsymbol{x}_0$
$\boldsymbol{r}_0 = \boldsymbol{f} - A\boldsymbol{x}_0$, choose $\boldsymbol{r}_0^*$ such that $(\boldsymbol{r}_0, \boldsymbol{r}_0^*) \neq 0$
$\boldsymbol{p}_0 = \boldsymbol{r}_0$,
for $j = 1, 2, \ldots, k$ **do**
    $\alpha_j = \frac{(\boldsymbol{r}_j, \boldsymbol{r}_0^*)}{(A\boldsymbol{p}_j, \boldsymbol{r}_0^*)}$
    $\boldsymbol{s}_j = \boldsymbol{r}_j - \alpha_j A\boldsymbol{p}_j$
    $w_j = \frac{(A\boldsymbol{s}_j, \boldsymbol{s}_j)}{(A\boldsymbol{s}_j, A\boldsymbol{s}_j)}$
    $\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \alpha_j \boldsymbol{p}_j + \omega_j \boldsymbol{s}_j$
    $\boldsymbol{r}_{j+1} = \boldsymbol{s}_j - \omega_j A\boldsymbol{s}_j$
    $\beta_j = \frac{(\boldsymbol{r}_{j+1}, \boldsymbol{r}_0^*)}{(\boldsymbol{r}_j, \boldsymbol{r}_0^*)} \times \frac{\alpha_j}{\omega_j}$
    $\boldsymbol{p}_{j+1} = \boldsymbol{r}_{j+1} + \beta_j(\boldsymbol{p}_j - \omega_j A\boldsymbol{p}_j)$
end for

4.3.1 Preconditioning Methods

A preconditioner can be applied from the *left*, resulting in the preconditioned system

$$M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{f}. \tag{4.11}$$

Alternatively, the preconditioner can be applied to the *right*, leading to system

$$AM^{-1}\boldsymbol{u} = \boldsymbol{f}, \ \boldsymbol{x} = M^{-1}\boldsymbol{u}. \tag{4.12}$$

When the preconditioning matrix can be factorized as $M = M_1 M_2$, one can apply *split* preconditioning

$$M_1^{-1}AM_2^{-1}\boldsymbol{u} = M_1^{-1}\boldsymbol{f}, \ \boldsymbol{x} = M_1^{-1}\boldsymbol{u}. \tag{4.13}$$

Note that taking $M_1^{-1} = I$ leads to right preconditioning as in (4.12), while $M_2^{-1} = I$ leads to left preconditioning as in (4.11).

Preconditioned GMRES

In the left-preconditioned GMRES algorithm, Arnoldi's orthogonalization method constructs an orthogonal basis of the left-preconditioned Krylov subspace

$$\mathcal{K}_k(M^{-1}A, M^{-1}\boldsymbol{r}_0) = \operatorname{span}\left\{M^{-1}\boldsymbol{r}_0, M^{-1}AM^{-1}\boldsymbol{r}_0, \ldots, (M^{-1}A)^{k-1}M^{-1}\boldsymbol{r}_0\right\}, \tag{4.14}$$

resulting from linear system (4.11). The algorithm of the left-preconditioned GMRES method is similar to the un-preconditioned GMRES as in Algorithm 2, but all actions of coefficient matrix A are replaced with with $M^{-1}A$. A downside of the left-preconditioned GMRES method is that the un-preconditioned residuals, $\boldsymbol{r}_k$ are not explicitly computed. It is therefore difficult to define a stopping criterium based on these residuals unless they are explicitly computed.

In the right-preconditioned GMRES algorithm, Arnoldi's orthogonalization method constructs an orthogonal basis of the right-preconditioned subspace

$$\mathcal{K}_k(AM^{-1}, \boldsymbol{r}_0) = \operatorname{span}\left\{\boldsymbol{r}_0, AM^{-1}\boldsymbol{r}_0, \ldots, (AM^{-1})^{k-1}\boldsymbol{r}_0\right\}, \tag{4.15}$$

based in linear system (4.12). The algorithm of the right-reconditioned GMRES method is similar to the un-preconditioned GMRES as in Algorithm 2, but all actions of coefficient matrix A are replaced with AM^{-1} and the iterate solution computation is replaced with $\boldsymbol{x}_k = \boldsymbol{x}_0 + M^{-1}V_k\boldsymbol{y}_k$. The transformed variable $\boldsymbol{u}$ is not directly computed.

The preconditioned algorithms for GMRES can be found in Chapter 9 of [22]. The difference in convergence behaviour of the left- and right-preconditioned GMRES is not significant, unless the preconditioner M is ill-conditioned [22].

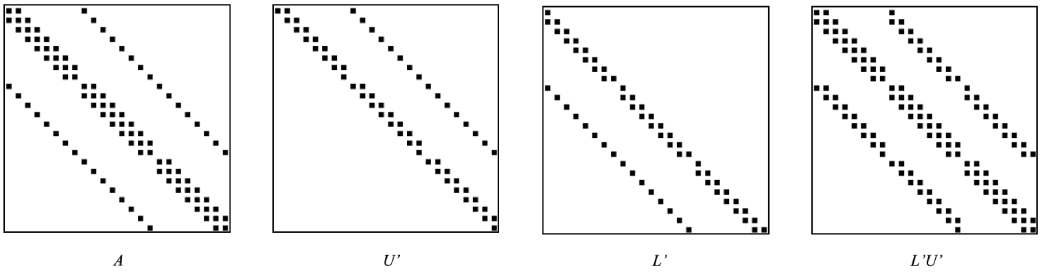


Figure 4.1: Taken from [28]: 'ILU(0) Factorization.'

Preconditioned Bi-CGSTAB

In the left-preconditioned Bi-CGSTAB algorithm an orthonormal basis is constructed of the same left-preconditioned subspace as for the GMRES method, (4.14). The left-preconditioned algorithm is similar to Algorithm 4, where all actions of A are replaced by $M^{-1}A$.

The Krylov subspace for the right-preconditioned Bi-CGSTAB method is given in (4.15). In Algorithm 4 all actions of A are replaced by AM^{-1} and the update of $\boldsymbol{x}$ is multiplied from the left with M^{-1}. The transformed variable $\boldsymbol{u}$ does not have to be computed directly.

The fully preconditioned algorithms for Bi-CGSTAB can be found in [27].

4.3.2 Preconditioning Techniques

Finding a good preconditioner is an important step of applying iterative solvers such as Krylov subspace methods. Despite general results and guidelines, there are no fixed detailed theoretical results on how a preconditioner should be constructed and what their effect is on the solver performance. However, a general principle is that the preconditioning matrices are derived from the original coefficient matrix and the action of the preconditioner should be inexpensive to apply to a vector.

Incomplete LU factorization

An intuitive way of defining a preconditioner is by an *incomplete LU (ILU) factorization* of the original coefficient matrix,

$$A = \tilde{L}\tilde{U} - R,$$

where R is the residual matrix of the factorization. For a sparse matrix A, the matrices L and U often become more dense during the factorization process. The term *fill-in* refers to elements of the upper triangular part of U and lower triangular part of L that are non-zero, that were originally zero in the coefficient matrix A. This is an undesirable phenomenon since sparse matrices are less computationally expensive to work with.

There are different variations of ILU methods and preconditioners. Most ILU preconditioning methods aim at reducing the number of fill-ins so that the preconditioner is inexpensive to apply. How one determines which fill-ins to tackle differs per method and can be done in various ways. Most commonly either the location of the fill-in or its magnitude is used. Below, several commonly used ILU preconditioning techniques will be discussed. A more comprehensive list can be found in Section 10.3 of [22].

The zero fill-in ILU factorization technique, denoted by ILU(0), takes the zero pattern of the incomplete factorization to be precisely the zero pattern of A. No fill-in is thus allowed, which is what the '0' indicates. The zero pattern of the factorization refers to the zero pattern of the lower triangular part of L and the upper triangular part of U. Note that the product LU can therefore have a different zero pattern than A, see Figure 4.1.

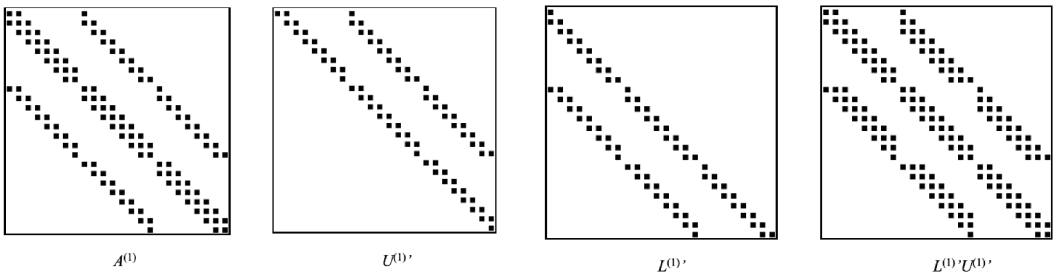


Figure 4.2: Taken from [28]: 'ILU(1) Factorization.'

In the ILU(p) factorization technique, all fill-in elements of level p or smaller are kept during the factorization process. For example, in ILU(1) only first-level fill-in elements are allowed. The rationale of the fill level is that the level of an element should be indicative of the size, such that a higher level corresponds to smaller elements. The initial level of fill of an element a_{ij} of sparse matrix A is defined by

$$\text{lev}(a_{ij}) = \begin{cases} 0, & \text{if } a_{ij} \neq 0 \text{ or } i = j, \\ \infty, & \text{otherwise.} \end{cases}$$

During the Gaussian elimination process (the LU factorization) the fill-level is updated through

$$\text{lev}(a_{ij}) = \min\left\{\text{lev}(a_{ij}),\ \text{lev}(a_{ik}) + \text{lev}(a_{kj}) + 1\right\}$$

According to the above definition, the fill-level of an element does not increase during the elimination process. The zero pattern of ILU(p) may thus be written as

$$P_p = \left\{(i,j) : \text{lev}(a_{ij}) \leq p\right\}.$$

An example of the zero patterns of the ILU(1) factorization is given in Figure 4.2, based on the same coefficient matrix A as the zero pattern of the ILU(0) factorization from Figure 4.1.

Another commonly used ILU factorization uses a *dropping tolerance* in determining which fill-in elements to keep or discard. This can be done in various ways, but the general aim of applying a dropping rule to an element is to replace the element with a zero if it satisfies a set of criteria. In the ILUT approach, where 'T' stands for 'threshold', denoted by ILUT(p,τ), two general dropping criteria can be considered. First, during the Gaussian elimination process an element is dropped if it is less than the relative tolerance τ_i obtained by multiplying τ by the original norm of the i-th row. The goal of this criterion is to drop all elements that are of small magnitude. As the second dropping criteria, keep only the p largest elements in the row in addition to the diagonal element (which is always kept). The goal of this dropping rule is to control the number of elements per row, which can help reduce memory usage.

The ILUT approach can be combined with pivoting of the original coefficient matrix, leading to the ILUTP approach. Using pivoting is beneficial when the ILUT method encounters a zero pivot, over- or underflow or results in an unstable preconditioner. At the i-th step of the elimination process the largest entry of the row is selected to be the new i-th variable. The corresponding permutation matrices are updated accordingly.

Target matrix

An ILU preconditioner can be based on different matrices related to the coefficient matrix A. In the i-th NR iteration, the coefficient matrix is given by the Jacobian J_i. In [29] the performance of ILU methods based on the Jacobian J_i, initial Jacobian J_0 and a Jacobian simplification known as *fast decoupled load flow (FDLF)* were compared. More about the conclusions of this research can be read in Section 5.3.

4.4 Non-Linear Systems

The systems of equations that describe steady-state load flow of electricity and gas networks, and thus multi-carrier networks, are non-linear. A commonly used principle in solving these non-linear equations is to linearize and use a direct or iterative linear solver, as described in Sections 4.1 and 4.2. This section introduces the classic NR as well as inexact NR method. Additionally, the general Jacobian matrices of electricity, gas and coupled networks are derived.

4.4.1 Newton-Raphson

The *NR* method is used to iteratively solve non-linear systems of the form

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{0}.$$

The *Jacobian matrix* J of $\boldsymbol{F}(\boldsymbol{x})$, $\boldsymbol{F} : \mathbb{R}^N \to \mathbb{R}^N$, is given by

$$J_F = \nabla \boldsymbol{F} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_N}{\partial x_1} & \cdots & \frac{\partial F_N}{\partial x_N} \end{bmatrix}.$$

The NR method is based on a first-order multivariate Taylor expansion around iterate $\boldsymbol{x}_k$, such that for the next iterate $\boldsymbol{x}_{k+1}$ we have

$$\boldsymbol{F}(\boldsymbol{x}_{k+1}) = \boldsymbol{F}(\boldsymbol{x}_k) + J_F(\boldsymbol{x}_k)(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \text{h.o.t.}$$

The aim is to find $\boldsymbol{x}_{k+1}$ such that $\boldsymbol{F}(\boldsymbol{x}_{k+1})$ gets close to zero. The above equation therefore becomes

$$J_F(\boldsymbol{x}_k)\boldsymbol{s}_k \approx -\boldsymbol{F}(\boldsymbol{x}_k),$$

where $\boldsymbol{s}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$, referred to as the update. The NR update equation is thus given as

$$J_F(\boldsymbol{x}_k)\boldsymbol{s}_k = -\boldsymbol{F}(\boldsymbol{x}_k), \tag{4.16}$$

During each iteration, the linear system is solved using a linear solution method (direct or iterative), after which the new solution approximation $\boldsymbol{x}_{k+1}$ is determined using the computed update, $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{s}_k$.

4.4.2 Inexact Newton-Raphson

The class of *Inexact Newton* methods compute an approximate solution to the Newton equations (4.16) in some unspecified manner such that

$$\frac{||\boldsymbol{r}_k||}{||\boldsymbol{F}(\boldsymbol{x}_k)||} \leq \eta_k. \tag{4.17}$$

Here $\boldsymbol{r}_k = J_F(\boldsymbol{x}_k)\boldsymbol{s}_k + \boldsymbol{F}(\boldsymbol{x}_k)$ is the residual, $||\boldsymbol{r}_k||/||\boldsymbol{F}(\boldsymbol{x}_k)||$ the relative residual and η_k a non-negative *forcing sequence* used to control the level of accuracy [30, 31, 29]. Under the assumption that the forcing sequence is uniformly less than one, all such inexact Newton methods are locally convergent and exhibit the same quadratic convergence as the NR method [30].

4.4.3 Jacobian Matrices of Energy Networks

The Jacobian of a (non)linear system of equations is fundamental in the application of NR. It is insightful to investigate the general form of the Jacobian matrices of energy networks, since their spectral properties influence the convergence and accuracy of NR. This section gives the general form of the Jacobian matrix of electricity, gas and coupled networks, using the load flow equations of Subsection 2.4.1, 2.4.2 and 3.2, respectively.

Jacobian matrix electricity network

The Jacobian matrix of an electricity network based on steady-state load flow (2.30) is given as

$$J_{F^e} = \begin{pmatrix} \frac{\partial \boldsymbol{F}^P}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{F}^P}{\partial |\boldsymbol{V}|} \\ \frac{\partial \boldsymbol{F}^Q}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{F}^Q}{\partial |\boldsymbol{V}|} \end{pmatrix}. \tag{4.18}$$

In (2.28), note that the load flow equations contain the components

$$\mathcal{P}_{ij} = |V_i||V_j| \left(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}\right), \quad \mathcal{Q}_{ij} = -|V_i||V_j| \left(B_{ij} \cos \delta_{ij} - G_{ij} \sin \delta_{ij}\right).$$

Therefore we can write the non-linear steady-state load flow equations for an electricity network as

$$\begin{aligned} \boldsymbol{F}_i^P(\boldsymbol{x}^e) &= P_i + \sum_j \mathcal{P}_{ij}(\boldsymbol{x}^e) = P_i + |V_i|^2 G_{ii} + \sum_{j,j\neq i} \mathcal{P}_{ij}(\boldsymbol{x}^e) = 0, \\ \boldsymbol{F}_i^Q(\boldsymbol{x}^e) &= Q_i + \sum_j \mathcal{Q}_{ij}(\boldsymbol{x}^e) = Q_i - |V_i|^2 B_{ii} - \sum_{j,j\neq i} \mathcal{Q}_{ij}(\boldsymbol{x}^e) = 0. \end{aligned}$$

The Jacobian (4.18) then contains the components

$$\begin{aligned} \frac{\partial \boldsymbol{F}_i^P}{\partial \delta_j} &= -|V_i||V_j| \left(B_{ij} \cos \delta_{ij} - G_{ij} \sin \delta_{ij}\right) && = \mathcal{Q}_{ij}, \\ \frac{\partial \boldsymbol{F}_i^P}{\partial \delta_i} &= \sum_{j,j\neq i} |V_i||V_j| \left(B_{ij} \cos \delta_{ij} - G_{ij} \sin \delta_{ij}\right) && = -\sum_{j,j\neq i} \mathcal{Q}_{ij} = -Q_i - |V_i|^2 B_{ii}, \\ \frac{\partial \boldsymbol{F}_i^Q}{\partial \delta_j} &= -|V_i||V_j| \left(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}\right) && = -\mathcal{P}_{ij}, \\ \frac{\partial \boldsymbol{F}_i^Q}{\partial \delta_i} &= \sum_{j,j\neq i} |V_i||V_j| \left(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}\right) && = \sum_{j,j\neq i} \mathcal{P}_{ij} = P_i - |V_i|^2 G_{ii}, \\ \frac{\partial \boldsymbol{F}_i^P}{\partial |V_j|} &= |V_i| \left(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}\right) && = \frac{\mathcal{P}_{ij}}{|V_j|}, \\ \frac{\partial \boldsymbol{F}_i^P}{\partial |V_i|} &= 2|V_i| G_{ii} + \sum_{j,j\neq i} |V_j| \left(G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}\right) && = \frac{\sum_{j,j\neq i} \mathcal{P}_{ij}}{|V_i|} + 2|V_i| G_{ii} = \frac{P_i + |V_i|^2 G_{ii}}{|V_i|}, \\ \frac{\partial \boldsymbol{F}_i^Q}{\partial |V_j|} &= |V_i| \left(B_{ij} \cos \delta_{ij} - G_{ij} \sin \delta_{ij}\right) && = \frac{\mathcal{Q}_{ij}}{|V_j|}, \\ \frac{\partial \boldsymbol{F}_i^Q}{\partial |V_i|} &= -2|V_i| B_{ii} - \sum_{j,j\neq i} |V_j| \left(B_{ij} \cos \delta_{ij} - G_{ij} \sin \delta_{ij}\right) && = \frac{\sum_{j,j\neq i} \mathcal{Q}_{ij}}{|V_i|} - 2|V_i| B_{ii} = \frac{Q_i - |V_i|^2 B_{ii}}{|V_i|}. \end{aligned}$$

The matrices $\frac{\partial \boldsymbol{F}^P}{\partial \boldsymbol{\delta}}$ and $\frac{\partial \boldsymbol{F}^Q}{\partial \boldsymbol{\delta}}$ have the property that the diagonal element is the negative value of the sum of the other elements in the row. The Jacobian is a not-symmetric matrix.

Jacobian matrix gas network

The Jacobian of a gas network based on steady-state load flow (2.35) is given as

$$J_{F^g} = \begin{pmatrix} \frac{\partial \boldsymbol{F}^q}{\partial \boldsymbol{q}} & \frac{\partial \boldsymbol{F}^q}{\partial \boldsymbol{p}} \\ \frac{\partial \boldsymbol{F}^L}{\partial \boldsymbol{q}} & \frac{\partial \boldsymbol{F}^L}{\partial \boldsymbol{p}} \end{pmatrix}, \tag{4.19}$$

where conservation of mass system $\boldsymbol{F}^q$ is given by (2.37) and $\boldsymbol{F}_k^L$ is given by gas pipe link equation (2.38). The components of the Jacobian are given as

$$\begin{aligned}
\frac{\partial \boldsymbol{F}^q}{\partial \boldsymbol{q}} &= \boldsymbol{A}^{\boldsymbol{g}\prime}, \\
\frac{\partial \boldsymbol{F}^q}{\partial \boldsymbol{p}} &= \boldsymbol{0}, \\
\frac{\partial \boldsymbol{F}_k^L}{\partial q_k} &= -2(C^g)^{-2}|q_k|\left(f_k + \frac{1}{2}q_k\frac{\partial f_k}{\partial q_k}\right), \\
\frac{\partial \boldsymbol{F}_k^L}{\partial q_l} &= 0, \\
\frac{\partial \boldsymbol{F}_k^L}{\partial p_i} &= \begin{cases} 1, & \text{low-pressure,} \\ 2p_i, & \text{high-pressure.} \end{cases} \\
\frac{\partial \boldsymbol{F}_k^L}{\partial p_j} &= \begin{cases} -1, & \text{low-pressure,} \\ -2p_j, & \text{high-pressure,} \end{cases}
\end{aligned}$$

where link q_k represents gas flow from nodes v_i to v_j. It follows that $\frac{\partial \boldsymbol{F}^L}{\partial \boldsymbol{q}}$ is a diagonal matrix. If the pipe friction factor f_k is independent of the gas pipe flow, then $\frac{\partial f_k}{\partial q_k} = 0$. Note that in a low-pressure gas network we have

$$\frac{\partial \boldsymbol{F}^L}{\partial \boldsymbol{p}} = \boldsymbol{A}^{\boldsymbol{g}\prime T},$$

the transpose of the reduced incidence matrix. The general matrix structure of the Jacobian of a low-pressure gas network is

$$J_{F^g} = \begin{pmatrix} \boldsymbol{A}^{\boldsymbol{g}\prime} & \boldsymbol{0} \\ \operatorname{diag}\left[-2(C^g)^{-2}|\boldsymbol{q}|\left(\boldsymbol{f} + \frac{1}{2}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}\boldsymbol{q}\right)\right] & \boldsymbol{A}^{\boldsymbol{g}\prime T} \end{pmatrix}.$$

The Jacobian matrix of a gas network is thus a (non-symmetric) lower block-triangular matrix.

Jacobian matrix coupled energy network

The non-linear system of equations for a steady-state load flow of a coupled energy network with electricity and gas is given by (3.3). The Jacobian of a coupled electricity and gas network is given as

$$J_{F^c} = \begin{pmatrix} \frac{\partial \boldsymbol{F}^g(\boldsymbol{x}^g,\boldsymbol{x}^c)}{\partial \boldsymbol{x}^g} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}^g(\boldsymbol{x}^g,\boldsymbol{x}^c)}{\partial \boldsymbol{x}^c} \\ \boldsymbol{0} & \frac{\partial \boldsymbol{F}^e(\boldsymbol{x}^e,\boldsymbol{x}^c)}{\partial \boldsymbol{x}^e} & \frac{\partial \boldsymbol{F}^e(\boldsymbol{x}^e,\boldsymbol{x}^c)}{\partial \boldsymbol{x}^c} \\ \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}^c(\boldsymbol{x}^c)}{\partial \boldsymbol{x}^c} \end{pmatrix} = \begin{pmatrix} J_{F^g} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}^g(\boldsymbol{x}^g,\boldsymbol{x}^c)}{\partial \boldsymbol{x}^c} \\ \boldsymbol{0} & J_{F^e} & \frac{\partial \boldsymbol{F}^e(\boldsymbol{x}^e,\boldsymbol{x}^c)}{\partial \boldsymbol{x}^c} \\ \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}^c(\boldsymbol{x}^c)}{\partial \boldsymbol{x}^c} \end{pmatrix}. \tag{4.20}$$

The coupling parameters $\boldsymbol{x}_c$ consist of $\boldsymbol{q}_c$, $\boldsymbol{P}_c$ and $\boldsymbol{Q}_c$, as in (3.2). In the gas component $\boldsymbol{F}^g$, only the conservation of mass equations $\boldsymbol{F}_q^g$ can depend on $\boldsymbol{q}_c$. The matrix $\frac{\partial \boldsymbol{F}_q^g}{\partial \boldsymbol{q}^c}$ is thus a slice of the reduced incidence matrix of the coupled gas network of the columns corresponding to the dummy links. Denote this incidence matrix by A_c^q.

$$\frac{\partial \boldsymbol{F}^g}{\partial \boldsymbol{x}_c} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_q^g}{\partial \boldsymbol{q}_c} & \frac{\partial \boldsymbol{F}_q^g}{\partial \boldsymbol{P}_c} & \frac{\partial \boldsymbol{F}_q^g}{\partial \boldsymbol{Q}_c} \\ \frac{\partial \boldsymbol{F}_L^g}{\partial \boldsymbol{q}_c} & \frac{\partial \boldsymbol{F}_L^g}{\partial \boldsymbol{P}_c} & \frac{\partial \boldsymbol{F}_L^g}{\partial \boldsymbol{Q}_c} \end{pmatrix} = \begin{pmatrix} A_c^q & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{pmatrix}$$

A similar observation can be done for the energy network components. The coupling parameters $\boldsymbol{P}_c$ only occur in $\boldsymbol{F}_P^e$, and $\boldsymbol{Q}_c$ only in $\boldsymbol{F}_Q^e$. The matrices $\frac{\partial \boldsymbol{F}_P^e}{\partial \boldsymbol{P}^c}$ and $\frac{\partial \boldsymbol{F}_Q^e}{\partial \boldsymbol{Q}^c}$ are slices of the incidence matrix of the coupled electrical network of the columns corresponding to the dummy links. Denote these matrices by A_c^P and A_c^Q, respectively. Since a dummy link connects to only one node in a single-carrier network,

the matrices A_c^q, A_c^P and A_c^Q are square with the number of rows/columns equal to the number of connected coupling units.

$$\frac{\partial \boldsymbol{F}^e}{\partial \boldsymbol{x}_c} = \begin{pmatrix} \frac{\partial \boldsymbol{F}_{\boldsymbol{P}}^e}{\partial \boldsymbol{q}_c} & \frac{\partial \boldsymbol{F}_{\boldsymbol{P}}^e}{\partial \boldsymbol{P}_c} & \frac{\partial \boldsymbol{F}_{\boldsymbol{P}}^e}{\partial \boldsymbol{Q}_c} \\ \frac{\partial \boldsymbol{F}_{\boldsymbol{Q}}^e}{\partial \boldsymbol{q}_c} & \frac{\partial \boldsymbol{F}_{\boldsymbol{Q}}^e}{\partial \boldsymbol{P}_c} & \frac{\partial \boldsymbol{F}_{\boldsymbol{Q}}^e}{\partial \boldsymbol{Q}_c} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} & A_c^P & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & A_c^Q \end{pmatrix}$$

The components of matrix $\frac{\partial \boldsymbol{F}^c}{\partial \boldsymbol{x}^c}$ depend on the equations used to model the coupling units.

The non-zero blocks in (4.20) are generally not square, due to the occurrence of new node types in coupled networks, see Subsection 3.2.2. Actually, if the Jacobian of a coupled network is upper block-triangular with square sub-matrices on the diagonal, the single-carrier parts of the network are not interdependent on each other [6].

4.5 Domain Decomposition Methods

The class of *domain decomposition (DD)* methods consists of methods where a problem on a global domain is split into several problems on smaller domains, and iterating to determine the global solution. These methods revolve around the principle of 'divide-and-conquer', where a large problem is split into smaller sub-problems of the same type, until the sub-problems become simple enough to solve. Domain decomposition techniques can be advantageous, since sub-problems might be easier to solve, but also for their natural parallelizability. Saad [22] mentions that DD techniques are distinguished by four features: Type of partitioning, overlap between subdomains, processing of interface values, and subdomain solution methods and qualities.

4.5.1 Schur Complement Approaches

Assume that the linear system $A\boldsymbol{x} = \boldsymbol{f}$ can be written in block-form

$$\begin{bmatrix} B & E \\ F & C \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{g} \end{bmatrix}, \quad \text{with } A = \begin{bmatrix} B & E \\ F & C \end{bmatrix}, \ \boldsymbol{x} = \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix}, \ \boldsymbol{f} = \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{g} \end{bmatrix},$$

such that B is non-singular. The reduced system of $A\boldsymbol{x} = \boldsymbol{f}$ is given by

$$\left(C - FB^{-1}E\right)\boldsymbol{z} = \boldsymbol{g} - FB^{-1}\boldsymbol{b}. \tag{4.21}$$

The matrix

$$S = \left(C - FB^{-1}E\right),$$

is called the *Schur complement* of A. Solving the original system $A\boldsymbol{x} = \boldsymbol{f}$ is then equivalent to solving the reduced system, whose result can be used to compute

$$\boldsymbol{y} = B^{-1}\left(\boldsymbol{b} - E\boldsymbol{z}\right),$$

which leads to the solution vector $\boldsymbol{x}$.

Schur complement methods are based on solving the reduced system (4.21) by a (preconditioned) Krylov subspace method.

4.5.2 Schwarz Alternating Procedures

The methods referred to as Schwarz alternating procedures (SAPs), are based on alternating between overlapping subdomains, solving a new problem on each subdomain and exchanging boundary information. We thus have subdomains Ω_i^δ such that

$$\Omega = \bigcup_{i=1}^{N} \Omega_i^\delta, \quad \Omega_i \subset \Omega_i^\delta,$$

with a δ-level overlap between neighbouring subdomains. An overlap of $\delta = 1$ implies that only the actual boundary of the subdomain is used. There is a distinction between *multiplicative* and *additive* Schwarz procedures, based on how boundary information is exchanged. In the multiplicative Schwarz procedure, the algorithm sweeps through the N subdomains and solves the original problem in each subdomain by using the most recent boundary information (initial guess or solution of a previous subdomain). In the additive Schwarz (AS) procedure, the boundary components in each subdomain are updated after the sweep through all subdomains, not during. The multiplicative Schwarz procedure uses the most recent available information and will thus require less iterations than the AS procedure. However, the subdomain solves in the AS procedure can be done in parallel, which can reduce total computational time to find the global solution. Note that the multiplicative Schwarz procedure is analogous to the block Gauss-Seidel iteration, whereas the AS procedure is analogous to the block Jacobi iteration.

We aim to solve the linear system $A\boldsymbol{x} = \boldsymbol{f}$, on global domain Ω using an iterative Schwarz procedure. Let $\Omega \subseteq \mathbb{R}^n$ and $\Omega_i^\delta \subseteq \mathbb{R}^{n_{i,\delta}}$ for overlapping subdomains Ω_i^δ. Define $R_i^\delta : \Omega \to \Omega_i^\delta$ as the *restriction operator* from global domain Ω to subdomain Ω_i^δ. It keeps only those components of a vector in Ω that are also in Ω_i^δ, and thus R_i^δ is an $n_{i,\delta} \times n$ matrix. Conversely, define the transpose of the restriction operator as a *prolongation operator*, ${R_i^\delta}^T : \Omega_i^\delta \to \Omega$, which takes the variables from a vector with components in Ω_i^δ and extends them to the equivalent components of Ω. The prolongation operator is thus an $n \times n_{i,\delta}$ matrix. Using the restriction and prolongation operators, we can define the associated linear system on a subdomain. Consider

$$A_i = R_i^\delta A {R_i^\delta}^T. \tag{4.22}$$

Here, A_i is an $n_{i,\delta} \times n_{i,\delta}$ matrix, which is the restriction of global matrix A to subdomain Ω_i^δ.

Multiplicative Schwarz preconditioner

In the multiplicative Schwarz procedure, the global solution $\boldsymbol{x}$ is updated after each local solve. Here, the subscript $\boldsymbol{x}_i$ is used to denote the restriction of $\boldsymbol{x}$ to subdomain Ω_i, thus $\boldsymbol{x}_i = {R_i^\delta}^T \boldsymbol{x}$. The superscript $\boldsymbol{x}^i$ denotes the global solution $\boldsymbol{x}$ after updating with the solution of subdomain Ω_i^δ. With $\Delta\boldsymbol{x}_i$, denote the update on the subdomain Ω_i^δ, such that the global update is computed using the prolongation operator ${R_i^\delta}^T \Delta\boldsymbol{x}_i$. The local update is thus denoted by

$$\boldsymbol{x}_i^i = \boldsymbol{x}_i^{i-1} + \Delta\boldsymbol{x}_i, \tag{4.23}$$

while the global update is given by

$$\boldsymbol{x}^i = \boldsymbol{x}^{i-1} + {R_i^\delta}^T \Delta\boldsymbol{x}_i. \tag{4.24}$$

The local update is computed through solving local problem $A_i \boldsymbol{x}_i^i = \boldsymbol{f}_i$. Substitution of (4.23) in the local solve an rewriting results in

$${R_i^\delta}^T \Delta\boldsymbol{x}_i = {R_i^\delta}^T A_i^{-1} R_i^\delta (\boldsymbol{f} - A\boldsymbol{x}^{i-1}).$$

This update is performed for each subdomain Ω_i^δ, and $\boldsymbol{x}_{\text{new}} = \boldsymbol{x}^N$ after all N subdomain updates. The multiplicative Schwarz procedure can be found in Algorithm 5. Note that in the update formula there is an inverse of the restricted coefficient matrix A_i. This does not necessarily imply that the inverse is directly computed, but indicates that the smaller subdomain problem has to be solved. Depending on the problem size and application, the resulting linear system subdomain problem can be solved with a direct or iterative method. The error after each subdomain update is given by $\boldsymbol{e}_i = \boldsymbol{x}^* - \boldsymbol{x}^i$, where $\boldsymbol{x}^*$ is the exact solution. The following relation holds for this error

$$\boldsymbol{e}_i = (I - P_i)\, \boldsymbol{e}_{i-1}, \quad P_i = {R_i^\delta}^T A_i^{-1} R_i^\delta A.$$

This relation indicates that the final error $\boldsymbol{e}_{\text{new}} = \boldsymbol{e}_N$ can be expressed using the P_i and initial error $\boldsymbol{e}_0$,

$$\boldsymbol{e}_{\text{new}} = \boldsymbol{e}_N = (I - P_N)\,(I - P_{N-1}) \dots (I - P_2)\,(I - P_1)\, \boldsymbol{e}_0.$$

Algorithm 5 Multiplicative Schwarz procedure

1: **for** $i = 1, \ldots, N$ **do**
2: $\quad \boldsymbol{x}^i = \boldsymbol{x}^{i-1} + {R_i^\delta}^T A_i^{-1} R_i^\delta (\boldsymbol{f} - A\boldsymbol{x})$
3: **end for**
4: $\boldsymbol{x}_{\text{new}} = \boldsymbol{x}^N$

Denote this product by Q_N such that

$$\boldsymbol{e}_{\text{new}} = Q_N \boldsymbol{e}_0, \quad Q_N = (I - P_N) \ldots (I - P_1),$$

explaining the name of the multiplicative Schwarz method. This implies that the updated solution vector $\boldsymbol{x}_{\text{new}}$ can be computed from $\boldsymbol{x}$ directly,

$$\boldsymbol{x}_{\text{new}} = Q_N \boldsymbol{x} + (I - Q_N)\boldsymbol{x}^*. \tag{4.25}$$

This multiplicative Schwarz procedure update formula is written as a fixed-point iteration. Let a general fixed point iteration be of the form

$$\boldsymbol{x}_{\text{new}} = G\boldsymbol{x} + \boldsymbol{b}. \tag{4.26}$$

For the multiplicative Schwarz update (4.25) we then have

$$G = Q_N, \quad \boldsymbol{b} = (I - Q_N)\boldsymbol{x}^*.$$

Note that a left-preconditioned system $M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{f}$ is also solved using a fixed-point iteration of the form 4.26, where

$$G = I - M^{-1}A, \quad \boldsymbol{b} = M^{-1}\boldsymbol{f}. \tag{4.27}$$

Therefore, the multiplicative Schwarz procedure sweep is equivalent to a fixed-point iteration for the left-preconditioned system in which

$$M^{-1}A = I - Q_N, \quad M^{-1}\boldsymbol{f} = (I - Q_N)\boldsymbol{x}^*,$$

referred to as the multiplicative Schwarz preconditioner. The exact solution $\boldsymbol{x}^*$ is of course unknown and cannot be used directly in computing $\boldsymbol{x}_{\text{new}}$. However, one can operate with M^{-1} iteratively without having to explicitly form A^{-1}, using new definitions and recurrence relations. Define

$$T_i = {R_i^\delta}^T A_i^{-1} R_i^\delta, \quad Z_i = I - Q_i, \quad M_i = Z_i A^{-1}, \tag{4.28}$$

such that $M^{-1} = M_N$. Then the following recurrence relations hold

$$\begin{aligned} &Z_1 = P_1, \quad Z_i = Z_{i-1} + P_i(I - Z_{i-1}), \\ &M_1 = T_1, \quad M_i = M_{i-1} + T_i(I - AM_{i-1}). \end{aligned} \tag{4.29}$$

Using these definitions and recurrence relations, we can compute the action of $M^{-1} = M_N$ on an arbitrary vector $\boldsymbol{v}$ and on $A\boldsymbol{v}$ iteratively, see Algorithm 6. In the algorithm, the contribution of each subdomain solve to the preconditioner application is computed and added to form the final application of preconditioner. Note that the T_i and P_i include local solves with A_i, which can be done using a direct or iterative solver.

Additive Schwarz preconditioner

In the AS procedure, the updates from the local solves in the form of (4.24) are added to the global solution vector after all updates have been computed, leading to Algorithm 7. The update in line 4 of Algorithm 7 is already in a fixed-point form, (4.26), with

$$G = I - \sum_{i=1}^{N} P_i, \quad \boldsymbol{b} = \sum_{i=1}^{N} T_i A^{-1} \boldsymbol{f}.$$

Algorithm 6 Multiplicative Schwarz preconditioner action on $\boldsymbol{v}$

1: Input: $\boldsymbol{v}$
2: $\boldsymbol{z}_1 = T_1\boldsymbol{v}$
3: **for** $i = 2, \ldots, N$ **do**
4: $\quad \boldsymbol{z}_i = \boldsymbol{z}_{i-1} + T_i(\boldsymbol{v} - A\boldsymbol{z}_{i-1}) = \boldsymbol{z}_{i-1} + {R_i^\delta}^T A_i^{-1} R_i^\delta(\boldsymbol{v} - A\boldsymbol{z}_{i-1})$
5: **end for**
6: Output: $M_{MS}^{-1}\boldsymbol{v} = \boldsymbol{z}_N$

Algorithm 7 Additive Schwarz procedure

1: **for** $i = 1, \ldots, N$ **do**
2: $\quad \Delta\boldsymbol{x}_i = {R_i^\delta}^T A_i^{-1} R_i^\delta(\boldsymbol{b} - A\boldsymbol{x})$
3: **end for**
4: $\boldsymbol{x}_{\text{new}} = \boldsymbol{x} + \sum_{i=1}^N \Delta\boldsymbol{x}_i$

The left-preconditioned system is solved using a fixed-point iteration with (4.27). Therefore, the AS procedure sweep is equivalent to a fixed-point iteration for the left-preconditioned system in which

$$M^{-1}A = \sum_{i=1}^N P_i, \quad M^{-1} = \sum_{i=1}^N T_i = \sum_{i=1}^N {R_i^\delta}^T A_i^{-1} R_i^\delta, \tag{4.30}$$

explaining the name of the AS method. Using the same definitions (4.28) as for the multiplicative Schwarz preconditioner, the action of the preconditioner on an arbitrary vector $\boldsymbol{v}$ and $A\boldsymbol{v}$ can be computed iteratively, see Algorithm 8. Similarly to the multiplicative procedures, the contribution of each subdomain solve to the preconditioner application is computed and stored. Once all updates have been computed, the final application of the preconditioner is returned. Note that the computations in line 4 of Algorithm 8 can be done in parallel.

Algorithm 8 Additive Schwarz preconditioner action on $\boldsymbol{v}$

1: Input: $\boldsymbol{v}$
2: **for** $i = 1, \ldots, N$ **do**
3: $\quad \boldsymbol{z}_i = T_i\boldsymbol{v} = {R_i^\delta}^T A_i^{-1} R_i^\delta\boldsymbol{v}$
4: **end for**
5: $\boldsymbol{z} = \boldsymbol{z}_1 + \cdots + \boldsymbol{z}_N$
6: Output: $M_{AS}^{-1}\boldsymbol{v} = \boldsymbol{z}$

Schwarz preconditioners and Krylov subspace methods

The multiplicative and additive Schwarz preconditioners can be applied to solving the general system $A\boldsymbol{x} = \boldsymbol{f}$ with a Krylov subspace method, such as GMRES. The algorithm of the preconditioned Krylov method is used, as described in Subsection 4.3.1, combined with the algorithms used to compute the action of the preconditioner on the desired vector, see Algorithms 6 and 8. These Schwarz preconditioners apply local corrections that are easier to fix than the global error. The overlap of the subdomains allows for propagation of the corrections such that the global error is smoothed. The combination of the local corrections results in a reduction of the global error.

The level of overlap δ used in SAPs has an influence on the performance and effectiveness of the multiplicative and additive Schwarz preconditioners.

Restricted Additive Schwarz preconditioners

The restricted additive Schwarz (RAS) preconditioner is a variant of the AS preconditioner that reduces the number of iterations needed and CPU time required to compute the preconditioner, proposed by Cai and Sarkis [32]. The RAS preconditioner is defined as

$$M_{RAS}^{-1} = \sum \hat{R_i^0}^T A_i^{-1} R_i^\delta. \tag{4.31}$$

Here, $\hat{R_i^0}^T$ is the prolongation operator corresponding to a zero-level overlap ${R_i^0}^T$, extended with zero-columns such that it is a $n \times n_{i,\delta}$ matrix. Equivalently, $\hat{R_i^0}^T$ is the prolongation operator corresponding to a δ-level overlap, ${R_i^\delta}^T$, where the non-zero entries of the columns corresponding to the overlap points are set to zero. This RAS preconditioner uses known information from all points in Ω_i^δ, including overlap, to compute the local update. However, only the update for the internal points, so those in Ω_i^0, is used for the global update. As a result, half of the communication costs can be saved in a parallel implementation, since $\hat{R_i^0}^T A_i^{-1}$ does not involve any data exchange.

Cai and Sarkis [32] found that the RAS preconditioner outperformed the standard AS preconditioner for all considered test cases, in terms of iteration counts, CPU time and communication costs if implemented in parallel. It was also shown that for M-matrices the RAS preconditioner does not require a damping parameter for convergence, while AS often does [33].

Two-level additive Schwarz preconditioners

The principle of a *multigrid* method can also be combined with AS preconditioners, referred to as *two-level* AS methods. In addition to the regular AS preconditioner, as in (4.30), that consists of the sum of solves on subdomains, a coarse solve on a coarse domain is also done. This results in the two-level AS preconditioner, given by

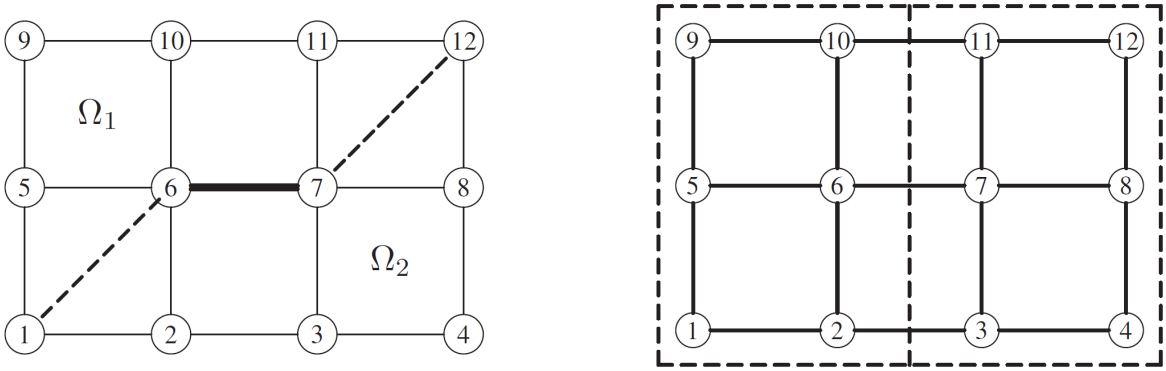
$$M_{AS,2L}^{-1} = \sum_{i=1}^{N} {R_i^\delta}^T A_i^{-1} R_i^\delta + \Phi^T A'^{-1} \Phi, \tag{4.32}$$

where $\Phi : \Omega \to \Omega'$ is the restriction operator for coarse space Ω', and $A' = \Phi A \Phi^T$ is the restriction of the global matrix A to the coarse subdomain [34, 35]. The goal of the added coarse space correction to the preconditioner is to correct global error that are not fixed by the local subdomain solves. The coarse subspace does not have to coincide or depend on the subdomains Ω_i^δ, but it is done often in practice. As an example, a node v_i in the coarse space representing a gas network can represent the average value of the nodal pressures for the nodes in subdomain Ω_i^δ.

4.5.3 Graph Partitioning

The method used for partitioning of an energy network directly determines the computational efficiency of preconditioner-based algorithms [36]. There are various methods to partition a graph, generally classified as *link-based* (or edge-based) and *node-based* (or vertex-based) partitioning, see Figure 4.3. A link-based partitioning does not allow links to be split between two subdomains. Instead, the boundary is made up of vertices. Interface links are the links that connect nodes that do not belong to the same subdomain, such as the link connecting nodes 6 and 7 in Figure 4.3a. Conversely, a node-based partitioning does not allow vertices to be split between to subdomains such that the boundary consists of links. Interface nodes are the nodes that are connected by links belonging to at least two different subdomains, such as node 6 in Figure 4.3b. In a node-based partitioning, the links are cut to create the subdomains. The number of *link-cuts* refers to the number of links $e_k = \{v_i, v_j\}$ between nodes v_i and v_j, such that $v_i \in \Omega_\alpha$, while $v_2 \in \Omega_\beta$.

A multi-level graph partitioning algorithm is an algorithm that determines a partition in three phases [37].



(a) Link-based partitioning

(b) Node-based partitioning

Figure 4.3: Taken from [22]: 'Link-based and node-based partitioning of a of a 4×3 mesh into two subregions.'

1. **Coarsening phase**: The original graph is transformed into a sequence of coarsened graph, such that each coarsening reduces the number of nodes.
2. **Partitioning phase**: A partitioning of the coarsened graph is determined.
3. **Un-coarsening and refinement phase**: The partition is projected back to the original graph and refined.

Such multi-level graph partitioning algorithms that are often applied in parallel processes are METIS and ParMETIS [37, 38, 36]. The METIS algorithm is based on the three steps mentioned previously, and is widely applicable to both structured and unstructured graphs. It is an efficient algorithm for balanced partitions, but slower on irregular structures. The parMETIS algorithm is a parallelized version of the METIS algorithm, using parallel methods in the coarsening and un-coarsening and refinement phase. Runtime improves with the parallelized algorithms, but scalability diminishes due to synchronization costs.

Chapter 5

State of the Art

This Chapter gives an overview of research done related to large scale multi-carrier energy networks. The overview includes research into an integrated multi-carrier energy system as well as research of a single carrier, such as electricity and gas. The aim of this chapter is to get an insight into recent developments related to modelling and simulating large scale multi-carrier energy networks and to identify research gaps.

The chapter starts with an overview of research identifying when and why Jacobian matrices of energy networks are ill-conditioned. The next section identifies which solution methods, direct and iterative, are used in simulating the multi-carrier and single-carrier networks and includes the argumentation of the authors. Section 5.3 lists two types of preconditioners found in recent developments and compares their performances in different studies. Next, Section 5.4 briefly discusses studies that proposed domain decomposition type methods for single-carrier networks. Section 5.5 includes studies that focus on solving multi-carrier networks, either using decomposed or integrated methods. The different types of partitioning methods used in studies through domain decomposition methods as well as decomposed multi-carrier solution methods are listed in Section 5.6. The chapter concludes with a section that summarizes the important findings and conclusions of the literature study.

5.1 Ill-Conditioned Jacobian Matrices

The convergence of Krylov solvers is highly dependent on the condition number of the coefficient matrix. If a matrix has poor spectral properties, the method will converge very slowly or even diverge. The application of an appropriate preconditioner can improve the condition number, and therefore the convergence behaviour of the Krylov solver. It is thus relevant to investigate how the coefficient matrices (Jacobians) are conditioned prior to applying an iterative solver. Literature mentions several aspects that worsen the condition number of the Jacobian matrices of power and gas networks.

In [21], it was observed that power systems with more buses have larger condition numbers and that the eigenvalues take up larger regions than systems with less buses. The same observation was done for differently-sized gas networks in [39].

High and extreme loading of power systems leads to a higher condition number of the associated Jacobian [29, 40, 41]. High loading of a power system pushes it closer to an infeasible operation zone, leading to an ill-conditioned Jacobian. The methods used in [29] were able to solve the load-flow problems of sample networks with loading levels up to 160%, which they mention is beyond practical loading levels.

The electrical properties of the transmission lines also affects the condition number of the Jacobian of power systems [21, 41]. The R/X-ratio is the ratio of the inductive reactance and the resistance and

a high ratio indicates that the system is more inductive. It results in the off-diagonal elements of the Jacobian to become much smaller than the diagonal elements, leading to near singularity.

A fourth aspect that influences the structure and conditioning of power system Jacobians is the topology of the network. In [41], it is mentioned that radial power systems have high condition numbers. In a radial power system there is only one path in the network from a source to a load point. This topology occurs often in distribution systems. There are no loops in a radial network, thus a fault anywhere in the network will result in a loss of power for all downstream users. The corresponding Jacobian matrix has more zero-entries, leading to a highly sparse structure that can result in a high condition number. In [21], the condition numbers of longitudinal and meshed networks are compared. A longitudinal network models weak and long-distance power grids, often with a radial structure. A meshed network models an interconnected network where power can flow along multiple paths (loops). It was concluded that the longitudinal systems have condition numbers that are significantly higher than meshed systems. Additionally, they conclude that the location of the slack bus is more influential in meshed networks than in longitudinal networks.

5.2 Choice of Solution Method

Various methods exist to solve the linear systems resulting for energy flow simulations. The choice of linear solution method is important to ensure convergence, efficiency and to keep the required computational time within reasonable limits. Solution methods for linear systems can be direct methods or iterative methods. Commonly used methods are described in Sections 4.1 and 4.2.

Direct methods, such as LU decomposition, are often used in practice [40], as there are powerful direct solvers available that can solve large, sparse and ill-conditioned systems. Additionally, a direct method is usually less difficult to work with and implement than iterative methods. Methods based on LU decomposition have become the standard for the linear systems solve in the power systems community [24], and are also employed for small coupled gas-electric systems [7]. Current research on direct solvers for power systems focuses on their performance and implementation based on central processing unit (CPU)/graphics processing unit (GPU) accelerations [24, 42].

Direct solvers scale poorly in problem size, increasing the computational efforts needed to converge significantly. This indicates that iterative solvers are more suitable to the problem. In research by Idema et al. [29], all considered preconditioned generalized minimum residual (GMRES) methods converged in less than one minute, whereas the direct linear solver took more than an hour to solve the large-scale linear problem. It is thus generally recommended to use iterative solvers for solving large power flow problems [21, 28, 43]. The same results have been observed for transient gas networks [44]. In their experiments with large-scale networks, the direct solver either took up too much CPU time or ran out of memory, while the Krylov solvers converged. However, for smaller networks, the direct solver shows an advantage over the preconditioned Krylov solvers due to their big overhead. Thus for smaller energy networks a direct solver is more suitable, while for larger networks the class of Krylov solvers should be considered.

The most commonly used Krylov solver for the linear solution step in power flow, as well as gas network simulations, is the GMRES method, although different arguments are given for this choice. The only general requirement for the coefficient matrix for GMRES is non-singularity, making it widely applicable, also to coupled networks [31]. Other argumentations for using GMRES is related to its behaviour and convergence properties. Full GMRES solves the problem in the least number of matrix-vector multiplications, which is the reason for Idema et al. [29] to use this method. Additionally, they state that the performance of GMRES can indicate whether bi-conjugate gradient stabilized (Bi-CGSTAB) or IDR(s) would be more suitable. Another advantage of GMRES is given in [40], where it is mentioned that GMRES is free of problems that interrupt the iterative process of constructing an orthonormal basis due to the underlying use of the Arnoldi method with Gram-Schmidt, opposed to methods based

on Lanczos' method.

Another Krylov subspace method that is mentioned in literature is Bi-CGSTAB. The performance of Bi-CGSTAB is investigated in [28] as it is a variant of the extended (preconditioned) CG method, which had shown good convergence characteristics in DC power flow calculations. The preconditioned Bi-CGSTAB method outperformed the preconditioned conjugate gradient squared (CGS) method and preconditioned conjugate residual (CR) method in terms of computational effort. Additionally, preconditioned Bi-CGSTAB outperformed other Krylov methods such as CGS, GMRES and Bi-CG in the experiments done in [21]. In the application of steady-state gas networks, the authors of [39] observed similar performance for preconditioned GMRES, Bi-CGSTAB and IDR(s).

The iterative Krylov method applied to energy system modelling that is used most often in practice is GMRES. It is a robust method that is applicable to a large range of problems and serves as a good method to start with. Moreover, the choice of preconditioner is at least as important and influential on the convergence and performance as the selected method [22].

5.3 Preconditioning

The choice of preconditioner is at least as important, if not even more, as the iterative method in solving large linear systems. The general preconditioning procedure is discussed in Section 4.3.

5.3.1 ILU preconditioner

A class of preconditioners that is often used in energy system modelling is based on an incomplete LU factorization.

Roque et al. [40] combined an incomplete LU (ILU) preconditioner with GMRES to study the performance of Krylov solvers for ill-conditioned systems resulting from the Newton-Raphson (NR) method applied to electricity systems. They investigate the effect *dropping tolerance* on the performance of the preconditioner. In the LU decomposition, every element with $|l_{ij}| < \epsilon_{ILU}$ and $|u_{ij}| < \epsilon_{ILU}$ is replaced with a zero, with dropping tolerances ϵ_{ILU} ranging from 10^{-3} to 10^{-7}. A larger tolerance drops more fill-in elements, making the preconditioner cheaper to apply, but the approximation of the ILU preconditioner of the original coefficient matrix is less accurate. As the dropping tolerance increases, the preconditioner gets closer to the full LU matrix and the linear system tends to converge faster. In a later article, the same authors mentioned that this tolerance-based dropping rule can introduce large errors, resulting in a low quality preconditioner [45]. As a remedy, they propose a dropping rule based on different criteria and highlight the effects reordering can have on the preconditioner quality.

Nguyen, Romate, and Vuik [39] also combined an ILU preconditioner with GMRES, but in the application of gas network load-flow equations. More specifically, the ILUTP preconditioner is used, which is based on an incomplete LU factorization with a dropping threshold and pivoting. They conclude that for the used test networks, the ILU preconditioner is not suitable, since the computation of the preconditioner requires nearly the same amount of work as for the LU factorization. In that scenario, using a direct solver based on LU is more suitable. For future research they recommend the Inexact-Newton method, keeping the same ILU factorization until the non-linear system has changed sufficiently.

Idema et al. [29] compared different ILU(k) preconditioners in simulations of large power flow problems, where good performance of ILU(k) with GMRES was observed. Additionally, they emphasize the importance of reordering methods in computing ILU factorizations. They concluded that using the approximate minimum degree (AMD) reordering for an ILU(k) preconditioner in GMRES resulted in around 25% less iterations needed for convergence compared to using no reordering. They also compared the performance of the ILU(k) preconditioner of different target matrices, the Jacobian J_i, the initial Jacobian J_0 and the fast decoupled load flow (FDLF) matrix Φ. Using J_0 or Φ generally per-

formed better than J_i. Since the J_0 preconditioner is the same in each iteration, only a single expensive factorization has to be made.

Mori and Iizuka [28] compared the performance of the ILU(k) preconditioner for $k = 0, 1, 2$ for three different Krylov subspace methods, applied to three differently sized power systems. For the largest test case considered, the use of the ILU(2) preconditioner resulted in the lowest total computation times as well as iterations. For the smaller test cases, the ILU(2) preconditioner always resulted in the fewest amount of iterations needed for convergence, but not always the lowest computation time. This is attributed to the more expensive factorization of ILU(2) compared to ILU(1) or ILU(0).

The differences in performance and results of ILU preconditioned GMRES for electricity load-flow and gas networks, highlight that preconditioning is problem-dependent. The different versions and techniques used in ILU preconditioning such as reordering, dropping rule definitions and tolerances, all influence convergence and performance. On one hand this can lead to customized ILU preconditioners that can enhance convergence of Krylov methods. On the other hand, this makes the search for the best combination of parameters a complicated procedure.

5.3.2 Additive Schwarz Preconditioner

Additive and multiplicate Schwarz alternating procedures (SAPs) form the basis of additive and multiplicative Schwarz preconditioners, where corrections on local subdomains aim to reduce the global error. The additive Schwarz (AS) preconditioner is popular in research focussing on reducing total computational time to solve load-flow problems, due to its natural parallelizability.

The overlapping restricted additive Schwarz (RAS) preconditioner applied to very large power flow problems was evaluated by Abhyankar, Smith, and Constantinescu [46] and in [47]. They combine the RAS preconditioner with the GMRES method to solve several large power flow problems. They compare the performance of the preconditioner with a block-Jacobi preconditioner and a parallel direct solver based on LU decomposition. Their results show a substantial speed-up of the RAS preconditioner compared to the other methods, ranging from 6 to 9 on 16 cores. The influence of the level of overlap was also investigated, and the authors concluded that a speed-up saturation and slowdown were observed for a level of overlap higher than 3. This observation is attributed to the increase in communications for higher levels of overlap.

5.4 Domain Decomposition in Single-Carrier Energy Networks

A single-carrier (SC) energy network can be solved using the principle of divide-and-conquer, where the global domain is split into several subdomains. Solving the local problems sequentially then solves the global problem. Several applications in SC networks can be found in literature.

A study of Kootte, Sereeter, and Vuik [43] reviewed the numerical performance of connection methods of integrated transmission-distribution electrical networks. A transmission network is often modelled differently than a distribution network due to differences in assumptions, simplifications and physical properties. It is therefore not straightforward to integrate these separate domains. The study considers unified methods that solve the integrated network as a whole using a transformer, and splitting methods that keep the sub-networks separate to iteratively solve on the boundary of the domain. These splitting methods are also referred to as master-slave splitting (MSS) methods. Their results show that the unified methods outperform the splitting methods for the considered test cases, since splitting methods must solve the separate transmission and distribution networks multiple times until convergence, increasing the total CPU time. However, the authors mention that MSS methods are suitable in practical situations where full network information cannot be shared. For future work, they recommend to consider preconditioned Newton-Krylov methods for very large systems and note that MSS methods are parallelizable.

Zhang et al. [8] introduced a partial decoupling method for fast calculation of energy flow in large-scale heat and electricity integrated energy systems, that operate in a heat-load-following mode. They propose a partial decoupling method for disctrict heating system (DHS) to decouple a network of arbitrary topology into several radial subsystems and one main system. Radial systems are preferred over looped systems as the hydraulic part of the load-flow equations of radial systems are linear. All unknowns of the subsystems are initialized, after which the DHS load-flow equations are solved iteratively. The values of the boundary nodes are updated after each subsystem solve. The global system has converged when the differences in temperatures and nodal mass flows of the boundary nodes in both systems become sufficiently small. This principle is similar to the multiplicative Schwarz procedure. Their partial decoupling method for various DHS performs about the same as traditional methods in terms of accuracy, while improving computational efficiency and sensitivity to improper hydraulic initialization.

5.5 Solvers for Multi-Carrier Energy Systems

Studies on hybrid energy flow of multi-carrier energy systems often focus on the establishment of models and formulations [9, 10], while only a few studies are dedicated to the solution methods used in such energy flow calculations [31]. Several studies on smaller multi-carrier energy networks make use of NR, combined with direct solvers for the linear systems [7, 9]. Solving large integrated energy networks all at once remains an open challenge [25].

5.5.1 Decomposed Methods

Some studies on a solving multi-energy system (MES) investigate the possibility of decoupling a coupled MES, and using direct solvers to solve the smaller or simpler sub-systems.

Markensteijn formulated a decoupled approach to solve load flow problems of coupled multi-carrier energy networks [6]. Through the use of interface conditions (IFCs), she describes how a coupled network can be formulated such that the single-carrier networks can be solved separately. The blocks in the Jacobian of a coupled network are generally not square. In her decoupled formulation, the SC submatrices in the Jacobian are made square by including information obtained from the coupling in the SC subnetwork. It is mentioned that the decoupled formulation is equivalent to a permutation of the Jacobian matrix of the coupled system, as in (4.20). Markensteijn states that when the Jacobian matrix of a coupled network is permuted such that the SC sub-matrices are square, domain decomposition or block iterative methods can be used to solve the linear system.

Zhang et al. [8] introduced a partial decoupling method for fast calculation of energy flow in large-scale heat and electricity integrated energy systems, that operate in a heat-load-following mode. Although the focus of the study is on the partial decoupling method for a DHS, they also solve a coupled electro-thermal system. Since they consider a coupled system in heat-load-following mode, their methodology of solving the coupled system consists of first solving the DHS iteratively, then updating the electric power of the combined heat and power (CHP) units, and finally calculating the electric power flow using NR combined with a direct solver. The errors in the partial decoupling method for the DHS have little effect on the convergence of NR for the coupled electrical network, which is desirable.

5.5.2 Integrated Methods

Jia et al. [11] conducted an investigation into problems of the integrated method for electro-thermal energy systems. The integrated method fails in exceptional cases when energy reverses during iterations. Additionally, it is sensitive to the choice of initial values.

Krylov subspace methods in multi-carrier systems

Qi, Li, and Bie [31] studied the Inexact Newton method with preconditioned GMRES for hybrid energy flow calculation of integrated energy systems, using an electro-thermal coupling system as a test case [31]. Their results show a significant acceleration effect on the calculation speed compared to classical methods, and it is therefore recommended to further investigate (preconditioned) iterative solvers applied to integrated energy systems.

5.6 Partitioning Methods

Various methods of partitioning SC and multi-carrier energy networks are used in research. Sections 5.4 and 5.5.1 discussed the effects of the partitioning on solving the load-flow equations of the various energy networks. In this section, the partitioning methods used in literature are mentioned and compared.

The coupled MES in the study of Markensteijn [6] are partitioned using a node-based partitioning, by cutting all dummy links that connect a homogeneous node to a coupling node. This cutting of dummy links results in two terminal links, one in each subdomain. Due to this natural partition, the subdomains consist of the individual electricity, gas and heat networks, as well as a subdomain with only coupling nodes. A similar partitioning was used in [8] to decouple an electro-thermal system into several DHS, one large electrical system and coupling units.

The partial decoupling method for a DHS used by Zhang et al. [8] makes use of link-based partitioning (see Subsection 4.5.3) by splitting an intermediate node (branch bus, no load or source) into a load node for the first subsystem, and a slack node for the second subsystem. The goal of the partitioning is to create only radial subsystems that are more robust regarding initialization.

Abhyankar, Smith, and Constantinescu [46] also make use of a partitioning method of power systems in order to use a RAS preconditioner. It is mentioned that obtaining a good graph partition is a critical aspect for parallel linear solution methods, but that it is beyond the scope of their research. For the partitioning of the electricity network, they used a multi-level graph partitioning algorithm.

The study of [36] compares efficient graph partitioning algorithms for sparse matrices, considering various multi-level algorithms. The application of METIS and ParMETIS, among other methods, to standard datasets showed that METIS had consistent performance, ideal for small and medium number of partitions. ParMETIS demonstrated superior scalability for a higher number of partitions.

From the literature mentioned above we can see that partitioning used in energy systems can be based on the physical structure of the network as well as computational optimality. The physics-based partitioning used to partition a multi-carrier energy network is generally node-based, such that nodes of the same carrier are partitioned together. The partitioning of nodes of the same carrier is done in various ways, either focussing on accuracy of the obtained solution of the partitioned system or computational efficiency.

5.7 Conclusion and Summary of Findings

Research into solvers for large scale multi-carrier energy networks (MCNs) has been limited so far. Most studies either focus on the formulation of such MCNs or on the solvers used in a SC network. Extensive research has been done on power networks, for small as well as large scale networks. It was found that for large scale SC as well as MCNs, the use of iterative solvers is necessary to achieve accurate and fast solutions to the load-flow equations. Direct solvers should be used for smaller (sub)networks or as a local solver within a decomposition method.

The use of a preconditioner is important when using an iterative solver, and the choice of precon-

ditioner is a highly influential choice in terms of solver performance. However, choosing an appropriate preconditioner is problem-dependent. Due to the customizability of many preconditioning methods, finding an appropriate one is not always a straightforward task. A commonly used preconditioning technique is based on ILU decomposition, and many variants of this method are used in energy network modelling. These methods show good results for power flow problems in multiple studies. However, it was found that ILU is not a well-suited preconditioner for gas network modelling.

A different class of methods and preconditioners arises from the domain decomposition (DD) principle. A large scale and/or coupled energy system naturally leads to partitioning the global problem into smaller and local problems. Research has been done on solving SC as well as MCNs using such DD methods, although the test problems were mostly small or medium scale. The performance of a variant of the AS preconditioner was studied in a power flow setting, but not yet for gas and coupled networks. With the usage of a DD method, comes the task of partitioning the domain. In the context of energy network modelling, such a partition can either be done based on the physical properties of the network, or on the mathematical properties of the network graph. Research on partitioning methods has been done in other fields where DD is often used and in some SC energy network settings, but not in the context of gas and coupled energy networks.

Some research has been done on solving MCNs, either by decomposing the network into SC components, or by employing an integrated method. These studies show promising results, but extensive research on this topic is still lacking.

Chapter 6

Conclusion

This chapter contains the concluding remarks of this literature study. The findings of the literature review are summarized in Section 5.7. In this chapter, the implications of the literature study are listed, and can be found in Section 6.1. Finally, the research question and sub-questions are presented and elaborated on in Section 6.2.

6.1 Implications and Open Problems

The following implications and corresponding open problems in modelling and simulating large scale multi-carrier energy networks (MCNs) were determined, based on the findings in Chapter 5.

- The selection of an appropriate preconditioner can be as important as the choice of iterative solution method. Furthermore, the performance of a preconditioned iterative method is problem-dependent and the number of hyperparameters often makes them highly customizable.
- Integrated solution methods for MCNs are at an early stage and present research opportunities.
- The numerical performance of solution methods is the limiting factor for fast, robust and accurate simulations for large scale MCNs and requires research into preconditioned iterative methods, using a decomposition of the network.
- The application of preconditioners based on Schwarz alternating procedures (SAPs) and graph-based (multi-level) partitioning methods applied to gas and coupled energy networks present research opportunities.
- The application of incomplete LU (ILU) preconditioners in power flow problems has been studied extensively and is generally found to perform well. ILU preconditioning is not appropriate for gas network simulations. Alternative preconditioning strategies are therefore an open problem.

6.2 Preliminary Research Questions and Scope

Based on the literature review in Chapter 5 and the implications in Section 6.1, the research scope, question and sub-questions for the continuation of the research were determined.

The scope of this research project is to study fast and robust preconditioned Krylov methods for solving the steady-state load-flow equations of large scale MCNs, focussing on finding a suitable preconditioner. Based on the literature review and implications it was decided to focus the research on the application of additive Schwarz (AS) preconditioners. This choice was made firstly because the restricted AS preconditioner gave good numerical results in research done by Abhyankar, Smith, and Constantinescu [46]. Secondly, the AS preconditioner was chosen instead of the ILU preconditioner for this research, since the

performance of ILU has been studied extensively already, also in the application to power flow problems.

The preliminary research question for the next phase of this research study is:

> *How does an additive Schwarz preconditioner perform when applied to a Krylov solver in the solution process of the steady-state load flow equations of large scale coupled gas-electricity energy networks?*

In order to answer this research question, there were several sub-questions formulated to support the main research goal:

1. How does an additive Schwarz preconditioner perform when solving steady-state load flow equations of gas and electricity networks separately?
2. How do different partitioning strategies affect the performance of the additive Schwarz preconditioner?
3. How do variants of additive Schwarz preconditioner compare?
4. What is the effect of coupling gas and electricity networks on the performance of an additive Schwarz preconditioner?

The first steps of the next phase of the research project consist of acquiring and constructing appropriate data sets, and gaining insight into the application and implementation of AS preconditioners, combined with partitioning methods. Note that parallel implementations are out of scope for this phase of the research, since the focus is to find a suitable solution method that leads to fast and accurate load flow simulations.

The main preliminary research question is about investigating the performance of the AS preconditioner. In order to answer this question, it has to be clarified on what performance metrics the preconditioner will be analysed and against what alternative methods. Various methods of comparison used in research to get an indication of the performance of a preconditioner are:

- Accuracy: Compute and keep track of the error and/or residual during the iterative solution process. Most importantly, this indicates whether the methods converge. Additionally, this gives an indication of how accurate the approximate solution is at what iteration and how many iterations are required to reach a predetermined tolerance.
- Speed: Keep track of the number of iterations required for convergence, for the inner as well as the outer loop. Additionally, it is insightful to measure the CPU time of the phases of the solution process. Both of these metrics indicate how fast the solution is computed, and what potential bottlenecks are.
- Cost: The amount of memory and number of matrix-vector products required by the solution method indicates how much storage is required and how costly it is to execute the method. These results can also be used to identify potential bottlenecks and areas of improvement.
- Robustness: The conditioning of the underlying coefficient matrix is influential on the performance of the methods. Comparing the robustness of solution methods indicates how robust the method is and whether it is sensitive to the matrix conditioning or initialization.
- Scalability: The research area of this study includes large scale energy networks. Therefore it is important to consider the scalability of the solution method, such that it is able to solve large systems of linearized load-flow equations. The algorithmic scalability can be determined by comparing the performance of the solution method in terms of speed, accuracy and/or cost for test problems varying in problem size.

Bibliography

[1] *Global Warming of 1.5° C: IPCC Special Report on Impacts of Global Warming of 1.5° C above Pre-Industrial Levels in Context of Strengthening Response to Climate Change, Sustainable Development, and Efforts to Eradicate Poverty*. Tech. rep. Cambridge: Intergovernmental Panel on Climate Change (IPCC), 2022. URL: https://www.ipcc.ch/sr15/.

[2] *Paris Agreement*. Tech. rep. Paris, France: United Nations Framework Convention on Climate Change (UNFCCC), Dec. 2015.

[3] *Klimaatwet*. Dutch. Tech. rep. Dutch Government, 2019. URL: https://wetten.overheid.nl/BWBR0042394/2023-07-22.

[4] Martin Scheepers et al. "Towards a climate-neutral energy system in the Netherlands". In: *Renewable and Sustainable Energy Reviews* 158 (Apr. 2022). ISSN: 18790690. DOI: 10.1016/j.rser.2022.112097.

[5] Kelvin E. Bassey. "Hybrid Renewable Energy Systems Modeling". In: *Engineering Science & Technology Journal* 4.6 (2023), pp. 571–588. DOI: 10.51594/estj/v4i6.1255.

[6] Anne S. Markensteijn. "Mathematical models for simulation and optimization of multi-carrier energy systems". PhD thesis. Delft University of Technology, 2021. DOI: 10.4233/uuid:044c083a-4eb1-4999-b789-db7150c4c7df.

[7] Xiangyang Men et al. "Coupling analysis of gas-electric hybrid system based on Newton–Raphson method". In: *The Journal of Engineering* 2017.13 (Oct. 2017), pp. 1505–1510. ISSN: 2051-3305. DOI: 10.1049/joe.2017.0582.

[8] Suhan Zhang et al. "Partitional decoupling method for fast calculation of energy flow in a large-scale heat and electricity integrated energy system". In: *IEEE Transactions on Sustainable Energy* 12.1 (Jan. 2021), pp. 501–513. ISSN: 19493037. DOI: 10.1109/TSTE.2020.3008189.

[9] Jianzhang Wu et al. "Energy Flow Calculation Method for Multi-Energy Systems: A Matrix Approach Considering Alternative Gas Injection and Dynamic Flow Direction". In: *Applied Sciences (Switzerland)* 15.9 (May 2025). ISSN: 20763417. DOI: 10.3390/app15094815.

[10] Mengting Zhu et al. "An integrated multi-energy flow calculation method for electricity-gas-thermal integrated energy systems". In: *Protection and Control of Modern Power Systems* 6.1 (Jan. 2021), pp. 1–2. ISSN: 23670983. DOI: 10.1186/s41601-021-00182-2.

[11] Mengshuo Jia et al. "An Investigation on the Applicability of the Integrated Method for Multi-Carrier Energy Flow Analysis". In: *2018 IEEE Power & Energy Society General Meeting (PESGM)*. Portland, USA: IEEE, Aug. 2018, pp. 1–5. ISBN: 9781538677032. DOI: 10.1109/PESGM.2018.8585831.

[12] Pieter Schavemaker and Lou van der Sluis. *Electrical power system essentials*. 2nd ed. Chichester, West Sussex: John Wiley & Sons, 2017. ISBN: 9781118803479.

[13] Arthur R. Bergen and Vijay Vittal. *Power System Analysis*. 2nd ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 2000. ISBN: 0136919901.

[14] Charles K. Alexander and Matthew N. O. Sadiku. *Fundamentals of electric circuits*. 5th ed. New York, New York: McGraw-Hill, 2013. ISBN: 9780073380575.

[15] James L. Kirtley. *Electric power principles: sources, conversion, distribution, and use*. John Wiley & Sons, 2010. ISBN: 9780470686362.

[16] Andrzej J. Osiadacz. *Simulation and analysis of gas networks*. London: E. & F.N. Spon, 1987. ISBN: 9780419124801.

[17] Thorsten Koch et al. *Evaluating Gas Network Capacities*. Philadelphia, Pennsylvania: Society for Industrial and Applied Mathematics, 2015. ISBN: 9781611973686. DOI: 10.1137/1.9781611973693.

[18] Reinhard Diestel. *Graph Theory*. 6th. Heidelberg: Springer Berlin, 2024. ISBN: 9783662701065. DOI: 10.1007/978-3-662-70107-2.

[19] Anne S. Markensteijn, Johan E. Romate, and Cornelis Vuik. "A graph-based model framework for steady-state load flow problems of general multi-carrier energy systems". In: *Applied Energy* 280 (Dec. 2020), p. 115286. ISSN: 0306-2619. DOI: 10.1016/J.APENERGY.2020.115286.

[20] Reijer Idema et al. "Towards faster solution of large power flow problems". In: *IEEE Transactions on Power Systems* 28.4 (2013), pp. 4918–4925. ISSN: 08858950. DOI: 10.1109/TPWRS.2013.2252631.

[21] Francisco De León and A. Semlyen. "Iterative solvers in the Newton power flow problem: Preconditioners, inexact solutions and partial Jacobian updates". In: *IEE Proceedings: Generation, Transmission and Distribution*. Vol. 149. 4. July 2002, pp. 479–484. DOI: 10.1049/ip-gtd:20020172.

[22] Yousef Saad. *Iterative methods for sparse linear systems*. 2nd ed. Society for Industrial and Applied Mathematics, 2003, p. 528. ISBN: 0898715342.

[23] Reijer Idema and Domenico J.P. Lahaye. *Computational Methods in Power System Analysis*. Ed. by Will Schilders. Paris: Atlantis Press, 2014. ISBN: 9789462390638. DOI: 10.2291/9789462390645.

[24] Lukas Razik et al. "A comparative analysis of LU decomposition methods for power system simulations". In: *2019 IEEE Milan PowerTech*. Milan, Italy: IEEE, Aug. 2019, pp. 1–6. DOI: 10.1109/PTC.2019.8810616.

[25] Buu-Van Nguyen, Johan Romate, and Cornelis Vuik. "Modeling an Electrolyzer in a Graph-Based Framework". In: *Energies 2025, Vol. 18, Page 729* 18.3 (Feb. 2025), p. 729. ISSN: 1996-1073. DOI: 10.3390/EN18030729.

[26] Yousef Saad and Martin H Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (July 1986), pp. 856–869. DOI: 10.1137/0907058.

[27] H.A. Van der Vorst. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systes". In: *SIAM Journal on Scientific and Statistical Computing* 13.2 (Mar. 1992), pp. 631–644. DOI: https://doi.org/10.1137/0913035.

[28] Hiroyuki Mori and Fumitaka Iizuka. "An ILU(p)-Preconditioner Bi-CGStab Methodfor Power Flow Calculation". In: *2007 IEEE Lausanne Power Tech*. IEEE, July 2007, p. 2235. ISBN: 9781424421909. DOI: 10.1109/PCT.2007.4538533.

[29] Reijer Idema et al. "Scalable Newton-Krylov solver for very large power flow problems". In: *IEEE Transactions on Power Systems* 27.1 (Feb. 2012), pp. 390–396. ISSN: 08858950. DOI: 10.1109/TPWRS.2011.2165860.

[30] Ron S. Dembot, Stanley C. Eisenstatt, and Trond Steihaug. "Inexact Newton Methods". In: *SIAM Journal on Numerical Analysis* 19.2 (Apr. 1982), pp. 400–408. DOI: 10.1137/0719025.

[31] Sen Qi, Gengfeng Li, and Zhaohong Bie. "Hybrid Energy Flow Calculation for Electric-Thermal Coupling System Based on Inexact Newton Method". In: *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*. Beijing, China: IEEE, Nov. 2019. ISBN: 9781728149301. DOI: 10.1109/iSPEC48194.2019.8975152.

[32] Xiao-Chuan Cai and Marcus Sarkis. "A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems". In: *SIAM* 21.2 (1999), pp. 792–797.

[33] Andreas Frommer and Daniel B. Szyld. "An Algebraic Convergence Theory for Restricted Additive Schwarz Methods Using Weighted Max Norms". In: *Society for Industrial and Applied Mathematics* 39.2 (2001), pp. 463–479. DOI: 10.1137/S0036142900370824.

[34] Andrea Tosseli and Olof B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Vol. 34. Springer Series in Computational Mathematics, 2005.

[35] Filipe Cumaru, Alexander Heinlein, and Joachim Schöberl. *Two-level additive Schwarz preconditioners for reduced integration methods*. Tech. rep. Delft Institute of Applied Mathematics, Dec. 2025. DOI: 10.48550/arXiv.2512.01706.

[36] Xue Chen, Jianwen Cao, and Xuesong Wu. "Research on Efficient Graph Partitioning Algorithms for Sparse Matrices". In: *2025 5th International Conference on Artificial Intelligence, Big Data and Algorithms, CAIBDA 2025*. Institute of Electrical and Electronics Engineers Inc., June 2025, pp. 571–574. ISBN: 9798331526641. DOI: 10.1109/CAIBDA65784.2025.11183378.

[37] George Karypis and Vipin Kumar. "Parallel multilevel k-way partitioning scheme for irregular graphs". In: *1996 ACM/IEEE Conference on Supercomputing*. Pittsburgh, USA: IEEE Computer Society, Nov. 1996, p. 35. DOI: 10.1145/369028.369103.

[38] George Karypis and Vipin Kumar. "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs". In: *SIAM Journal on Scientific Computing* 20.1 (1998), pp. 359–392. DOI: 10.1137/S1064827595287997. URL: http://www.siam.org/journals/sisc/20-1/28799.html.

[39] Buu-Van Nguyen, Johan Romate, and Cornelis Vuik. "Performance of Krylov solvers for gas networks". In: *E-ENERGY 2025 - Proceedings of the 2025 16th ACM International Conference on Future and Sustainable Energy Systems*. Association for Computing Machinery, Inc, June 2025, pp. 489–494. ISBN: 9798400711251. DOI: 10.1145/3679240.3734600.

[40] Matheus M. Roque et al. "Numerical Investigation on the Solution of Ill-Conditioned Load Flow Linear Equations". In: *Journal of Control, Automation and Electrical Systems* 30.4 (Aug. 2019), pp. 580–588. ISSN: 21953899. DOI: 10.1007/s40313-019-00461-2.

[41] M. Karimi et al. "Application of Newton-based load flow methods for determining steady-state condition of well and ill-conditioned power systems: A review". In: *International Journal of Electrical Power and Energy Systems* 113 (Dec. 2019), pp. 298–309. ISSN: 01420615. DOI: 10.1016/j.ijepes.2019.05.055.

[42] Manolo D'Orto et al. "Comparing Different Approaches for Solving Large Scale Power-Flow Problems with the Newton-Raphson Method". In: *IEEE Access* 9 (Apr. 2021), pp. 56604–56615. ISSN: 21693536. DOI: 10.1109/ACCESS.2021.3072338.

[43] Marieke E. Kootte, B. Sereeter, and Cornelis Vuik. *Solving the Steady-State Power Flow Problem on Integrated Transmission-Distribution Networks: A Comparison of Numerical Methods*. IEEE, Oct. 2020. ISBN: 9781728171005. DOI: 10.1109/ISGT-Europe47291.2020.9248852.

[44] Yue Qiu et al. "Efficient Numerical Methods for Gas Network Modeling and Simulation". In: *Networks and Heterogeneous Media* 15.4 (Nov. 2020), pp. 653–679. DOI: 10.3934/nhm.2020018.

[45] José E.O. Pessanha et al. "Critical investigation of preconditioned GMRES via incomplete LU factorization applied to power flow simulation". In: *International Journal of Electrical Power and Energy Systems* 33.10 (Dec. 2011), pp. 1695–1701. ISSN: 01420615. DOI: 10.1016/j.ijepes.2011.08.010.

[46] Shrirang Abhyankar, Barry Smith, and Emil Constantinescu. "Evaluation of overlapping restricted additive Schwarz preconditioning for parallel solution of very large power flow problems". In: *Proceedings of the 3rd International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*. Nov. 2013, pp. 1–8. ISBN: 9781450325103. DOI: 10.1145/2536780.2536784.

[47] Shrirang Abhyankar et al. "Parallel Dynamics Simulation Using a Krylov-Schwarz Linear Solution Scheme". In: *IEEE Transactions on Smart Grid* 8.3 (May 2017), pp. 1378–1386. ISSN: 19493053. DOI: 10.1109/TSG.2016.2610863.