



Master Thesis

Partial Outer Convexification for Compressor Optimization in a
Gas-to-Power Network

submitted by

Katharina Enin
born 28 May 1996 in Krasnoturinsk
Student ID: 1512450

University of Mannheim
Chair of Scientific Computing

Supervisor Prof. Dr. Simone Göttlich

Date of submission: 18.11.2021

Partial Outer Convexification for Compressor Optimization in a Gas-to-Power Network

Master thesis for obtaining the Master of Science Degree in Business Mathematics

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass diese Arbeit von mir persönlich verfasst wurde und dass ich keinerlei fremde Hilfe in Anspruch genommen habe. Ebenso versichere ich, dass diese Arbeit oder Teile daraus weder von mir selbst noch von anderen als Leistungsnachweise andernorts eingereicht wurden. Wörtliche oder sinngemäße Übernahmen aus anderen Schriften und Veröffentlichungen in gedruckter oder elektronischer Form sind gekennzeichnet. Sämtliche Sekundärliteratur und sonstige Quellen sind nachgewiesen und in der Bibliographie aufgeführt. Das Gleiche gilt für graphische Darstellungen und Bilder sowie für alle Internetquellen.

Ort, Abgabedatum

Unterschrift

Abstract

When solving a Mixed-Integer Optimal Control (MIOC) problem with partial differential equations as constraints, various techniques can be considered, such as the branch-and-bound method. However, in order to find an adequate representation, a fine discretization is necessary. This leads to excessive computing times due to a high number of integer variables.

A three-step optimization approach, which is based on the idea of partial outer convexification (POC) can be used to solve this issue. This approach has already been successfully applied to PDE constrained MIOC problems such as transmission lines [1] and traffic light optimization in road networks [2]. These papers concluded that the three-step heuristic usually works faster and delivers better control variables compared to a direct solver.

In this work we apply the three-step approach on a gas-to-power network where our aim is to minimize the compressor costs and find the corresponding pressure and mass flow in every grid point of the gas network. In order to apply the three-step heuristic based in partial outer convexification we first prove the applicability of an approximation result for positive density and positive mass flow. We compare the results of the three-step approach with those of a direct solver based on a branch-and-bound algorithm, where we consider both a POC and a non-POC formulated optimization problem. In the final step, we pose additional constraints that couple over time and compare the behaviour of the compressor state with the case where no additional constraints are present.

A comparison with a direct solver shows a great advantage of the three-step approach, as it was the fastest approach in all considered examples (up to 14 times faster) and is also able to find good solutions. We note that by imposing additional conditions that couple over time, we can even find better objective function values.

Contents

List of Symbols	ix
List of Abbreviations	xi
1. Introduction	1
2. Mathematical Theory	5
2.1. Conservation and Balance Laws	5
2.2. Discretization Methods	6
2.2.1. Discretization Methods for Hyperbolic Systems of Conservation Laws . .	6
2.2.2. Discretization Methods for Hyperbolic Systems of Balance Laws . . .	7
2.2.3. CFL-Condition	7
2.3. Selected Types of Problems	8
3. Partial Outer Convexification and Three-Step Method	11
3.1. Rounding Strategies for CIAP	16
3.1.1. Sum Up Rounding Strategy with SOS1	19
4. Numerical Methods	21
4.1. Methods for Solving a System of Nonlinear Equations	21
4.1.1. Newton-Raphson Method	21
4.1.2. Levenberg-Marquardt Method	22
4.2. Approaches for Solving MINLP	24
4.2.1. NLP-Based Branch-and-Bound	24
4.3. Solving NLP: Interior Point Method	30
5. Physical Characteristics of Gas-to-Power Networks	35
5.1. Power-to-Gas in Practice	35
5.2. Compressor Stations	36
5.3. Power Grid Model	38
5.4. 1D Euler Equation	40
5.4.1. Weymouth Equation	43
5.5. Node Coupling Conditions	44
5.5.1. Coupling Conditions at Gas Nodes	44

5.5.2. Coupling Conditions at Slack Node	44
5.5.3. Conditions at Compressor Node	45
6. Compressor Cost Optimization Model	47
6.1. Comparing Weymouth and Euler Equations Using Two Discretization Methods .	47
6.2. Problems Concerning Modeling Gas Networks with Lax-Friedrich Method . .	51
6.3. Compressor Cost Optimization Problems	52
6.3.1. Optimization Problem without POC	53
6.3.2. Optimization Problem with POC	54
6.4. Additional Constraints	55
6.5. Showing Theorem 3.1 for Weymouth Equation discretized with Simple Upwind .	57
7. Numerical Analysis	61
7.1. Solving the Power Model	61
7.2. Simple Model: One Compressor	64
7.3. Advanced Model: Two Compressors	70
7.4. Additional Constraints	73
7.4.1. Simple Model: One Compressor	73
7.4.2. Advanced Model: Two Compressors	74
8. Conclusion and Outlook	77
A. Appendix: Discrete Gronwall Lemma	79
B. Appendix: Global and Local Minima	81
C. Appendix: Karush-Kuhn-Tucker Conditions	83
D. Appendix: Code	85

List of Symbols

Properties in a Gas Network

g	gravitational acceleration ($\left[\frac{m}{s^2} \right]$)
p	pressure ([bar])
q	mass flow ($\left[\frac{kg}{s} \right]$)
Q	volumetric flow ($\left[\frac{m^3}{s} \right]$)
v	velocity ($\left[\frac{m}{s} \right]$)
ρ	density ($\left[\frac{kg}{m^3} \right]$)
ρ_0	norm density ($\left[\frac{kg}{m^3} \right]$)

Properties in a Power Model

B	susceptance (<i>Siemens</i>)
G	conductance (<i>Siemens</i>)
P	real power (<i>Watt</i>)
Q	reactive power (<i>Voltage – Ampere – Reactive</i>)
V	voltage amplitude (<i>Volt</i>)
ϕ	phase (<i>rad</i>)

List of Abbreviations

CIAP	Combinatorial Integral Approximation Problem
MIOC	Mixed-Integer Optimal Control
MINPL	Mixed-Integer Nonlinear Programming
MIP	Mixed-Inter (linear) Programming
MIPDECO	Mixed-Integer PDE-Constrained Optimization
NLP	Non-Linear Programming
PDE	Partial Differential Equation
POC	Partial Outer Convexification
PtG	Power-to-Gas

1. Introduction

In 2016, 195 parties (countries and regional groups) signed the Paris Agreement with the goal to keep the rise in global temperature below 2 °Celsius (better 1.5 °Celsius) by 2100. Under this Agreement, each party must plan and report on the contribution of containment of the carbon emissions. The ambitions must be increased every five years, where the parties' contribution are determined by the parties themselves and are called nationally determined contributions (NDCs). Using a statistical probabilistic model for forecasting industry carbon emissions, a study published by Nature in 2021 ([3]) shows, that the probability of staying below 2 °Celsius by 2100 is 26%, provided that all countries meet their NCD's and continue to reduce emissions at the same rate after 2030.

Figure 1.1 shows the probabilistic forecast of global mean temperature anomaly from 2015 to 2100 based on current trends. Here the black curve represents historical observations, the red curve represents the forecast median and the shaded area represents the 90% and 95% prediction interval. As we can see, the median temperature increase forecast for 2100 is approximately at 2.8 °Celsius. Thus, meeting all promises under the Paris Agreement and continuing to reduce emissions at the same rate will not satisfy the set degree target [3].

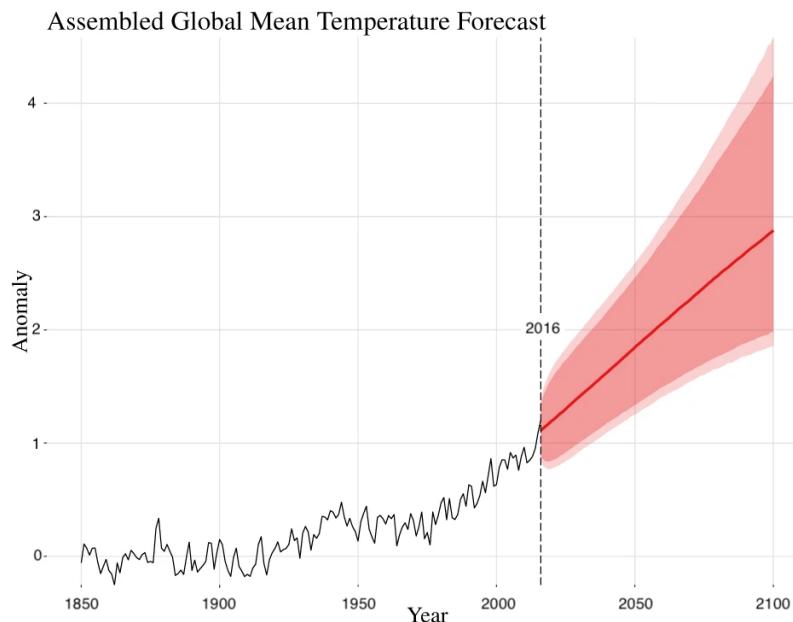


Figure 1.1.: Probabilistic forecast of the global temperature anomaly to 2100¹

¹ [3], p.3

The authors of the paper [3] state that in order to reach the 2 °Celsius target with 90% probability it would require quadrupling the annual rate of decline in emissions with the assumption reducing carbon emissions to 10% of the current level (global net zero emissions) by 2070. In addition, to have a chance to stay below 1.5 °Celsius (the actual aimed at target) with 90% probability would require multiplying the annual rate of decline in emissions by 30 and reaching global net zero by 2023. Staying below the 1.5 °Celsius thus turns out to be a difficult task, since the estimated temperature increase from the pre-industrial times to 2021 is already estimated to be at 1.1 °Celsius [3]. There is thus no time left to loose.

Certainly, the biggest approach to reduce carbon emissions is renewable energy. Renewable energy comes e.g. in form of sunlight, wind, geothermal heat and water. In comparison to coal or nature gas, the disadvantage of the renewable power sources is its volatile character. Sunshine and wind are weather phenomenona, that depend strongly on the day time, season and the region. A report by DLR (German Aerospace Center) shows the technical potential of renewable energy in different regions [4]. In Figure 1.2a we can see the annual average wind speed at 80 m above ground level in *m/s*. In Figure 1.2b the annual global irradiance on a surface in *kWh/m²* per *year* is shown. Considering Germany the pictures show that the largest potential for wind power lies in the north of Germany, whereas the solar radiation is the highest in the south of Germany.

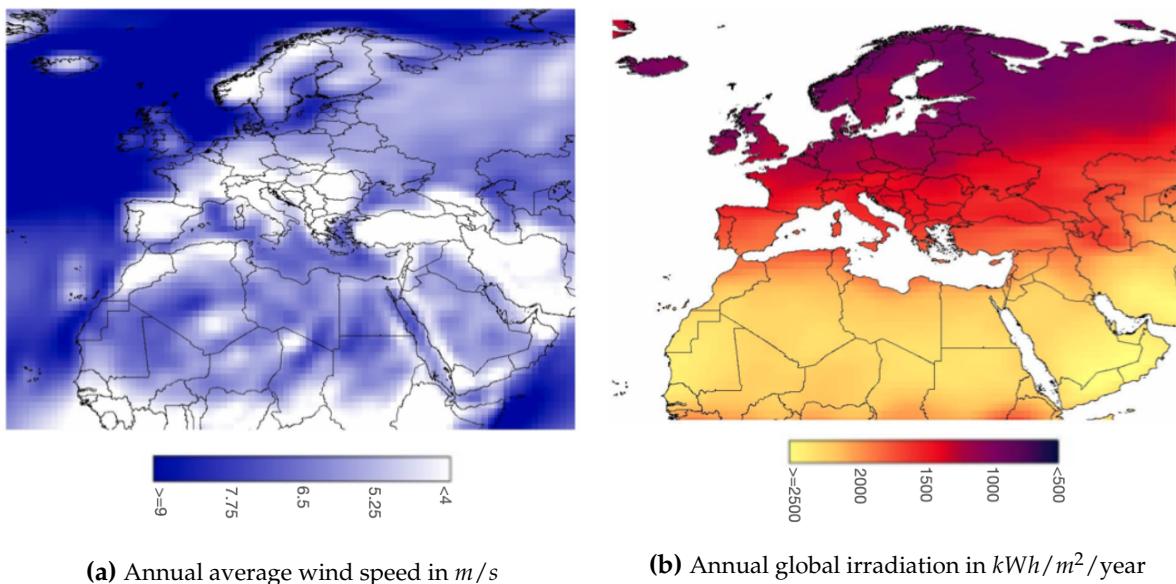


Figure 1.2: Technical area potentials for wind- and solar power generation ²

In order to provide reliable and stable power supply with renewable energy, a large-scale energy storage is required to compensate for weather variations and to save excess power. One of the most promising approach is converting electricity into hydrogen and vice versa, the so called „power-to-gas“ or „gas-to-power“ process. The key technology for „power-to-gas“ is

² [4], p.68-69

electrolysis, where electric energy is used to split water into hydrogen and oxygen.

Hydrogen is a favourable option for the future, as it does not contain any carbon and thus does not contribute to the greenhouse effect. The hydrogen produced in power-to-gas plants can be used to transport renewable energy over long distances. E.g. one could produce hydrogen out of photovoltaic plants in rich solar irradiation regions and transport it via gas distribution systems to regions with limited sun potential.

For transport underground, gas networks can be utilized. Most countries have a well-developed infrastructure, connecting the major supply sites with the consumers. The total length of natural gas pipe lines in Germany is 520,000 km, situated 1 m under the earth surface. These pipelines are connected to roughly 50 % of all German households. The maximum allowable concentration of H_2 is restricted to single digit percentage in order to ensure standard quality, which means that hydrogen has to be mixed with other natural gases like nitrogen, methane and argon [5]. To achieve concentrations greater than this restriction one would require significant major modifications to existing infrastructure and end-use applications [6].

Another important component of the gas network are compressors. Since pressure is decreasing throughout the gas-network, compressors are necessary for maintaining it.

In this thesis we are going to optimize a "gas-to-power" network while minimizing the compressors work. Chapter 2 introduces definitions and theorems that are necessary for further understanding of the model. In Chapter 3, we talk about an approach on how to solve Mixed-Integer Non-Linear (MINLP) Problem in combination with Partial Outer Convexification, the so called three-step method. Chapter 4 first introduce the Levenberg-Marquardt method which will be useful for solving our power-grid model. Afterwards we will introduce the branch-and-bound method as our comparison approach. In this Chapter we will also talk about the Interior Point Method - a method for solving nonlinear problems, which will be used for the three-step approach. In Chapter 5 we introduce physical characteristics of a gas-to-power network that will be necessary for constructing our optimization model. Chapter 6 provides a compressor cost optimization model and shows in which cases our model fulfills the error-minimizing theorem for the three-step approach. In Chapter 7 we optimize our power-to-gas model, where we first start with a simple model which contains only one compressor and later move on to a gas network model containing two compressors. For both model, we will also investigate how the compressors behave if additional constraints that couple over time are involved.

2. Mathematical Theory

2.1. Conservation and Balance Laws

First, we present the definition of a system of conservation laws in one dimensional case.

Definition 2.1 (System of n Conservation Equations ([7], p.6)). *Let $u = u(x, t) : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}^n$ be a function and $F = F(u) : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Then*

$$u_t + (F(u))_x = 0 \quad (2.1)$$

is a system of n conservation laws in \mathbb{R} with flux function F .

Next, we provide a definition of a hyperbolic system of conservation laws.

Definition 2.2 (Hyperbolic System ([7], p.6)). *The system*

$$u_t + (F(u))_x = 0 \quad (2.2)$$

is called hyperbolic, if the following holds: Let

$$A(u) := DF(u) = \left(\frac{\partial}{\partial u_k} F^{(j)}(u) \right)_{1 \leq j, k \leq n} \quad (2.3)$$

be the Jacobian matrix of $F(u)$. If for every $u \in \mathbb{R}^n$ and every $w \in \mathbb{R}, w \neq 0$, it holds that the matrix

$$A(u, w) = wA(u) \quad (2.4)$$

has n real eigenvalues $\lambda_1(u, w) \leq \dots \leq \lambda_n(u, w)$ and n corresponding linearly independent eigenvectors u_1, \dots, u_n , i.e.

$$A(u, w)u_k = \lambda_k u_k, \quad 1 \leq k \leq n \quad (2.5)$$

then the system is called hyperbolic. If additionally, the eigenvalues $\lambda_k(u, w)$ are all distinct, then the system in Equation 2.2 is called strictly hyperbolic.

If additionally a source term appears in the system, we call the system a *system of balance laws*. We provide again its definition in one-dimensional case.

Definition 2.3 (System of n Balance Laws ([8], p.375)). Let $u = u(x, t) : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}^n$ be a function, $\psi(u) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $F = F(u) : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Then

$$u_t + (F(u))_x = \psi(u) \quad (2.6)$$

is a system of n balance laws in \mathbb{R} with flux function F and source term $\psi(u)$.

2.2. Discretization Methods

2.2.1. Discretization Methods for Hyperbolic Systems of Conservation Laws

We will consider the linear advection equation in one space dimension with $u : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$, ($n = 1$)

$$u_t + au_x = 0, \quad -\infty < x < \infty, \quad t \geq 0, \quad (2.7)$$

where $a \in \mathbb{R}$ is a constant.

We discretize the $x - t$ space by choosing a mesh width Δx and a time step size Δt . We define the discrete grid points (x_j, t_i) by

$$x_j = j\Delta x, \quad j = \dots, -1, 0, 1, 2, \dots \quad (2.8)$$

$$t_i = i\Delta t, \quad i = 0, \dots, T \quad (2.9)$$

The discretization methods introduced in this section will produce approximations $u_j^i \in \mathbb{R}$ to the solution $u(x_j, t_i)$ at the mesh points.

First Order Upwind Method

We first assume $a \geq 0$ for Equation 2.7. For this case the First Order Upwind Method is formulated as follows

$$u_j^{i+1} = u_j^i - \frac{a\Delta x}{\Delta t}(u_j^i - u_{j-1}^i). \quad (2.10)$$

For $a < 0$, the First Order Upwind Method is taking data from the other direction

$$u_j^{i+1} = u_j^i - \frac{a\Delta x}{\Delta t}(u_{j+1}^i - u_j^i). \quad (2.11)$$

([9], p.109)

Lax-Friedrich Method

The Lax-Friedrich method takes the symmetric difference for the space derivative and the forward difference for the time derivative, but replaces u_j^i by $\frac{1}{2}(u_{j-1}^i + u_{j+1}^i)$.

For the advection equation in 2.7, the Lax-Friedrich method is formulated as follows

$$u_j^{i+1} = \frac{1}{2} (u_{j-1}^i + u_{j+1}^i) - a \frac{\Delta t}{2\Delta x} (u_{j+1}^i - u_{j-1}^i). \quad (2.12)$$

([9], p.104)

The Lax Friedrich Scheme is quite popular because it fulfills a theorem which states that the method converges against the entropy solution ([7], p.49). The Simple Upwind Scheme, on the other hand, can converge against a solution which is not the entropy solution ([7], p.33).

2.2.2. Discretization Methods for Hyperbolic Systems of Balance Laws

We will consider the linear advection similiar to Equation 2.7, but will add a source term to the right side

$$u_t + au_x = \psi(u), \quad -\infty < x < \infty, \quad t \geq 0. \quad (2.13)$$

Equation 2.13 becomes a *balance law*. The most basic approach is to add this source term to the numerical scheme. This basic approach was applied on the simple advection-reaction in LeVeque's *Finite Volume Methods for Hyperbolic Problems* ([8], p.378). The approach is denoted in general as

$$u_j^{i+1} = u_j^{scheme} + \Delta t \psi_j^i, \quad (2.14)$$

where ψ_j^i is an approximation of $\psi(u_j^i)$.

E.g. with the Simple Upwind Scheme we obtain

$$u_j^{i+1} = u_j^i - a \frac{\Delta x}{2\Delta t} (u_j^i - u_{j-1}^i) + \Delta t \psi_j^i. \quad (2.15)$$

2.2.3. CFL-Condition

The CFL-Condition is a grid size restriction equation. This condition was introduced by Courant, Friedrichs and Lewy in 1928 in the first papers on finite difference methods for PDEs. They recognized: A necessary (but not always sufficient) stability condition for any numerical method is that the domain of dependence of the difference method should include the domain of dependence of the PDE ([9], p.110).

Let's consider the linear advection equation $u_t + au_x = 0$. Using a numerical method, our information transport travels with speed $\Delta x / \Delta t$. The condition $a < \Delta x / \Delta t$ means, that the analytical information transport is slower than the numerical information transport, so the numerical method can transport information with speed a ([7], p.28). In other words, the numerical influence area has to contain the analytical area of influence. The condition can be also formulated for non-linear cases and its definition is provided in the following.

Definition 2.4 (CFL Condition ([7], p.28)). *The condition*

$$\left| \frac{\Delta t}{\Delta x} f'(v) \right| \leq 1, \quad \forall v \in [-\|u\|_\infty, \|u\|_\infty] \quad (2.16)$$

with u as the solution of the conservation law, is called CFL Condition. ([7], p.28)

The CFL Condition is a necessary but not always sufficient condition for convergence of the numerical solution towards the analytical solution. If the condition is violated, no convergence towards the analytical solution can be achieved.

2.3. Selected Types of Problems

Consider the following Problem:

$$\begin{aligned} \min \quad & f(x_1, \dots, x_n) \\ \text{subject to:} \quad & g_i(x_1, \dots, x_n) \leq 0 \quad \text{for } i \in I \\ & h_j(x_1, \dots, x_n) = 0 \quad \text{for } j \in J \\ & x \in X \end{aligned} \quad (2.17)$$

with $f, g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}$ and $X \subseteq \mathbb{R}^n$.

There are multiple ways of classifying optimization problems. They can be classified e.g. into convex and non-convex, constrained and unconstrained or smooth and non-smooth optimization problems. A convex optimization problem is a problem in which the objective function as well as the feasible set are convex. For convex optimization problems it holds that every local minimizer is also a global minimizer (see Appendix: Global and Local Minima). An optimization problem is called unconstrained if $I, J = \emptyset$.

In the following table some types of optimization problems are introduced, which will be mentioned in the following chapters. We restrict ourselves to continuous functions f, g_i, h_j and assume that at least one set I or J is a non-empty set.

LP	Linear Programming $f, g_i, h_j, \forall i \in I, \forall j \in J$ are linear functions and $X = \mathbb{R}^n$
QP	Quadratic Programming f is a quadratic function, $g_i, h_i, \forall i \in I, \forall j \in J$ are linear functions and $X = \mathbb{R}^n$
NLP	Non-Linear Programming at least one function in $\{f, \{g_i\}_{i \in I}, \{h_j\}_{j \in J}\}$ is non-linear and $X = \mathbb{R}^n$

Table 2.1.: Selected types of Problems (1)

MILP	Mixed-Integer Linear Programming $f, g_i, h_j, \forall i \in I, \forall j \in J$ are linear functions and $X = \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ with $n = n_1 + n_2; n_1, n_2 \in \mathbb{N}$
MINLP	Mixed-Integer Non-Linear Programming at least one function in $\{f, \{g_i\}_{i \in I}, \{h_j\}_{j \in J}\}$ is a non-linear function and $X = \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ with $n = n_1 + n_2; n_1, n_2 \in \mathbb{N}$
MIPDECO	Mixed-Integer PDE-Constrained Optimization at least one function in $\{f, \{g_i\}_{i \in I}, \{h_j\}_{j \in J}\}$ is a Partial Differential Equation and $X = \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ with $n = n_1 + n_2; n_1, n_2 \in \mathbb{N}$
MIOC	Mixed-Integer Optimal Control Programming problem constrained by a dynamical system with control variables and state variables in X with $X = \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$ with $n = n_1 + n_2; n_1, n_2 \in \mathbb{N}$
IP	(Pure) Integer Programming $f, g_i, h_j, \forall i \in I, \forall j \in J$ are linear functions and $X = \mathbb{Z}^n$

Table 2.2.: Selected types of Problems (2)

3. Partial Outer Convexification and Three-Step Method

The optimization problems of our gas-to-power network presented in Chapter 6 contain additional binary variables. These type of problem belong to the class of Mixed-Integer Optimal Control (MIOC) Problems. Furthermore our optimization problem is a PDE constrained model, thus it can also be classified as a Mixed-Integer-PDE constrained Optimization Problem (MIPDECO). After discretization, we obtain a large-scale Mixed-Integer Nonlinear Problem (MINLP). Handling nonlinear functions and integer values are the two main difficulties of an MINLP. Common approaches for solving MINLP are branch-and-bound strategies or multitree methods, e.g. outer approximation. These methods suffer from excessive computing times due to high number of integer variables. However, this high number is necessary since only a fine discretization in the PDEs provides an adequate representation of the dynamics ([10]). In this chapter we introduce a new method of optimization, which is based on *Partial Outer Convexification* (POC) reformulation. We will refer to this method as the **three-step method**.

For a discretized PDE-constrained optimization problem the partial outer convexification reformulation reads as follows:

$$\min_{u^t} f(x^t, u^t) \quad (3.1)$$

$$\text{subject to: } x^{t+1} = x^t + \Delta t \Phi(x^t, u^t) w^t \quad (3.2)$$

$$\sum_{s=1}^{n_{oc}} w_s^t = 1 \text{ with } w^t \in \{0, 1\}^{n_{oc}} \quad (3.3)$$

$$(+ \text{ some additional constraints for } x, u \text{ and } w) \quad (3.4)$$

where $f(x^t, u^t)$ is the objective function and constraint 3.2 is the discretized PDE with x as the state variable and u as the control variable. Constraint 3.3 is also known as the *Special Ordered Set of Type 1* (SOS1) condition. It is important to mention, that the term convexification might be misleading, since the resulting problem is only convex if also $f(\cdot)$ is convex in x and u . The expression *convexification* only addresses the integer component. Possible examples in real world scenarios for the problem above are traffic lights that show either red or green, valves that can be open or closed, or in our case, compressors that can either increase the pressure or leave it at the same level. If we consider n switches (e.g. n compressors, n valves etc.) without any further restrictions, we obtain $n_{oc} = 2^n$ possible switch combinations.

We consider an example with $n = 2$ switches. Without any further restrictions there are $n_{oc} = 2^2 = 4$ possible switch combinations, which are: $c_1 = [0, 0]$, $c_2 = [1, 0]$, $c_3 = [0, 1]$ and $c_4 = [1, 1]$. We let these vectors be located in the set $\Omega = \{c_1, c_2, c_3, c_4\}$. Control c_1 implies that both switches are off, control c_2 implies that the first switch is on and the second switch is off, control c_3 implies that the first switch is off and the second switch is on, and the last control c_4 implies that both switches are on. By introducing new variables $w^t \in \mathbb{R}^{n_{oc}}$, which fulfill the SOS1 condition 3.3 we ensure that only one of these controls can hold at time t . We let $w_p^t = 1$ if the combination c_p is active in t and $w_j^t = 0$ if combination $j \neq p$ is not active in t . In paper [1] these binary values w_s are also referred to as *convex multiplier controls*.

The reason we have introduced $w_1, \dots, w_{n_{oc}}$ in addition to the binary variables $c_1, \dots, c_{n_{oc}}$ is a Theorem for POC reformulated optimization problem, for which no equally strong relaxation results are known when regarding solely $c_1, \dots, c_{n_{oc}}$.

Based on the POC reformulation and a corresponding Theorem a three-step method was developed for solving MINLP Problems restricted by a PDE, which was discretized with a explicit method. The general idea of this approach is to approximate the solution of the MINLP by first solving its NLP-relaxation and afterwards reconstructing the binary variables by a Mixed-Integer Linear Problem (MILP) called Combinatorial Intergral Approximation Problem (CIAP). This approach was first introduced in the field of optimal control problems with ODE contraints by S. Sager et al. in 2005 [11]. It was further developed and applied on optimization problems constrained by PDEs for traffic light optimization in road networks [2] and transmission lines [1].

Algorithm 1 Three-Step Approach ([1], p.440)

Discretize problem 3.1 - 3.4 with appropriate step sizes Δt and Δx fulfilling the CFL-condition.

Step 1: Relax the integrality conditions

$$w_c(t_e) \in \{0, 1\} \rightarrow \bar{w}_c(t_e) \in [0, 1], \forall e \in \{0, \dots, n_T\}, \forall c \in \Omega \quad (3.5)$$

and solve the arising continuous NLP. This yields the optimal vector \bar{w} .

Step 2: Compute a feasible binary solution w out of \bar{w} by solving CIAP.

Step 3: Simulate the dynamics again with w to obtain a feasible trajectory and a corresponding objective value.

The three-step approach is shown in Algorithm 1. The advantage of this scheme is a theorem which yields an upper bound on the error between the trajectories of the relaxed problem and those of the solution obtained by Step 2 and Step 3 of Algorithm 1, if no additional constraints that couple over time are present.

For this theorem we first introduce

$$H = \left\{ \alpha \in \mathbb{R}_{\geq 0}^{n_{oc}} \mid \sum_{i=1}^{n_{oc}} \alpha_i = 1 \right\} \quad (3.6)$$

as the set of all relaxed controls that fulfill the SOS type 1 conditions. Furthermore we need two

norms: $\|\cdot\|_X : \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ and $\|\cdot\|_\Omega : \mathbb{R}^{n_{oc}} \rightarrow \mathbb{R}_{\geq 0}$.

Theorem 3.1 (Approximation Theorem for POC ([2], B62-B63)). *Let $\mathcal{I} = \{0, \dots, n_T - 1\}$, $\mathcal{D} \subseteq \mathbb{R}^N$ and $\Phi : \mathcal{I} \times \mathcal{D} \rightarrow \mathbb{R}^{n \times n_{oc}}$ be a matrix-valued function that is continuous with respect to the second argument and satisfies*

$$\|\Phi(t, \mu)v\|_X \leq M_{oc}\|v\|_\Omega, \quad \forall t \in \mathcal{I}, \mu \in \mathcal{D}, v \in \mathbb{R}^{n_{oc}} \quad (3.7)$$

$$\|(\Phi(t, \mu) - \Phi(t, \eta))\alpha\|_X \leq L_{oc}\|\mu - \eta\|_X, \quad \forall t \in \mathcal{I}, \mu, \eta \in \mathcal{D}, \alpha \in H \quad (3.8)$$

for constants $M_{oc}, L_{oc} < \infty$. Furthermore, for each $t \in \mathcal{I}$ let $h_t > 0$, $\alpha^t, \beta^t \in H$ such that $T = \sum_{t=0}^{n_T-1} h_t$ and for each $t \in \mathcal{I} \cup \{n_T\}$ let $\mu^t, \eta^t \in \mathcal{D}$ be given such that for all $t \in \mathcal{I}$

$$\mu^{t+1} = \mu^t + h_t\Phi(t, \mu^t)\alpha^t \quad \text{and} \quad \eta^{t+1} = \eta^t + h_t\Phi(t, \eta^t)\beta^t. \quad (3.9)$$

If for some set $\mathcal{I}' \subseteq \mathcal{I}$, constants $C_{oc}, \epsilon < \infty$ and some vector $\delta^0 \in \mathbb{R}^{n_{oc}}$ it holds that

$$\|(\Phi(t+1, \mu^{t+1}) - \Phi(t, \mu^t))v\|_X \leq h_t C_{oc}\|v\|_\Omega, \quad \forall t \in \mathcal{I}', v \in \mathbb{R}^{n_{oc}} \quad (3.10)$$

$$\|\delta^0 + \sum_{t=0}^{k-1} h_t(\alpha^t - \beta^t)\|_\Omega \leq \epsilon, \quad \forall k \in \mathcal{I} \cup \{n_T\} \quad (3.11)$$

then it follows with $T'_k = k \max\{h_t | t = 0, \dots, k-1\}$ and $n_{jump} = |\mathcal{I} \setminus (\mathcal{I}' \cup \{n_T - 1\})|$ that for all $k \in \mathcal{I} \cup \{n_T\}$

$$\sum_{t=0}^k h_t \|\mu^t - \eta^t\|_X \leq \frac{\exp(T'_k L_{oc}) - 1}{L_{oc}} (\|\mu^0 - \eta^0\|_X + (2M_{oc}(1 + n_{jump}) + T C_{oc})\epsilon). \quad (3.12)$$

Proof. For $k \in \mathcal{I} \cup \{n_T\}$, we define

$$e^k = \mu^k - \eta^k \quad \text{and} \quad \delta^k = \delta^0 + \sum_{t=0}^{k-1} h_t(\alpha^t - \beta^t). \quad (3.13)$$

We then obtain

$$\|e^k\|_X \stackrel{3.13}{=} \left\| \mu^k - \eta^k \right\|_X \quad (3.14)$$

$$\stackrel{3.13}{=} \left\| \mu^0 + \sum_{t=0}^{k-1} h_t \Phi(t, \mu^t) \alpha^t - \eta^0 - \sum_{t=0}^{k-1} h_t \Phi(t, \eta^t) \beta^t \right\| \quad (3.15)$$

$$= \left\| (\mu^0 - \eta^0) + \sum_{t=0}^{k-1} h_t \Phi(t, \mu^t) (\alpha^t - \beta^t) + \sum_{t=0}^{k-1} h_t (\Phi(t, \mu^t) - \Phi(t, \eta^t)) \beta^t \right\|_X \quad (3.16)$$

$$\leq \|e^0\|_X + \left\| \sum_{t=0}^{k-1} h_t \Phi(t, \mu^t) (\alpha^t - \beta^t) \right\|_X + \left\| \sum_{t=0}^{k-1} h_t (\Phi(t, \mu^t) - \Phi(t, \eta^t)) \beta^t \right\|_X \quad (3.17)$$

For the next steps we will deduce the following summation by parts or Abel transformation.

Suppose $\{f_k\}$ and $\{g_k\}$ are two sequences. Then it holds,

$$\begin{aligned} \sum_{k=m}^n f_k(g_{k+1} - g_k) &= \sum_{k=m}^n f_k g_{k+1} - \sum_{k=m}^n f_k g_k \\ &= f_n g_{n+1} - f_m g_m + \sum_{k=m}^{n-1} f_k g_{k+1} - \sum_{k=m+1}^n f_k g_k \\ &\stackrel{(*)}{=} (f_n g_{n+1} - f_m g_m) + \sum_{k=m}^{n-1} f_k g_{k+1} - \sum_{k=m}^{n-1} f_{k+1} g_{k+1} \\ &= (f_n g_{n+1} - f_m g_m) - \sum_{k=m}^{n-1} g_{k+1} (f_{k+1} - f_k) \end{aligned}$$

where we use index shift in (*). Besides Abels summation, we will use the result

$$\delta^{t+1} - \delta^t = \delta^0 + \sum_{k=0}^t h_k(\alpha^k - \beta^k) - \delta^0 - \sum_{k=0}^{t-1} h_k(\alpha^k - \beta^k) = h_t(\alpha^t - \beta^t). \quad (3.18)$$

as well as Equations 3.7, 3.10 and 3.11 to bound the middle term in 3.17:

$$\left\| \sum_{t=0}^{k-1} \Phi(t, \mu^t) h_t(\alpha^t - \beta^t) \right\|_X \stackrel{3.18}{=} \left\| \sum_{t=0}^{k-1} \Phi(t, \mu^t) (\delta^{t+1} - \delta^t) \right\|_X \quad (3.19)$$

$$\stackrel{\text{Abel}}{=} \left\| \Phi(k-1, \mu^{k-1}) \delta^k - \Phi(0, \mu^0) \delta^0 - \sum_{t=0}^{k-2} [\Phi(t+1, \mu^{t+1}) - \Phi(t, \mu^t)] \delta^{t+1} \right\|_X \quad (3.20)$$

$$\leq \underbrace{\left\| \Phi(k-1, \mu^{k-1}) \delta^k \right\|_X}_{\leq 2M_{oc}\epsilon} + \underbrace{\left\| \Phi(0, \mu^0) \delta^0 \right\|_X}_{\leq (TC_{oc} + 2M_{oc}n_{jump})\epsilon} + \underbrace{\sum_{t=0}^{k-2} \left\| [\Phi(t+1, \mu^{t+1}) - \Phi(t, \mu^t)] \delta^{t+1} \right\|_X}_{\leq (TC_{oc} + 2M_{oc}n_{jump})\epsilon} \quad (3.21)$$

$$\leq (2M_{oc} + TC_{oc} + 2M_{oc}n_{jump})\epsilon \quad (3.22)$$

In the last term of Equation 3.21, we split up the sum over indices in \mathcal{I}' (sum $\leq TC_{oc}\epsilon$) and indices not in \mathcal{I}' (sum $\leq 2M_{oc}n_{jump}\epsilon$). n_{jump} denotes the number of variables which are in \mathcal{I} but not in \mathcal{I}' .

Now we have to provide an upper bound for the last term of Equation 3.17. Using the triangle inequality for norms and Equation 3.8 the term can be bounded as follows

$$\begin{aligned} \left\| \sum_{t=0}^{k-1} h_t(\Phi(t, \mu^t) - \Phi(t, \eta^t)) \beta^t \right\|_X &\leq \sum_{t=0}^{k-1} h_t \|(\Phi(t, \mu^t) - \Phi(t, \eta^t)) \beta^t\|_X \\ &\stackrel{3.8}{\leq} \sum_{t=0}^{k-1} h_t L_{oc} \|\mu^t - \eta^t\|_X \\ &= L_{oc} \sum_{t=0}^{k-1} h_t \|e^t\|_X \end{aligned}$$

All in all we obtain

$$\|e^k\|_X \leq \|e^0\|_X + (2M_{oc}(1 + n_{jump}) + TC_{oc})\epsilon + L_{oc} \sum_{t=0}^{k-1} h_t \|e^t\|_X. \quad (3.23)$$

By defining $B^k = \sum_{t=0}^{k-1} h_t \|e^t\|_X$, we can rewrite Equation 3.23 into

$$\|e^k\|_X = \frac{B^{k+1} - B^k}{h_k} \leq \|e^0\|_X + (2M_{oc}(1 + n_{jump}) + TC_{oc})\epsilon + L_{oc} B^k.$$

With $h_k^{\max} = \max\{h_t | t = 0, \dots, k\}$ this implies for all $i \leq k$ that

$$B^{i+1} \leq h_k^{\max} \|e^0\|_X + h_k^{\max} (2M_{oc}(1 + n_{jump}) + TC_{oc})\epsilon + (1 + h_k^{\max} L_{oc}) B^i.$$

Using a discrete version of Gronwall's inequality (see Appendix: Gronwall Lemma) and $B^0 = 0$, we can deduce

$$B^{k+1} \leq \frac{\exp(L_{oc}kh_k^{\max}) - 1}{L_{oc}} (\|e^0\|_X + (2M_{oc}(1 + n_{jump}) + TC_{oc})\epsilon).$$

It follows with $T'_k = k \max\{h_t | t = 0, \dots, k-1\}$

$$\sum_{t=0}^k h_t \|e^t\|_X \leq \frac{\exp(T'_k L_{oc}) - 1}{L_{oc}} (\|e^0\|_X + (2M_{oc}(1 + n_{jump}) + TC_{oc})\epsilon).$$

■

More precisely, Theorem 3.1 provides an upper boundary on the mismatch of the trajectories obtained in the NLP Relaxation in Algorithm 1, Step 1 and the trajectories obtained in Step 2 and Step 3. We can decrease this mismatch in Equation 3.12 by decreasing ϵ . In CIAP we exploit this property and minimize ϵ over all binary feasible convex multipliers. The CIAP reads as follows ([2], B64):

$$\begin{aligned} & \min_{\substack{\beta^t \in H \cap \{0,1\}^{n_{oc}}, \\ t=0, \dots, n_T-1, \\ \delta^0 \in \mathbb{R}^{n_{oc}}, \epsilon \in \mathbb{R}_{\geq 0}}} \epsilon \\ & \text{subject to: } \left\| \delta^0 + \sum_{t=0}^{k-1} \Delta t (\alpha^t - \beta^t) \right\|_{\Omega} \leq \epsilon \quad \text{for all } k = 0, \dots, n_T \end{aligned} \quad (3.24)$$

where α^t are the relaxed binary variables in time t of the NLP solution. In next section we introduce a rounding strategy for CIAP. If no additional constraints that couple over time are present, we can solve CIAP using *Sum Up Rounding*. We will see that in this case for Theorem 3.1 it means that we can decrease the error or mismatch in Equation 3.12 by decreasing Δt .

3.1. Rounding Strategies for CIAP

In the following, we will introduce a method on how to solve CIAP in 3.24 if no additional constraints that couple over time are present. This method was mentioned by Sebastian Sager in *Numerical methods for mixed-integer optimal control problems* ([11], p.88) for SOS type 1 conditions.

Consider given measurable functions¹ $\alpha_j : [0, t_m] \rightarrow [0, 1]$ with $j = 1, \dots, n_{oc}$ and time grid $0 < t_1 < \dots < t_m$, which fulfills the SOS1 condition

$$\sum_{j=1}^{n_{oc}} \alpha_j(t) = 1, \quad t \in [0, t_m] \quad (3.25)$$

and on which we approximate the control $\alpha(\cdot) : [0, t_m] \rightarrow [0, 1]^{n_{oc}}$. We define $\Delta t_i := t_{i+1} - t_i$ and Δt as the maximum distance

$$\Delta t := \max_{i=0, \dots, m-1} \Delta t_i. \quad (3.26)$$

Let a function $\beta(\cdot) : [0, t_m] \rightarrow \{0, 1\}^{n_{oc}}$ be defined as

$$\beta_j(t) = p_{j,i}, \quad t \in [t_i, t_{i+1}) \quad (3.27)$$

where $p_{j,i}$ are binary values given by

$$\begin{aligned} \hat{p}_{j,i} &= \int_0^{t_{i+1}} \alpha_j(\tau) d\tau - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \\ p_{j,i} &= \begin{cases} 1, & \text{if } \hat{p}_{j,i} \geq \hat{p}_{k,i}, \forall k \neq j \text{ and } j < k \forall k : \hat{p}_{j,i} = \hat{p}_{k,i} \\ 0, & \text{else} \end{cases} \end{aligned} \quad (3.28)$$

For Equation 3.25 - 3.27 the following Theorem can then be formulated.

Theorem 3.2 (Approximation by Sum Up Rounding ([10], p.12)). *Let a measurable function $\alpha : [0, t_m] \rightarrow [0, 1]^{n_{oc}}$ fulfill 3.25 and a function $\beta : [0, t_m] \rightarrow \{0, 1\}^{n_{oc}}$ defined by 3.27 and 3.28 be given for $n_{oc} \geq 2$. Then it holds*

$$\left\| \int_0^t \alpha(\tau) - \beta(\tau) d\tau \right\| \leq (n_{oc} - 1) \Delta t \quad (3.29)$$

where $\beta(\cdot)$ also fulfills the SOS1 property (compare 3.25).

Proof. Since both function $\alpha(\cdot)$ and $\beta(\cdot)$ fulfill the SOS1 type condition, it implies

$$\sum_{j=1}^{n_{oc}} \int_0^t \alpha_j(\tau) - \beta_j(\tau) d\tau = \int_0^t \sum_{j=1}^{n_{oc}} (\alpha_j(\tau) - \beta_j(\tau)) d\tau = 0, \quad \forall t \in [0, t_m]. \quad (3.30)$$

1 Let $(\Omega_1, \mathcal{A}_1)$ and $(\Omega_2, \mathcal{A}_2)$ be measurable spaces. Let $f : \Omega_1 \rightarrow \Omega_2$ be a function that satisfies $f^{-1}(A_2) \in \mathcal{A}_1$ for each $A_2 \in \mathcal{A}_2$. Then we say that f is $\mathcal{A}_1 - \mathcal{A}_2$ measurable or simply measurable (if the σ -algebra are understood by context).

We restrict our proof to one fixed time step $t = t_r$. According to (3.27) we can discretize $\beta(\cdot)$ as follows

$$\int_0^{t_r} \beta_j(\tau) d\tau = \sum_{i=0}^{r-1} p_{j,i} \Delta t_i. \quad (3.31)$$

We further define k as the index for which $\left| \int_0^{t_r} \alpha_j(\tau) d\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i \right|$ takes its largest value

$$k = \arg \max_{j=1 \dots n_{oc}} \left| \int_0^{t_r} \alpha_j(\tau) d\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i \right|. \quad (3.32)$$

We will proof this theorem by contradiction and formulate the assumption, that there exists an $r \in \{0, \dots, m\}$ such that

$$\left| \int_0^{t_r} \alpha_k(\tau) d\tau - \sum_{i=0}^{r-1} p_{k,i} \Delta t_i \right| > (n_{oc} - 1) \Delta t. \quad (3.33)$$

We distinguish two cases. Our first assumption, which we want to contradict is

$$\int_0^{t_r} \alpha_k(\tau) d\tau - \sum_{i=0}^{r-1} p_{k,i} \Delta t_i < -(n_{oc} - 1) \Delta t. \quad (3.34)$$

Define \hat{i} as the highest index for control k for which $p_{k,\hat{i}} = 1$ holds:

$$\hat{i} = \arg \max_{0 \leq i \leq r-1} \{i : p_{k,i} = 1 \text{ and } p_{k,l} = 0 \forall l : i < l \leq r-1\}. \quad (3.35)$$

With assumption 3.34 it holds

$$\begin{aligned} \sum_{i=0}^{\hat{i}} p_{k,i} \Delta t_i &= \sum_{i=0}^{r-1} p_{k,i} \Delta t_i > \int_0^{t_r} \alpha_k(\tau) d\tau + (n_{oc} - 1) \Delta t \\ &\geq \int_0^{t_{\hat{i}+1}} \alpha_k(\tau) d\tau + (n_{oc} - 1) \Delta t \end{aligned} \quad (3.36)$$

or by reformulating 3.36

$$\int_0^{t_{\hat{i}+1}} \alpha_k(\tau) d\tau - \sum_{i=0}^{\hat{i}} p_{k,i} \Delta t_i < -(n_{oc} - 1) \Delta t. \quad (3.37)$$

According to 3.32, the difference on the left side takes its maximum value at index k , so it also holds for arbitrary $j = 1, \dots, n_{oc}$:

$$\int_0^{t_{\hat{i}+1}} \alpha_j(\tau) d\tau - \sum_{i=0}^{\hat{i}} p_{j,i} \Delta t_i < -(n_{oc} - 1) \Delta t. \quad (3.38)$$

Thus this also has to hold for the sum of all $j = 1, \dots, n_{oc}$. Summing up over all controls j on both

sides we obtain

$$\underbrace{\sum_{j=1}^{n_{oc}} \left(\int_0^{t_{i+1}} \alpha_j(\tau) d\tau - \sum_{i=0}^{\hat{i}} p_{j,i} \Delta t_i \right)}_{=0 \text{ (3.30)}} < -n_{oc} (n_{oc} - 1) \Delta t. \quad (3.39)$$

With Equation 3.30 we receive the contradiction $0 < n_{oc} - n_{oc}^2$ with $n_{oc} \geq 2$.

Next we assume

$$\int_0^{t_r} \alpha_k(\tau) d\tau - \sum_{i=0}^{r-1} p_{k,i} \Delta t_i > (n_{oc} - 1) \Delta t. \quad (3.40)$$

In Equation 3.30 we take index k out of the integral

$$\sum_{1=j \neq k}^{n_{oc}} \left(\int_0^{t_r} \alpha_j(\tau) d\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i \right) + \int_0^{t_r} \alpha_k(\tau) d\tau - \sum_{i=0}^{r-1} p_{k,i} \Delta t_i = 0. \quad (3.41)$$

We can approximate the last two terms on the left side with Equation 3.40

$$\sum_{1=j \neq k}^{n_{oc}} \left(\int_0^{t_r} \alpha_j(\tau) d\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i \right) + (n_{oc} - 1) \Delta t < 0. \quad (3.42)$$

We rewrite the left side of Equation 3.42 as the sum of $n_{oc} - 1$ terms

$$\int_0^{t_r} \alpha_j(\tau) d\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i + \Delta t. \quad (3.43)$$

At least one of the terms $\int_0^{t_r} \alpha_j(\tau) d\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i + \Delta t$ has to be negative, thus there exists an index \hat{j} such that

$$\int_0^{t_r} \alpha_{\hat{j}}(\tau) d\tau - \sum_{i=0}^{r-1} p_{\hat{j},i} \Delta t_i + \Delta t < 0. \quad (3.44)$$

Define \hat{i} as the highest index for control \hat{j} for which $p_{\hat{j},\hat{i}} = 1$ holds

$$\hat{i} = \arg \max_{0 \leq i \leq r-1} \left\{ i : p_{\hat{j},i} = 1 \text{ and } p_{\hat{j},l} = 0 \forall l : i < l \leq r-1 \right\}. \quad (3.45)$$

Then it holds

$$\begin{aligned} \int_0^{t_{\hat{i}+1}} \alpha_{\hat{j}}(\tau) d\tau - \sum_{i=0}^{\hat{i}-1} p_{\hat{j},i} \Delta t_i &\leq \int_0^{t_r} \alpha_{\hat{j}}(\tau) d\tau - \sum_{i=0}^{r-1} p_{\hat{j},i} \Delta t_i \\ &\leq \Delta t + \int_0^{t_r} \alpha_{\hat{j}}(\tau) d\tau - \sum_{i=0}^{r-1} p_{\hat{j},i} \Delta t_i \stackrel{3.44}{<} 0. \end{aligned} \quad (3.46)$$

At time \hat{i} the following equation must hold because of the rounding strategies in 3.28

$$\hat{j} = \arg \max_{1 \leq j \leq n_{oc}} \left\{ \int_0^{\hat{t}_{i+1}} \alpha_j(\tau) d\tau - \sum_{i=0}^{\hat{i}-1} p_{j,i} \Delta t_i \right\}. \quad (3.47)$$

Since Equation 3.46 holds for \hat{j} , it also holds for arbitrary $j = 1, \dots, n_{oc}$ and we obtain

$$\int_0^{\hat{t}_{i+1}} \alpha_j(\tau) d\tau - \sum_{i=0}^{\hat{i}} p_{j,i} \Delta t_i \leq \int_0^{\hat{t}_{i+1}} \alpha_j(\tau) d\tau - \sum_{i=0}^{\hat{i}-1} p_{j,i} \Delta t_i < 0. \quad (3.48)$$

This is a contradiction to our assumption in 3.40. \blacksquare

In this Theorem we see that the difference between $\alpha(\cdot)$ and $\beta(\cdot)$ is smaller, the smaller Δt is chosen. We can transfer this result to the MILP Problem CIAP in 3.24, concluding that we can diminish ϵ by decreasing Δt .

3.1.1. Sum Up Rounding Strategy with SOS1

The Sum Up Rounding Startegy in a discretized version is denoted as follows

$$\begin{aligned} \hat{p}_{j,i} &= \sum_{k=0}^i \tilde{p}_{j,k} - \sum_{k=0}^{i-1} p_{j,k} \\ p_{j,i} &= \begin{cases} 1, & \text{if } \hat{p}_{j,i} \geq \hat{p}_{k,i}, \forall k \neq j \text{ and } j < k \ \forall k : \hat{p}_{j,i} = \hat{p}_{k,i} \\ 0, & \text{else} \end{cases} \end{aligned} \quad (3.49)$$

where $\tilde{p}_{j,k} \in [0, 1]$ denote the relaxed solutions for $j = 1, \dots, n_{oc}$ and $p_{j,i} \in \{0, 1\}$ are the binary solutions. The calculated values $p_{j,i}$ can then be used in Step 3, Algorithm 1 for reconstructing the MINLP solution. Another strategy for solving CIAP with SOS1 property, which was also proposed in [11], is the *Standard Rounding Startegy*. In this work, however, we will only work with the Sum Up Rounding Strategy presented in Equations 3.49.

In the following, we will provide an example for the Sum Up Rounding Strategy with $n_{oc} = 2$. Assume that our NLP relaxation in Step 1 has provided us with the continuous values $\tilde{p}_{j,i}$ given in Table 3.1. These values fulfill the SOS1 constraint.

time i	0	1	2	3	4	5	6	7
$\tilde{p}_{1,i}$	0	0.5	0.25	0.1	0.9	0.6	0.33	0.8
$\tilde{p}_{2,i}$	1	0.5	0.75	0.9	0.1	0.4	0.67	0.2

Table 3.1.: Example: Continuous Values $\tilde{p}_{j,i}$

With the Sum Up Rounding Strategy we obtain discrete values for every time step i given in Table 3.2.

time i	0	1	2	3	4	5	6	7
$p_{1,i}$	0	1	0	0	1	0	1	0
$p_{2,i}$	1	0	1	1	0	1	0	1

Table 3.2.: Example: Discrete Values $p_{j,i}$ obtained from $\tilde{p}_{j,i}$ via Sum Up Rounding Strategy

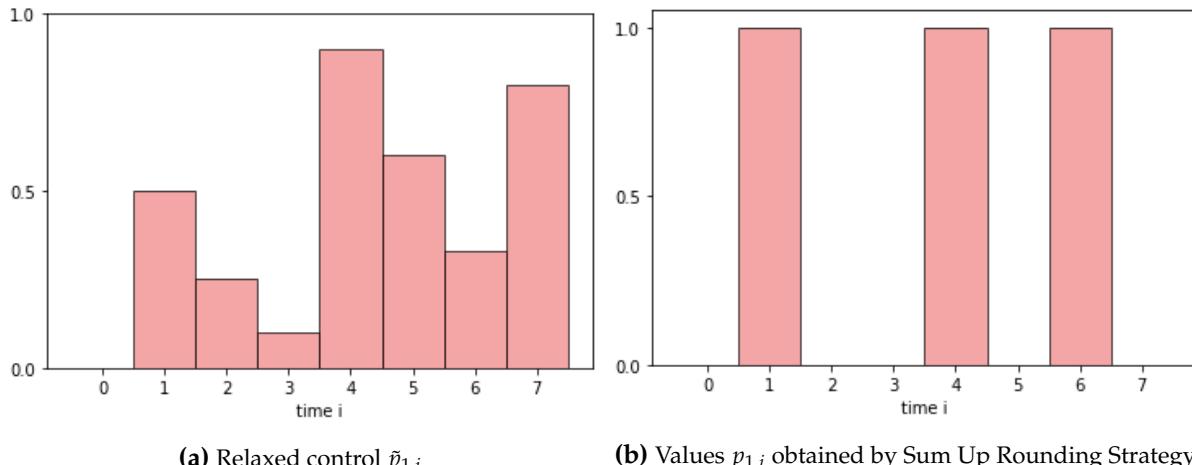
For further clarification, let us calculate $p_{i,j}$ for control $j = 1$ and time $i = 4$. With Equations 3.49 we obtain

$$\hat{p}_{1,4} = \sum_{k=0}^4 \tilde{p}_{1,k} - \sum_{k=0}^3 p_{1,k} = (0 + 0.5 + 0.25 + 0.1 + 0.9) - (0 + 1 + 0 + 0) = 0.75$$

and

$$\hat{p}_{2,4} = \sum_{k=0}^4 \tilde{p}_{2,k} - \sum_{k=0}^3 p_{2,k} = (1 + 0.5 + 0.75 + 0.9 + 0.1) - (1 + 0 + 1 + 1) = 0.25$$

Since $\hat{p}_{1,4} \geq \hat{p}_{2,4}$, we have $p_{1,4} = 1$ and $p_{2,4} = 0$ (compare Table 3.2). The figure below demonstrates graphically the results of the Sum Up Rounding Strategy in our example.



4. Numerical Methods

4.1. Methods for Solving a System of Nonlinear Equations

4.1.1. Newton-Raphson Method

We consider a system of nonlinear equations

$$F(x) = 0 \quad (4.1)$$

with $F \in C^1(\mathbb{R}^n), F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ where $F'(x)$ is the Jacobian matrix of F . One approach for finding a solution x^* of Equation 4.1 is the **Newton-Raphson Method**. If F is continuous and the Jacobian matrix $F'(x)$ exists, then we can apply first-order Taylor series expansion

$$F(x_k + d) = F(x_k) + F'(x_k)d + o(\|d\|_2^2). \quad (4.2)$$

Given a point x_k , we can determine x_{k+1} as follows

$$x_{k+1} = x_k + d_k \quad (4.3)$$

where d_k is calculated solving equation

$$F'(x_k)d_k = -F(x_k). \quad (4.4)$$

Here Equation 4.4 is derived from the Taylor series expansion in Equation 4.2 by omitting the term $o(\|d\|_2^2)$.

The calculation of the Jacobian matrix $F'(x_k)$ can be expensive. On the other hand, the Jacobian might also be singular or the Newton direction d_k may not be a descent direction, because the Jacobian matrix is not positive definite. A variety of simplifications of the Newton-Raphson Method were developed to solve the issues mentioned above. One variant is the **Newton-like Method**

$$M_k d_k = -F(x_k) \quad \text{with } x_{k+1} = x_k + d_k \quad (4.5)$$

where M_k is a 'sufficiently accurate' approximation of $F'(x_k)$.

However, the Newton-Raphson Method presented in 4.4 does not converge for arbitrary starting points x_0 . If the initial guess x_0 is not sufficiently close to x^* it might happen, that there is no convergence towards x^* . The Newton-Raphson method is therefore also called *local Newton*

Algorithm 2 Newton Method for optimization problems [12], p.25

Input: Starting point $x_0 \in \mathbb{R}^n$.

```

for  $k = 0, 1, 2, \dots$  do
    if  $F'(x_k) = 0$  then
        Stop
    end if

```

Compute Newton step $d_k \in \mathbb{R}^n$ by solving

$$F'(x_k)d_k = -F(x_k)$$

```

    Set  $x_{k+1} = x_k + d_k$ 
end for

```

Method. It is presented in Algorithm 2.

In the following we will take a closer look at a particular globalization concept, the **Levenberg-Marquardt Method**, that copes with bad guesses.

4.1.2. Levenberg-Marquardt Method

We can state, that finding a root of Equation 4.1 is the same as solving the following minimization problem:

$$\min_{x \in \mathbb{R}^n} \|F(x)\|_2^2 = \min_{x \in \mathbb{R}^n} F(x)^T F(x). \quad (4.6)$$

We can again apply a first-order Taylor expansion to Equation 4.6 at point x_k

$$\min_{d_k \in \mathbb{R}^n} \|F(x_k + d_k)\|_2^2 \approx \min_{d_k \in \mathbb{R}^n} \|F(x_k) + F'(x_k)d_k\|_2^2. \quad (4.7)$$

The first-order Taylor expansion is sufficiently good for small areas around x_k or analogue for small d_k .

The idea behind the Levenberg-Marquardt Method is to choose d_k such that the minimum of Equation 4.7 is as small as possible, while keeping d_k small as well in order to ensure that the Taylor series expansion is a good approximation for $F(x)$. We formulate the minimization problem combining both aspects as follows

$$\min_{d_k \in \mathbb{R}^n} \|F(x_k) + F'(x_k)d_k\|_2^2 + p\|\Delta d_k\|_2^2, \quad (4.8)$$

where $p \geq 0$ is a Lagrange multiplier, which controls the importance of minimizing $\|F(x_k) + F'(x_k)d_k\|_2^2$ versus constraining the size of d_k . If we choose p to be large, we penalize large d_k more and the minimization problem tries to keep d_k small. We can rewrite Equation 4.8 as the euclidian norm of one vector [13]

$$\min_{d_k \in \mathbb{R}^n} \left\| \begin{pmatrix} F'(x_k) \\ \sqrt{p}I_{n \times n} \end{pmatrix} d_k + \begin{pmatrix} F(x_k) \\ 0 \end{pmatrix} \right\|_2^2. \quad (4.9)$$

Using the Normal Equation we obtain

$$\begin{pmatrix} F'(x_k)^T & \sqrt{p}I_{n \times n} \end{pmatrix} \begin{pmatrix} F'(x_k) \\ \sqrt{p}I_{n \times n} \end{pmatrix} d_k = -\begin{pmatrix} F'(x_k)^T & \sqrt{p}I_{n \times n} \end{pmatrix} \begin{pmatrix} F(x_k) \\ 0 \end{pmatrix}. \quad (4.10)$$

In Equation 4.10 we multiply out all vectors and end up with the *Levenberg-Marquardt Method*, which reads as follows

$$(F'(x_k)^T F'(x_k) + pI_{n \times n})d_k = -F'(x_k)^T F(x_k) \quad \text{with} \quad x_{k+1} = x_k + d_k. \quad (4.11)$$

Solving for d_k we obtain

$$d_k = -(F'(x_k)^T F'(x_k) + pI_{n \times n})^{-1} F'(x_k)^T F(x_k) \quad \text{with} \quad x_{k+1} = x_k + d_k. \quad (4.12)$$

In comparison to the previously discussed Newton-Raphson Method the arising matrix $(F'(x_k)^T F'(x_k) + pI_{n \times n})^{-1}$ in the Levenberg-Marquardt Method is positive definite and is thus always invertible for $p > 0$.

We will take a further look at two interesting cases in the limit for $d_k(p)$:

$$p \rightarrow 0^+ : \quad d_k(0) = -F'(x_k)^{-1} F(x_k) \quad (4.13)$$

$$p \rightarrow \infty : \quad d_k(p) \rightarrow -\frac{1}{p} F'(x_k)^T F(x_k). \quad (4.14)$$

For $p \rightarrow 0^+$ the method becomes the ordinary Newton-Method, for $p \rightarrow \infty$ it turns into a steepest descent method ([14], p.118).

In order to evaluate our choice of p , we calculate the ratio of predicted reduction to the actual reduction by defining ρ_k :

$$\rho_k = \frac{\|F(x_k)\|_2^2 - \|F(x_k) + F'(x_k)(x_{k+1} - x_k)\|_2^2}{\|F(x_k)\|_2^2 - \|F(x_{k+1})\|_2^2}. \quad (4.15)$$

If our prediction was not good enough (ρ_k is below a certain threshold β_0), we increase the Lagrange multiplier p (here: $p := 10p$) in order to diminish d_k . We will not accept d_k until it reaches β_0 . If the ratio ρ_k turns out to be very good and exceeds a threshold β_1 , we can increase the step d_k by reducing the Lagrange multiplier p (here: $p := (1/10)p$), so that the penalty for large d_k is reduced.

The whole approach can be found in Algorithm 3.

Algorithm 3 Levenberg-Marquardt Method ([13],p.69)

Input: Starting point $x_0 \in \mathbb{R}^n$, $0 < \beta_0 < \beta_1 < 1$ and starting $p > 0$.

```

for  $k = 0, 1, 2, \dots$  do
    if  $F'(x_k)^T F(x_k) = 0$  then
        Stop
    end if
    repeat
        Compute Step  $d_k$  with  $d_k = -\left(F'(x_k)^T F'(x_k) + pI_{n \times n}\right)^{-1} F'(x_k)^T F(x_k)$ 
         $x_{k+1} = x_k + d_k$ 
        Calculate the ratio  $\rho_k$  with Equation 4.15.
        if  $\rho_k \leq \beta_0$  then
             $p := 10p$ 
        end if
        until  $\rho_k > \beta_0$ 
        if  $\rho_k > \beta_1$  then
             $p := \frac{1}{10}p$ 
        end if
    end for
    return  $x_0, x_1, x_2, \dots$ 

```

4.2. Approaches for Solving MINLP

4.2.1. NLP-Based Branch-and-Bound

We will first consider a MINLP in the following form

$$\begin{aligned}
 \min \quad & f(x, z) \\
 \text{s.t.} \quad & g_i(x, z) \leq 0, \quad i \in I \\
 & z \in Z \subset \mathbb{Z}^m \\
 & x \in \mathbb{R}^n
 \end{aligned} \tag{4.16}$$

where I is a finite set and f, g_i with $i \in I$ are continuously differentiable nonlinear functions and Z is a bounded polyhedral set. The form of this problem coincides with our application in Chapter 6, since our discrete values will be restricted to binary values, thus Z in our example is also bounded. In the following, we consider a *convex MINLP*, meaning that the objective function f and the constraints g_i are both convex. Since Z is a bounded set, we assume that finite lower and upper bounds are known for the integer values and we can rewrite Equation

4.16 into

$$\begin{aligned}
 \min \quad & f(x, z) \\
 \text{s.t.} \quad & g_i(x, z) \leq 0, \quad i \in I \\
 & z \in Z \subset \mathbb{Z}^m \\
 & l \leq z \leq u \\
 & x \in \mathbb{R}^n
 \end{aligned} \tag{4.17}$$

with $-\infty < l \leq u < \infty$.

Next, we relax the integer requirement on z and obtain a *NLP relaxation* of the MINLP in 4.17

$$\begin{aligned}
 \min \quad & f(x, z) \\
 \text{s.t.} \quad & g_i(x, z) \leq 0, \quad i \in I \\
 & z \in Z \subset \mathbb{R}^m \\
 & l \leq z \leq u \\
 & x \in \mathbb{R}^n
 \end{aligned} \tag{4.18}$$

This relaxation enlarges the feasible set as is demonstrated in Figure 4.1. On the left side of

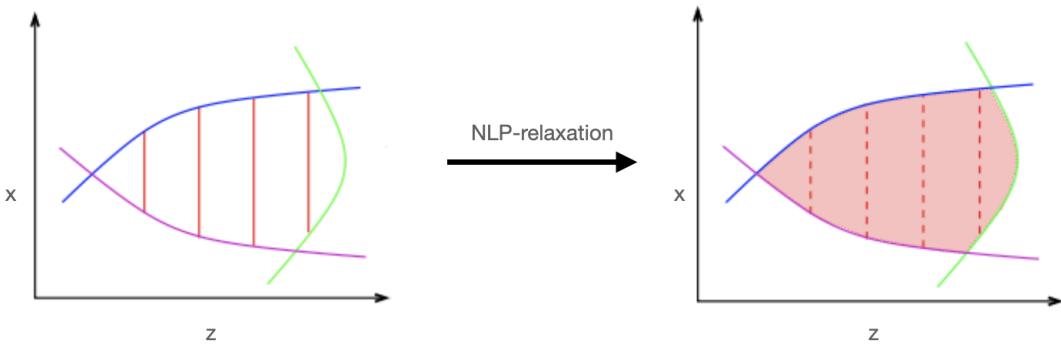


Figure 4.1.: Illustration of a NLP relaxation, ([15], p.7)

the figure, a graphical representation of a simple MINLP is shown, where z takes only integer values. In this figure, the red lines represent the feasible set. On the right side a NLP relaxation of the MINLP was carried out and z can now be a value in the real numbers. The feasible set for the NLP relaxation is represented in the red-shaded area. In comparison to the left side we obtain a larger feasible set. Since our feasible set became bigger, we can find a smaller objective function value for the NLP Relaxation. This result is summarized in the next Lemma.

Lemma 4.1 ([16], p.128). Let $l, u \in \mathbb{R}^m$ with $-\infty < l \leq u < +\infty$. Let σ_{NLP} be the objective function value to the optimal solution (x^*, z^*) of the relaxed problem in 4.18 and σ_{MINLP} the objective function value to the optimal solution of the MINLP problem in 4.17. If σ_{NLP} and σ_{MINLP} do not exist, they are set to $+\infty$. Then it holds

$$\sigma_{NLP} \leq \sigma_{MINLP}. \quad (4.19)$$

Remark: If additionally (x^*, z^*) is also a feasible solution to MINLP in 4.17, then (x^*, z^*) is already an optimal solution for this MINLP.

Lemma 4.1 provides upper bounds on the objective function value of the MINLP Problem and deduces optimality if z is already an integer value. For the Branch-and-Bound Algorithm we will also need another lemma.

Lemma 4.2 ([16], p.128-129). Let $l, u \in \mathbb{R}^m$ be finite. Furthermore let $z \in \mathbb{Z}^m$ be a feasible solution for 4.17 with bounds l and u . Let $i \in \{1, \dots, m\}$ with $l_i \leq s \leq u_i$ and $s \notin \mathbb{Z}$. Then z is a feasible solution for exactly one of the following problems:

1. MINLP 4.17 with bounds l and u , where

$$\underline{u}_j = \begin{cases} u_j, & j \neq i \\ \lfloor s \rfloor, & \text{else} \end{cases} \quad (4.20)$$

for $j = 1, \dots, m$ or

2. MINLP 4.17 with bounds \bar{l} and u , where

$$\bar{l}_j = \begin{cases} l_j, & j \neq i \\ \lceil s \rceil, & \text{else} \end{cases} \quad (4.21)$$

for $j = 1, \dots, m$.

For the algorithm we will use the notation $z^{(l,u)}, x^{(l,u)}$ to illustrate which boundaries were used in order to obtain the solutions to the NLP in 4.18.

The general NLP Branch-and-Bound approach can be found in Algorithm 4. Our first step in this Algorithm will be to initialize a set Q with boundaries l^{init} and u^{init} , which will be equal to the boundaries in the original MINLP problem. Furthermore, we set the *best objective value* σ equal $+\infty$, which we hope to update throughout the process. In the while loop we choose some bounds (l, u) from the set Q and solve a NLP with these particular bounds. If the problem is infeasible, we prune the node, which means, we do not consider any following subproblems of this problem, since they are also infeasible. If the objective function value is larger than σ , we discard this problem and all the following subnodes as we will not find any smaller objective function value for any following subproblems. If a solution $x^{(l,u)}, z^{(l,u)}$ to the NLP with bounds (l, u) was found and $z^{(l,u)}$ is in \mathbb{Z}^m , for which the objective function value is smaller than the last best objective value σ , we update our solution and set $\sigma = f(x^{(l,u)}, z^{(l,u)}), x^* = x^{(l,u)}, z^* = z^{(l,u)}$.

Algorithm 4 NLP-Based Branch and Bound ([16], [15])

```

Initial: Set  $\sigma \leftarrow +\infty$  and  $Q \leftarrow \{(l^{init}, u^{init})\}$ 
1: while  $Q \neq \emptyset$  do
2:   Choose  $(l, u)$  from  $Q$ .
3:    $Q \leftarrow Q \setminus \{(l, u)\}$  and solve NLP 4.18 with boundaries  $\{(l, u)\}$ .
4:   if NLP is infeasible then
5:     Node can be pruned, because it is infeasible.
6:   else if  $f(x^{(l,u)}, z^{(l,u)}) \geq \sigma$  then
7:     Node can be pruned, because we will not find a smaller objective function
8:     value for subnodes.
9:   else if  $z^{(l,u)} \in \mathbb{Z}^m$  then
10:    Update solution:  $\sigma = f(x^{(l,u)}, z^{(l,u)}), x^* = x^{(l,u)}, z^* = z^{(l,u)}$ 
11:   else
12:     Choose  $i$  with  $z_i^{(l,u)} \notin \mathbb{Z}$ 
13:      $Q \leftarrow Q \cup \{(l, \underline{u}), (\bar{l}, u)\}\}$ 
14:   end if
15: end while
16: if  $\sigma < \infty$  then
17:   return Optimal Solution  $(x^*, z^*, \sigma)$ 
18: else
19:   return "Problem is infeasible"
20: end if

```

If a solution with a better objective function value is found but $z^{(l,u)} \notin \mathbb{Z}^m$, we divide our problem into two subproblems according to Lemma 4.2 by choosing $i \in \{1, \dots, m\}$ and round up or off the value $z_i^{(l,u)} \notin \mathbb{Z}$. We add these two subproblems to the set Q . Finally, if there is no problem left in Q , we evaluate the *best objective value* σ . If $\sigma < \infty$, then we have found a feasible solution to our MINLP, otherwise the MINLP Problem is infeasible. There exist different continuations to Algorithm 4, which include specific node selection approaches and rules on how to choose the variable that should be rounded, also called *branching variable*.

To demonstrate the Branch and Bound approach, we consider an Integer Nonlinear Programming Problem (INLP).

$$\begin{aligned}
 \min \quad & (x_1 - \frac{1}{4})^2 + (x_2 - \frac{1}{4})^2 + x_3^2 \\
 \text{s.t. } & -2x_1 + 2x_2 - 1 \leq 0 \\
 & x_1, x_2 \in \mathbb{Z}
 \end{aligned} \tag{4.22}$$

First we compute an optimal solution of the NLP Relaxation of 4.22 and obtain $(x_1, x_2, x_3) = (1/4, 1/4, 0)$ with objective function value $f(x_1, x_2, x_3) = 0$. According to Lemma 4.1 this is the lower bound for the objective function value of the MINLP problem. x_1 and x_2 are both fractional values and we can select one of these variables to form our subproblem. We select x_1 as the first branching variable. The Algorithm creates two new subproblems: one, where we

append an additional constraint $x_1 \leq 0$ and one, where we add the constraint $x_1 \geq 1$. We start by solving the NLP with additional constraint $x_1 \leq 0$. Thus our NLP Problem becomes

$$\begin{aligned} \min \quad & (x_1 - \frac{1}{4})^2 + (x_2 - \frac{1}{4})^2 + x_3^2 \\ \text{s.t.} \quad & -2x_1 + 2x_2 - 1 \leq 0 \\ & x_1 \leq 0 \\ & x_1, x_2 \in \mathbb{R} \end{aligned} \tag{4.23}$$

We obtain an optimal objective function value $f(x_1, x_2, x_3) = 1/16$ with $x_1 = 0, x_2 = 1/4, x_3 = 0$ as solution values. The last variable that remains fractional is x_2 . The Algorithm again creates two new subproblems: one, where we add the constraint $x_2 \leq 0$ and one, where we add the constraint $x_2 \geq 1$ to the NLP in 4.23. The optimal solution to the first subproblem with additional constraints $x_1 \leq 0$ and $x_2 \leq 0$ is $(x_1, x_2, x_3) = (0, 0, 0)$ with $f(x_1, x_2, x_3) = 1/8$. Since all solutions are integer, we update $\sigma = 1/8$ as our best objective value. The second subproblem with additional constraints $x_1 \leq 0$ and $x_2 \geq 1$ is infeasible and is not taken into further consideration. Now, one tree branch remains untouched, which is the case $x_1 \geq 1$. The optimal solution to this problem is $(x_1, x_2, x_3) = (1, 1/4, 0)$ with $f(x_1, x_2, x_3) = 9/16$. This is larger than the so far best objective value $\sigma = 1/8$, thus we eliminate this problem from consideration. Now that we have worked through every case, our optimal solution obtained by the NLP-based Branch-and-Bound approach for Problem 4.22 is: $(x_1^*, x_2^*, x_3^*) = (0, 0, 0)$ with $f(x_1^*, x_2^*, x_3^*) = 1/8$.

We can represent these steps in a binary tree, the so called *Branch-and-Bound Tree* in Figure 4.2.

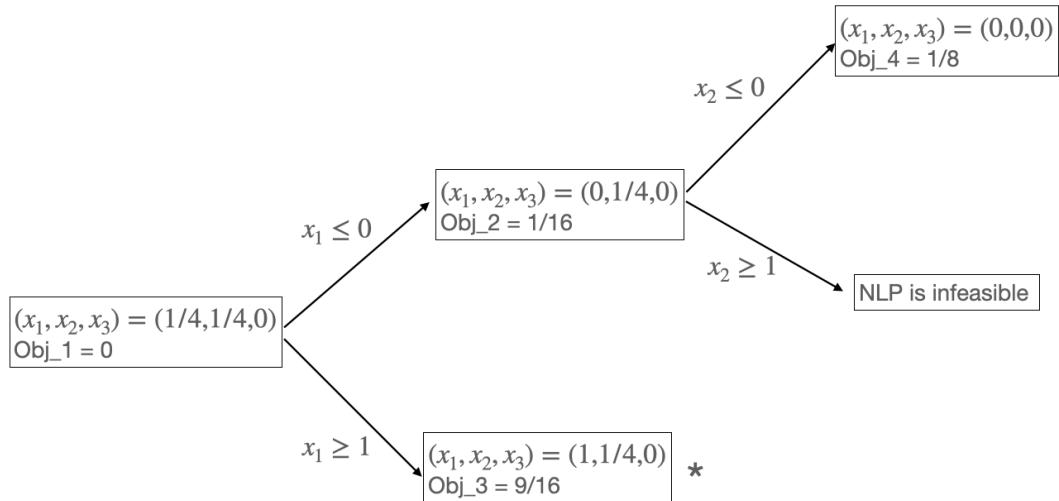


Figure 4.2.: Branch-and-Bound Tree for INLP 4.22

New challenges arise when we deal with non-convex MINLP. Considering convex MINLP problems, every local optima is also a global optima (see Appendix: Global and Local Minima). If the problem is non-convex, solving the NLP can initially lead to local optima and thus we will not have a lower valid bound for our MINLP, which means that we loose some solution quality. Another remarkable issue is that warm starts are not as efficient for NLP as they are for LP. Therefore, one common approach is to linearize the non-linear constraints and only use non-linear constraints, if its usage is justified or its linearization not possible or too coarse.

Another method is to find functions which cover or *envelope* the original function from above or beneath. We will provide a short outlook on this method in the following.

Definition 4.3 (Convex Underestimator and Convex Envelope ([16], p.137-138)). *Let $f : \Omega \rightarrow \mathbb{R}$ be a function over the convex set $\Omega \subset \mathbb{R}^n$. A function $\xi : \Omega \rightarrow \mathbb{R}$ is called a **convex underestimator** of f on Ω if ξ is a convex function and $\xi(x) \leq f(x)$ for all $x \in \Omega$. We denote the set of all convex underestimators as $\mathcal{U}(f, \Omega)$. Furthermore we define the function $vex_{\Omega}[f]$ as the supremum function of all functions in $\mathcal{U}(f, \Omega)$:*

$$vex_{\Omega}[f] := \sup\{\xi(x) : \xi \in \mathcal{U}(f, \Omega)\} \quad \text{for all } x \in \Omega. \quad (4.24)$$

$vex_{\Omega}[f]$ is denoted as **convex envelope** of f .

Similiar results can be formulated for concave overestimator/concave envelopes for maxima problems. We will provide a definition of a convex hull, which contains all convex

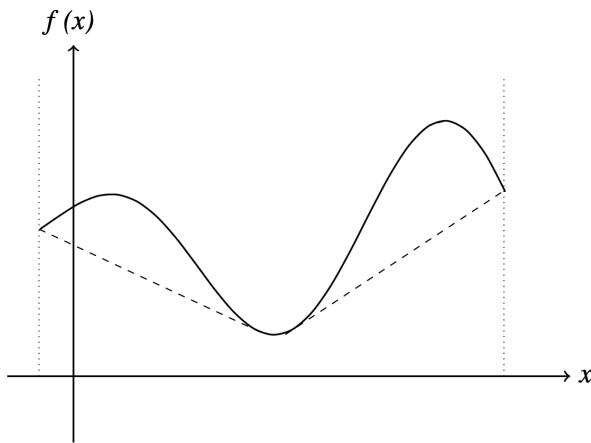


Figure 4.3.: Convex Envelope (dashed line) of a non-convex function (solid line)¹

combinations of points in Ω .

¹ [16], p.138

Definition 4.4 (Convex Hull ([16], p.139)). Let $\Omega \subseteq \mathbb{R}^n$ be a set. Then the convex hull $\text{conv } \Omega$ is defined as

$$\text{conv } \Omega = \left\{ \sum_{i=1}^n \lambda_i x_i : x_i \in \Omega, \lambda_i \geq 0 \text{ for all } i \in \{1, \dots, n\}, \sum_{i=1}^n \lambda_i = 1 \right\}. \quad (4.25)$$

With the two definitions provided above we can formulate the next lemma. This lemma allows us to use *convex envelopes* to compute global minima.

Lemma 4.5 ([16], p.139). Let $\Omega \subset \mathbb{R}^n$ be a compact set and $f : \Omega \rightarrow \mathbb{R}$ be a continuous function. Then it holds

$$\min_{x \in \Omega} f(x) = \min_{x \in \text{conv } \Omega} \text{vex}_\Omega[f](x). \quad (4.26)$$

Furthermore let \mathcal{M} be the set of global minima of f on Ω and let \mathcal{N} be the set of global minima of $\text{vex}_\Omega[f]$ over $\text{conv } \Omega$. Then it holds $\mathcal{N} = \text{conv } \mathcal{M}$.

Finding a convex underestimator, yet alone a convex envelope, turns out to be a difficult task in practice. A common approach is to use α -underestimators. Here we consider $\Omega \subseteq \mathbb{R}^n$ as a product of intervals, i.e.

$$\Omega = [\underline{x}, \bar{x}] = \prod_{i=1}^n [\underline{x}_i, \bar{x}_i] \quad (4.27)$$

and a function

$$\tilde{f}_\alpha = f(x) + \sum_{i=1}^n \alpha_i (\underline{x}_i - x_i)(\bar{x}_i - x_i), \quad \alpha \in \mathbb{R}_{\geq 0}^n. \quad (4.28)$$

Under certain conditions Function 4.28 is a convex underestimator of f . We can integrate these underestimator into the Branch-and-Bound Algorithm, where we no longer branch only on integer variables but also on continuous variables. This branching on continuous variables is known as *spatial branching* ([16], p.141).

4.3. Solving NLP: Interior Point Method

Consider the following Optimization Problem

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ & \text{s.t. } g_i(x) \leq 0, i \in I \\ & \quad c_j(x) = 0, j \in J \end{aligned} \quad (4.29)$$

with $I, J \neq \emptyset$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function and $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $c_j : \mathbb{R}^n \rightarrow \mathbb{R}$ are the constraint functions. Functions f , c_j and g_i can be either linear or non-linear. Furthermore we define

$$M = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, i \in I\} \neq \emptyset \quad (4.30)$$

as the set of all points $x \in \mathbb{R}^n$ which satisfy the inequality conditions and

$$M_< = \{x \in \mathbb{R}^n \mid g_i(x) < 0, i \in I\} \neq \emptyset \quad (4.31)$$

as the set of all $x \in \mathbb{R}^n$ which satisfy the **strict** inequality conditions g_i . We provide a definition of a barrier function, which can be interpreted as a function that penalizes values coming too close to the boundary of $M_<$.

Definition 4.6 (Barrier Function ([17], p.200)). *Function $\beta : M_< \rightarrow \mathbb{R}$ is called **barrier function** for M if for all sequences $(x_k) \subseteq M_<$ with limit $\lim_{k \rightarrow \infty} x_k = \bar{x} \in \partial M_<$ it holds*

$$\lim_{k \rightarrow \infty} \beta(x_k) = +\infty. \quad (4.32)$$

One important *barrier function* is the *logarithmic barrier function*

$$\beta(x) = - \sum_{i \in I} \log(-g_i(x)). \quad (4.33)$$

The Interior Point Method starts from a feasible point x_0 and approaches the optimum on the boundary of $M_<$ iteratively. The idea of the method is to replace the original problem with a problem where the inequality constraints g_i cannot be fulfilled with equality. We use the barrier function in the objective function, so in case if x_i fulfills g_i with equality, we have $\log(g_i) = \log(0) = -\infty$. We substitute Problem 4.29 with a new Optimization Problem containing the barrier function

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & B(\mu, x) := f(x) + \mu \beta(x) = f(x) - \mu \sum_{i \in I} \log(-g_i(x)) \\ \text{s.t. } & c(x) = 0 \end{aligned} \quad (4.34)$$

with $\mu > 0, \mu \in \mathbb{R}$.

As mentioned above objective function $B(\mu, x)$ is only defined for $M_<$. It conserves the smoothness property for $f(x)$ and $g_i(x)$ as long as $g_i(x) < 0$. For every $\mu > 0$, every optimal solution in Problem 4.34 is an inner point in M , therefore the method is called *Interior Point Method* or *Barrier Method*. If the numerical computation stops after some steps, we will achieve a feasible point with a usually improved objective function value.

But how can we solve Problem 4.34? The strategy is to solve a sequence of Problems 4.34 with starting μ_0, x_0 and decrease μ throughout the sequence. As μ is being decreased, the contribution of the penalty term is also being decreased and the sequence of x^k approaches the boundary.

Using KKT Conditions (see Appendix: Karush-Kuhn-Tucker Conditions) on the Barrier

Problem 4.34, we receive

$$\begin{aligned} \nabla f(x) + \sum_{j \in J} \beta_j \nabla c_j(x) + \sum_{i \in I} \alpha_i \nabla g_i(x) &= 0 \\ \alpha_i g_i(x) + \mu &= 0, \quad \forall i \in I \\ c_j(x) &= 0, \quad \forall j \in J \\ g_i(x) &\leq 0, \quad \forall i \in I \\ \alpha_i &\geq 0, \quad \forall i \in I \end{aligned} \tag{4.35}$$

where $\alpha_i, \beta_j \in \mathbb{R}$ are the Lagrange multipliers. Note that the complementary condition is disturbed by μ . For $\mu \rightarrow 0$ the complementary condition in 4.35 approaches the original complementary condition.

Applying Newton's Method to the equality constraints in 4.35, we obtain

$$\begin{bmatrix} \mathcal{L}_{xx} & \nabla g(x) & \nabla c(x) \\ \text{diag}(\alpha)^1 \nabla g(x)^T & \text{diag}(g(x)) & 0 \\ \nabla c(x)^T & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta x \\ \Delta \alpha \\ \Delta \beta \end{pmatrix} = - \begin{pmatrix} \nabla f(x) + \sum_{j \in J} \beta_j \nabla c_j(x) + \sum_{i \in I} \alpha_i \nabla g_i(x) \\ \text{diag}(\alpha) g(x) + \mu e \\ c(x) \end{pmatrix} \tag{4.36}$$

with $e = (1, \dots, 1)^T \in \mathbb{R}^{|I|}$, $g = (g_1, \dots, g_{|I|})^T \in \mathbb{R}^{|I|}$, $c = (c_1, \dots, c_{|J|})^T \in \mathbb{R}^{|J|}$, $\nabla g(x) = (\nabla g_1(x), \dots, \nabla g_{|I|}(x)) \in \mathbb{R}^{n \times |I|}$, $\nabla c(x) = (\nabla c_1(x), \dots, \nabla c_{|J|}(x)) \in \mathbb{R}^{n \times |J|}$. \mathcal{L} denotes the Lagrangian function

$$\mathcal{L} = f(x) + \alpha^T g(x) + \beta^T c(x). \tag{4.37}$$

with

$$\mathcal{L}_{xx} = H_f + \sum_{j \in J} \beta_j H_{c_j} + \sum_{i \in I} \alpha_i H_{g_i} \tag{4.38}$$

where $H_{(*)}$ denotes the Hesse matrix of the corresponding function.

After determining the new step length for step k denoted as $\Delta^k = (\Delta x^k, \Delta \alpha^k, \Delta \beta^k)$, a multiplication factor can be computed in two stages as is given below. First we determine the upper bound on step length γ_x^k and γ_α^k [18]:

$$\begin{aligned} (\gamma_x^{\max})^k &= \max\{\gamma \in (0, 1] : x^k + \gamma \Delta x^k \geq (1 - \tau)x^k\} \\ (\gamma_\alpha^{\max})^k &= \max\{\gamma \in (0, 1] : \alpha^k + \gamma \Delta \alpha^k \geq (1 - \tau)\alpha^k\} \end{aligned} \tag{4.39}$$

where $\tau \in (0, 1)$ (e. g. $\tau = 0.995$) is the fraction-to-the-boundary parameter. Afterwards a backtracking line search² with trial step size e.g. as given in [18]: $\gamma_p^{k,l} = 2^{-l}(\gamma_x^{\max})^k$, $l = 0, 1, \dots$ is performed, until "a step size $\gamma_p^k := \gamma_p^{k,l}$ is found that results in 'sufficient progress' towards solving

¹ Consider $x = (x_1, \dots, x_n)$. $\text{diag}(x)$ denotes a $\mathbb{R}^{n \times n}$ matrix with x on the diagonal and 0 elsewhere.

the barrier problem compared to the current iterate" ([18],p.9). We then obtain the final steplengths

$$\begin{aligned}\gamma_p^k &\in (0, (\gamma_x^{\max})^k] \subseteq (0, 1] \\ \gamma_d^k &= (\gamma_{\alpha}^{\max})^k.\end{aligned}\tag{4.40}$$

We can then compute new iterates $x^{k+1}, \alpha^{k+1}, \beta^{k+1}$ as follows

$$x^{k+1} = x^k + \gamma_p^k \Delta x^k, \quad \alpha^{k+1} = \alpha^k + \gamma_d^k \Delta \alpha^k, \quad \beta^{k+1} = \beta^k + \gamma_d^k \Delta \beta^k.\tag{4.41}$$

In the last step μ is being decreased allowing us to approach $\partial M_<$. We define the optimality error of the barrier problem as [19]

$$E(x^k, \beta^k, \alpha^k) := \max \left\{ \|\nabla f(x) + \sum_{j \in J} \beta_j \nabla c_j(x) + \sum_{i \in I} \alpha_i \nabla g_i(x^k)\|_{\infty}, \|c(x^k)\|_{\infty}, \|diag(\alpha)g(x^k) + \mu e\|_{\infty} \right\}\tag{4.42}$$

where the algorithm terminates if an approximate solution (x^*, β^*, α^*) is found satisfying $E(x^*, \beta^*, \alpha^*) < \epsilon_{tol}$. The whole approach for the Interior Point Method can be found in the following algorithm.

Algorithm 5 Interior Point Method

Input: starting point $x_0 \in M_<$, starting parameter μ_0 , barrier function β , tolerance ϵ

Output: approximation of a minimum of Problem 4.29

- 1: **repeat**
 - 2: Applying Newton-Raphson's method reformulate Problem 4.29 into Problem 4.36.
 - 3: Solve the new problem for $\Delta x_k, \Delta \alpha_k, \Delta \beta_k$.
 - 4: Determine step sizes $\gamma_p^k \in (0, (\gamma_x^{\max})^k]$ and $\gamma_d^k \in (0, (\gamma_{\alpha}^{\max})^k]$ using 4.39 - 4.40
 - 5: and backtracking line search.
 - 6: Compute new points $(x_{k+1}, \alpha_{k+1}, \beta_{k+1})$ according to 4.41.
 - 7: Decrease $\mu^{k+1} := c\mu^k$ with $c \in (0, 1)$.
 - 8:
 - 9: **until** $E(x^k, \beta^k, \alpha^k) < \epsilon_{tol}$
-

2 A backtracking line search starts with a large estimate of the step size and shrinks its step size. The shrinking continues until an expected decrease of the objective function is achieved along a given search direction Δx_k . Given a starting value x^k and a search direction Δx^k , we want to determine a step size γ^k which sufficiently reduces the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. This is fulfilled, if for some selected parameter $b \in (0, 1)$ it holds: $f(x^k + \gamma^k \Delta x^k) < f(x^k) + b\gamma^k \nabla f(x^k)^T \Delta x^k$.

5. Physical Characteristics of Gas-to-Power Networks

5.1. Power-to-Gas in Practice

This year mankind has counted more climate catastrophies than it has in a long time: heat records in North America, flood disasters in Germany and Belgium, burning forests in Italy and Turkey and many more. A sad résumé of the World Meteorological Organization shows that climate catastrophies have increased by a factor of five over the 50-year period starting in 1970 [20]. Among climate researchers, it is now undeniable that extreme weather events are related to increased carbon emissions since the industrial revolution. We must therefore find alternatives for coal, natural gas and oil. Renewable energies like hydropower, solar or wind energy have a great potential to replace carbon based energy sources. However, on days when there is little sunshine or wind, it may not be possible to generate enough energy for all households. One solution is to convert surplus renewable energy on windy or sunny days into hydrogen, save this hydrogen for days when there is little wind or sunshine and convert it back to power. The conversion from energy/power to hydrogen can be achieved by electrolysis. Electrolyzer technologies can be divided into three different methods: alkaline water electrolysis, proton exchange membrane (PEM) and high temperature electrolysis. Through the last years the energy conversion efficiency (see *energy conversion efficiency*, Glossary) of electrolyzers has increased up to 82% ([21], p. 34).

Regarding safety aspects (like avoiding oxyhydrogen reaction), the durability of the pipes and corrosion, multiple studies have concluded that only single digit percentages of hydrogen are acceptable to be mixed with natural gas and fed into the existing gas-pipe infrastructure. An alternative is to convert hydrogen into methane, which can be completely fed into the existing gas network and gas storage facilities. In a technical methanization process, methane can be produced through exothermal reaction of hydrogen and carbon dioxide. Carbon dioxide can be either taken from gases eliminated by the industry or extracted directly from the air. Another way to produce methane can be achieved through biological processes, such as using *archaea bacteria*. Nevertheless, the transformation of hydrogen to methane diminishes the energy conversion efficiency of power-to-gas in comparison to production of only hydrogen from 70.6% to 53.9% ([21], p. 39).

Some pilot projects in Germany have already proven the feasibility of the PtG-process,

with ZSW (Center of Solar Energy and Hydrogen Research) Stuttgart being the first one in 2009. Since then, around 20 more prototypes were constructed in Germany. A famous prototype was set up in Falkenhagen. In ca. three years the plant "Windgas-Falkenhagen" was run more than 10.000h accident-free and produced around 2.4 Mio. Nm^3 (see *standard volume*, Glossary) hydrogen ([21], p. 401).



Figure 5.1.: Power-to-Gas Station with methanization plant in Falkenhagen, Picture: Uniper

In order to convert gas back to electrical energy, gas power plants or cogeneration (combined heat and power - CHP) can be used, where thermal energy is being converted into kinetic energy. Gas turbines that run on traditional fuels can be upgraded in order to operate on hydrogen. This upgrade is offered by companies like General Electric or Siemens. When considering the power-to-power process solely for hydrogen (from renewable energy to gas-conversion and back to electrical energy) the energy conversion efficiency η is in range 30% to 38%, with CHP a higher energy conversion efficiency can be achieved with range 48% to 62% ([22], page 18), which does not deviate much from coal-fired power plants (range 30% – 45%).

5.2. Compressor Stations

In order to transport gas over long distances and overcome pressure loss caused by friction in pipes, pressure has to be increased throughout the way. Compressors have a very wide range of applications. Besides gas pipelines they can be found in refrigerators, air conditioners, in the blow moulding of PET bottles and in submarines to regulate buoyancy. The compressors main functionality is to increase pressure by reducing the gas' volume. In Figure 5.2 we can see different types of compressors and their operations range depending on the pressure and the volumetric flow of the gas. On the vertical axis the pressure in bar is displayed, on the horizontal axis the volumetric flow in (m^3/h) is shown. We can see that diaphragm- and piston compressors can achieve the highest pressure for hydrogen, whereas turbo compressors can work with the highest volumetric flow. A compressor is usually driven by electric motor or combustion engine.

Next we discuss three types of compressors: *piston compressor*, *diaphragm compressor* and *turbo compressor*.

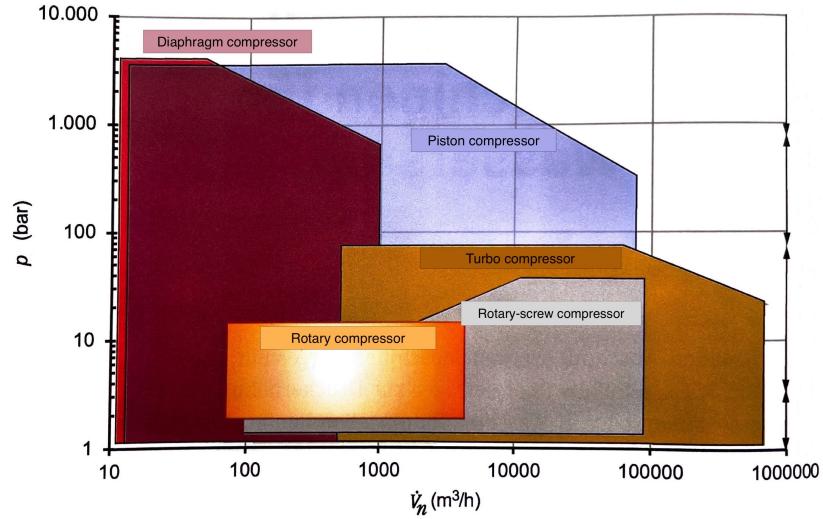


Figure 5.2.: Different types of compressors and its operating areas ¹

Piston compressor: In the following, we will describe the basic working process of a single unit piston compressor, which consists of a piston inside a cylinder and an engine connected to the piston through a crankshaft. If the piston is shifting down, the gas is being soaked into the cylinder through the open inflow valve. This inflow valve closes, when the piston reaches the bottom of the cylinder (also called dead center). Now the compression process begins. The piston travels upwards with both valves being closed. Through this upward movement the volume of the gas decreases and the pressure increases. When a certain pressure is reached, the second valve opens and lets the gas out. A visualization of the suction and the compression

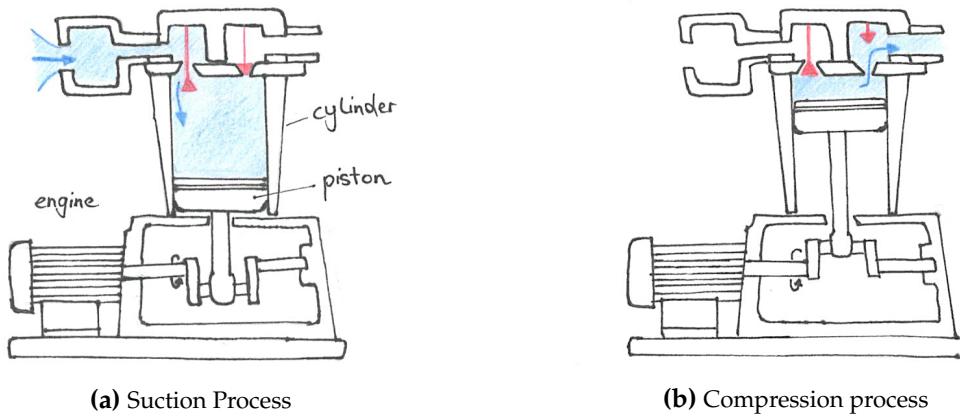


Figure 5.3.: Operating principle of a piston compressor

process is illustrated in Figure 5.3. A piston compressor can achieve high pressures through series connections of cylinders, where the gas is being cooled between the compressor stations.

¹ [23], p.338

Usually piston compressors are used in PtG pilot projects with small volumetric flow.

Diaphragm compressor: Diaphragm compressors operate similar to piston compressors, but in addition to a piston, a membrane is mainly responsible for the compression process. Diaphragm compressors (also known as membrane compressors) are suggested to use for achieving high outlet pressure for moderate volumetric flow (see Figure 5.2). The compression of gas is done via a flexible membrane and a lenticular area with two valves and a piston. When the inflow valve is open and the piston is travelling towards the dead end, the membrane is being deflected towards the bottom and the gas is being soaked in. As the piston travels up, the membrane bulges towards the other direction, thus reducing the volume of the gas. After a certain pressure is reached, the second valve, the outflow valve, opens ([23], p. 348).

Turbo compressor: Turbo compressors are used for high volumetric flows and are therefore the first choice when it comes to gas turbines, gas pipelines and refineries. An advantage of the turbo compressor is its high energy conversion efficiency. They are mainly divided into two types, axial compressor and centrifugal compressor, having both rotating blades in common. E.g. an axial compressor consists of alternating rows of rotating and stationary blades. The rotating blades accelerate the gas, the stationary blades convert the increased kinetic energy into pressure rise.

5.3. Power Grid Model

In Chapter 6 we will use the power flow equations to calculate the demand of gas of a power grid. We will first introduce the two terms *impedance* and *admittance*.

Impedance (measured in Ohm) is a measure of the opposition which extends the concept of resistance to AC (*alternating current*) circuits. *Admittance* (measured in Siemens) is defined as a measure of how easily a circuit will allow current to flow through it. Impedance consists of two parts, the real part which is resistance and the imaginary part reactance which comes from impeding (inductance and capacitance) mechanisms. Mathematically we write,

$$Z = R + iX \quad (5.1)$$

where R (measured in Ohm) is the resistance, X is the reactance (measured in Ohm) and Z is the impedance. Impedance and admittance are related to each other through the reciprocal. Choosing Y as a symbol for admittance we can write

$$Y = \frac{1}{Z} = \frac{1}{R + iX} = \frac{R}{R^2 + X^2} - i \frac{X}{R^2 + X^2}. \quad (5.2)$$

Thus we can also formulate the admittance as a complex number. We denote the real part as *conductance* G (measured in Siemens) and the imaginary part as *susceptance* B (measured in

Siemens) and rewrite Equation 5.2 in

$$Y = G + iB \quad (5.3)$$

with $G = \frac{R}{R^2+X^2}$ and $B = -\frac{X}{R^2+X^2}$. [24]

Another important term that we are going to use in our analysis is the *nodal admittance matrix*. The nodal admittance matrix is a quadratic $n \times n$ matrix which describes the admittance of n buses. More particular we denote Y_{jk} to be the admittance, G_{jk} to be the conductance and B_{jk} to be the susceptance between bus k and bus j with $Y_{jk} = G_{jk} + iB_{jk}$. The structure of the admittance matrix Y is typically symmetric. The diagonal elements of Y_{11}, Y_{22}, \dots are called the *self-admittances* at the corresponding buses.

Coefficients G_{jk} and B_{jk} will be important for the power flow study. The goal of the power flow study is to obtain the magnitude and phase angle of the voltage at each bus and the real and reactive power flowing through each transmission line.

We are now ready to introduce the power-flow equations

$$\begin{aligned} P_k &= \sum_{j=1}^N |V_k||V_j|(G_{kj}\cos(\phi_{k,j}) + B_{kj}\sin(\phi_{k,j})) \\ Q_k &= \sum_{j=1}^N |V_k||V_j|(G_{kj}\sin(\phi_{k,j}) - B_{kj}\cos(\phi_{k,j})) \end{aligned} \quad (5.4)$$

where P_k (measured in *Watt*) is the real power and Q_k is the reactive power (measured in *Voltage-Ampere-Reactive*) at node k , $|V_k|$ (measured in *Volt*) is the voltage amplitude, $\phi_{k,j}$ is the phase and B_{kj} , G_{kj} are parameters of the transmission lines between bus k and j . At each bus two of the four quantities ϕ_i , $|V_i|$, P_i , Q_i in Equation 5.4 are specified, the remaining two are unknown. The unknown and known variables are dependent on the bus type. In our optimization problem we will consider three types of buses: ([25],p.332)

Load bus: A *load bus* (or *PQ bus*) is a nongenerator bus. Here P and Q are usually known from historical record, load forecast or measurement. However, often in practice only real power is known and reactive power is based on an assumed power factor. The load bus is the most numerous bus type found in the power system.

Generator bus: A *generator bus* is a bus of the system at which the voltage magnitude is kept constant. Therefore the bus is also called *voltage-controlled bus*. At this bus we can specify V and P . Another name for the bus is thus *P-V bus*.

Slack bus: A slack bus is used to balance P and Q in a system by emitting and absorbing active and reactive power from and to the system. It provides angular reference for the angles of all other bus voltages. At the slack bus V and ϕ are given.

Bus	known variables	unknown variables
Load bus	P, Q	V, ϕ
Generator bus	P, V	Q, ϕ
Slack bus	V, ϕ	P, Q

Table 5.1.: Parameters in a power system

We can summarize all known and unknown parameters for each bus in Table 5.1. Our goal for the optimization problem is to obtain P at the slack node in order to find out how much gas is needed for the power-grid. Since for every bus there are two equations and two unknowns, we can solve a nonlinear system of Equations 5.4 using numerical methods. Most popular methods for obtaining a solution are Newton-Raphson or Gauss-Seidel method. However, we will make use of the Levenberg-Marquardt Method introduced in Chapter 4, since as mentioned before, it can cope better with bad guesses.

5.4. 1D Euler Equation

The gas dynamics in a single pipe can be modeled by the following set of nonlinear, hyperbolic PDEs, called *Isentropic Euler equation*, where the entropy is constant.

$$\frac{\partial}{\partial t} \rho + \frac{1}{A} \frac{\partial}{\partial x} \tilde{q} = 0 \quad (5.5)$$

$$\frac{1}{A} \frac{\partial}{\partial t} \tilde{q} + \frac{\partial}{\partial x} p + \frac{1}{A^2} \frac{\partial}{\partial x} \frac{\tilde{q}^2}{\rho} + g\rho s + \frac{\lambda(\tilde{q})}{2DA^2} \frac{\tilde{q}|\tilde{q}|}{\rho} = 0 \quad (5.6)$$

Here $\rho := \rho(x, t)$ is the density (in kg/m^3) of the gas, $\tilde{q} := \tilde{q}(x, t)$ is the (cross-sectional) mass flow (in kg/s), $p := p(x, t)$ is the pressure (in *bar*) with pressure-density relation $p = c^2 \rho^\gamma$ and constants $\gamma \geq 1$ and $c \in \mathbb{R}$. D is the diameter (in m) of the pipe, A is the cross-sectional area (in m^2) of the pipe and is calculated by $A = D^2 \pi / 4$, g is the gravitational acceleration (with standard value $9.80665 \frac{m}{s^2}$), $\lambda(q)$ is the friction factor of the gas and $s \in [-1, 1]$ denotes the slope of the pipe, i.e. the tangent of its inclination angle. In addition, we establish the following relationship: $\tilde{q} = A\rho v$, where $v = v(x, t) \in \mathbb{R}$ is the velocity in the direction of the pipe [26].

Equation 5.5, the so called *Continuity Equation*, describes the conservation of mass. It states that the rate of change of the gas density in time is equal to the rate of change of mass flow across its boundaries. Meaning, if the gas flow entering the pipe is growing constantly while the gas flow leaving the pipe stays the same, the gas density inside the pipe increases.

Equation 5.6, known as the *Momentum Equation*, describes the conservation of momentum. It includes the sum of all forces on gas particles. The third term on the left side of Equation 5.6 represents inertia (see Glossary). The fourth term on the left side represents the gravity force which depends on the slope of the pipe. The fifth term on the left side stands for the friction force.

We will further simplify Equation 5.5 and 5.6 by transforming cross-sectional mass flow \tilde{q} into a point-wise mass flow q (\tilde{q} could be interpreted as the mass-flow on a surface A) using the relation $\tilde{q} = Aq$ and $q = \rho v$:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}q = 0 \quad (5.7)$$

$$\frac{\partial}{\partial t}q + \frac{\partial}{\partial x}p + \frac{\partial}{\partial x}\frac{q^2}{\rho} = -g\rho s - \frac{\lambda(q)|q|q}{2D\rho} \quad (5.8)$$

There is another widely used representation of the flux, namely the *volumetric flux* Q . We can calculate Q using the following relation

$$\tilde{q} = \rho_0 Q \quad (5.9)$$

where ρ_0 (kg/m^3) is a norm density constant. Throughout this Thesis however, we will work with the mass flow q .

From a technical perspective we can state, that gas dynamics are influenced by the length L and the diameter D of a pipe in the following way: The longer the pipe, the larger is the pressure difference between its endpoints for a fixed amount of flow coming in and the larger the diameter D , the lower is the occurring pressure drop ([26],p.23).

Pipes are usually constructed out of carbon steel with varying surface characteristics. The pressure drop in a pipe is caused primarily by friction due to the roughness or unevenness of the inner pipe walls. Another cause of frictional resistance may come from factors like the presence of weld seams, the curvature of the pipe, corrosion and deposition of dirt [26]. Frictional forces are expressed by the friction factor $\lambda(q)$, which differ for laminar and turbulent flow.

A laminar flow is a movement of liquids and gases in which no visible turbulances occur and where all the particles of the fluid move in organized layers and the layers do not mix with each other. A turbulent flow on the other hand is characterized by irregular fluctuations or mixing of the layers. The best way to illustrate this is considering smoke rising from a candle flame - at first the smoke behaves laminar and as it rises, and the flow velocity increases, it becomes turbulent. The transformation from laminar to turbulent flow can, to some extent, be predicted by the *Reynolds number*. Laminar flow occurs at low Reynolds numbers, while turbulent flow occurs at high Reynolds numbers. Mathematically, the Reynolds number is defined as follows:

$$Re(q) = \frac{D}{\nu}|q|. \quad (5.10)$$

Here ν denotes the *dynamic viscosity* of the gas. In literature the flow is usually characterized as turbulent for $Re(q) \geq Re_{crit} \approx 2320$ [26].

The friction factor λ can then be formulated with the help of Reynolds number. There are different ways of calculating λ , depending on the flow type (laminar or turbulent) and thus the range of the Reynolds number.

In Table 5.2 some popular approaches for calculating the friction factor λ are presented.

Type of flow	$Re \cdot k/D$	Range	Equation of	Formula
laminar		$Re \leq 2320$	Hagen-Poiseuille	$\lambda(q) = \frac{64}{Re(q)}$
turbulent	≤ 65	$2320 < Re < 10^5$	Blasius	$\lambda(q) = 0.3164Re^{-0.25}$
		$10^5 < Re < 5 \cdot 10^5$	Nikuradse	$\lambda(q) = 0.0032 + 0.221Re^{-0.237}$
		$Re > 10^6$	Prandtl and Karman	$\frac{1}{\lambda} = 2\log(Re\sqrt{\lambda}) - 0.8$
turbulent	> 1300		Nikuradse	$\frac{1}{\sqrt{\lambda}} = 2\log\left(\frac{D}{k}\right) + 1.14$
turbulent	> 65 or < 1300		Prandtl and Colebrook	$\frac{1}{\sqrt{\lambda}} = -2\log\left(\frac{2.51}{Re\sqrt{\lambda}} + \frac{0.269k}{D}\right)$

Table 5.2.: Determination of the friction factor ([23], p.318)

For turbulent cases, the Prandtl and Colebrook formula is the first choice to work with since it is considered to be an accurate approximation of reality. For the implicit Prandtl and Colebrook formula a number of explicit approximations exist. One approximation is the Hofer-Equation

$$\lambda(q) = \left(-2\log\left(\frac{4.518}{Re(q)}\log\left(\frac{Re(q)}{7}\right) + \frac{k}{3,71D}\right) \right)^{-2}, \quad (5.11)$$

another approximation is the formula of Nikuradse

$$\lambda(q) = \left(2\log\left(\frac{D}{k}\right) + 1.138 \right)^{-2}. \quad (5.12)$$

Pipe type and condition	$k(\text{mm})$
Steel pipe not corroded, without seams or longitudinally welded	0.1
Slightly corroded steel pipe	0.5
Heavily corroded steel pipe	2
Pipe made of synthetic/plastic material (PVC, PE,...)	0.1

Table 5.3.: Roughness k for different types and conditions of pipes ([23], p.320)

In the equations and the table above a value k appears. This constant stands for the roughness of the inner wall. Some suggestions for k for different pipe types can be found in Table 5.3.

We return to the Isentropic Euler equations 5.7 and 5.8. If we choose $\gamma = 1$ for the pressure-density-relation $p(\rho) = c^2\rho^\gamma$, we obtain the *Isothermal Euler equation*. For our optimization

problem in Chapter 6 we choose c equal to the speed of sound which is approximately $c = 340 \frac{m}{s}$. By using the *Isothermal Euler equation* we assume that the temperature is constant throughout the gas network (*isothermal process*, see Glossary), which seems reasonable for pipes buried underground.

In the following we are going to introduce a simplification of the Euler equations 5.7 and 5.8, the so called *Weymouth equation*.

5.4.1. Weymouth Equation

In Paper [27] the authors suggest a more simplified version of the Euler equation, such that for a high-pressure gas pipes ($c^2\rho \approx 70 \text{ bar}$ with $c \approx 300 \text{ m/s}$ and $q/\rho \approx 10 \text{ m/s}$) the inertia term in *Momentum Equation* can be neglected. We can clarify this with the following example.

Assume that we choose $p = 70 \text{ bar}$ as is mentioned above. Then by using the pressure-density-relation we calculate the density

$$\rho = 70 \text{ bar} / (300 \frac{m}{s})^2 = 7000000 \text{ Pa} / (300 \frac{m}{s})^2 \approx 77.78 \frac{\text{kg}}{\text{m}^3}. \quad (5.13)$$

With $q/\rho \approx 10 \text{ m/s}$ we calculate the mass flux $q = \rho \cdot 10 \frac{m}{s} = 777.8 \frac{\text{kg}}{\text{m}^2 \text{s}}$. The inertia term in the momentum equation becomes then:

$$q^2/\rho = 777.8^2/77.8 \approx 7776 \quad (5.14)$$

whereas

$$c^2\rho = (300 \frac{m}{s})^2 \cdot 77.78 \approx 7000200 \quad (5.15)$$

which is 900 times higher than the inertia term 5.14. Thus for high-pressure gas pipes we deduce the following approximation:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}q = 0 \quad (5.16)$$

$$\frac{\partial}{\partial t}q + \frac{\partial}{\partial x}c^2\rho = -g\rho s - \frac{\lambda(q)|q|q}{2D\rho}. \quad (5.17)$$

This approximation is known as *Weymouth equation* [27].

A further simplification can be achieved by assuming that the pipes are not inclined, such that no gravity forces are present. By setting $s = 0$, where s denotes the slope of the pipe, the gravity term becomes zero. In our optimization model in Chapter 6 we will also neglect the gravity force.

From a computational point of view this approximation still appears to be expensive, which leads to even further simplifications like the *quasi-static model* or the ODE approximation by *SIMONE simulation program*. Nevertheless Weymouth equations, in comparison to Isothermal Euler equations, are more practical when it comes to taking control of the CFL condition and we

therefore can choose a uniformal space and time step.

5.5. Node Coupling Conditions

5.5.1. Coupling Conditions at Gas Nodes

In our compressor optimization model we will consider four type of nodes: *sources*, *sinks*, *compressor nodes* and *innodes*. We will state the set of all nodes as V .

We denote compressor nodes V_c as nodes at which pressure can be increased. At *source node* V_q the network is supplied with gas. The *sink node* V_p in our network is a node, which has no outgoing pipes. It is the "end node" of our network. The *innodes* are nodes inside the network connecting two or more pipes with each other.

Furthermore we define $\delta^+(v)$ as the set of outgoing edges at node $v \in V$ and $\delta^-(v)$ as the set of ingoing edges at node v .

We let E be the set of all pipes. Then, we denote ${}^e q(x, t)$ and ${}^e p(x, t)$ with $e \in E$ as the flow or pressure respectively at pipe e in point (x, t) . In the following we will introduce two Kirchhoff-type coupling conditions, which we will use for our compressor optimization problem. We will state them in a discretized manner.

The first condition ensures, that the sum of ingoing flows is equal to the sum of outgoing flows

$$\sum_{e \in \delta^-(v)} {}^e q(x^n, t) = \sum_{h \in \delta^+(v)} {}^h q(x^0, t), \quad \forall v \in V \setminus (V_q \cup V_p), \quad \forall t = 0, \dots, T. \quad (5.18)$$

The second condition claims, that there is no pressure loss near the node in all pipes connected to it. For a fix but arbitrary node in the innodes $v \in V \setminus (V_q \cup V_c \cup V_p)$ and a pressure $p(v, t)$ at node v in time t we state

$$\begin{aligned} {}^e p(x^n, t) &= p(v, t), & \forall e \in \delta^-(v), \quad \forall t = 0, \dots, T \\ {}^h p(x^0, t) &= p(v, t), & \forall h \in \delta^+(v), \quad \forall t = 0, \dots, T. \end{aligned} \quad (5.19)$$

5.5.2. Coupling Conditions at Slack Node

At the slack bus gas is being converted into real power. Therefore gas flow ϵ_t , $\forall t = 0, \dots, T$, is being taken out of the gas network at one node s and transported to the slack bus via edge h_{slack} . The coupling condition at node s can thus be stated as follows

$$\sum_{e \in \delta^-(s)} {}^e q(x^n, t) = \sum_{h \in \delta^+(s) \setminus h_{slack}} {}^h q(x^0, t) + \epsilon_t, \quad \forall t = 0, \dots, T. \quad (5.20)$$

At the same time we want the pressure to stay the same although flux is taken out. Thus conditions 5.19 also hold at the slack connection node s .

5.5.3. Conditions at Compressor Node

We consider an external power supply for the compressor and demand that at compressor nodes the sum of outgoing flows is equal to the sum of ingoing flows. Therefore condition 5.18 also has to hold. However, at compressor nodes, we allow the pressure to increase. In our optimization model we assume for the sake of simplicity that the compressor node is only connected to one ingoing - and one outgoing pipe. Then we formulate the pressure requirement as follows

$${}^e p(x^n, t) = {}^h p(x^0, t) + u_v^t, \quad \text{for } v \in V_c, e \in \delta^-(v), h \in \delta^+(v), \forall t = 0, \dots, T \quad (5.21)$$

where $u_v^t \geq 0$ denotes pressure increase (in bar) at compressor node v .

6. Compressor Cost Optimization Model

6.1. Comparing Weymouth and Euler Equations Using Two Discretization Methods

In the following we are going to compare both the Euler- and Weymouth equation with the Simple Upwind method as well as the Lax-Friedrich method in a single pipe. In order to verify our results with already existing pipe simulations, we will use the graphical results for flux and density presented in Lemma 1 of the Paper '*The isothermal Euler equations for ideal gas with source term*' by Martin Gugat and Stefan Ulbrich [28]. The authors of this paper considered a length of 2.5 m and a time horizon of 2 seconds. Furthermore the following initial conditions were considered

$$q(0, x) = 0 \quad \text{and} \quad \rho(0, x) = \exp(\beta x) \quad (6.1)$$

$$q(t, 0) = 10t^2 \quad \text{and} \quad \rho(t, 0) = 4.75t^2 + 1. \quad (6.2)$$

Here, the initial conditions in 6.1 were provided directly in the paper, whereas the initial conditions in 6.2 had to be determined graphically.

The constants are chosen as follows: $c^2 = 1$, $\frac{\lambda}{D} = 0.7$ and $\beta = -1$. The gravity force is not considered ($s = 0$). For the constants and initial conditions provided as above, Gugat and Ulbrich show the graphical solutions presented in Figure 6.1.

For our simulation we choose $\Delta t = 0.0001 \text{ sec}$ as our time step size and $\Delta x = 0.0625 \text{ m}$ as our spatial step size, such that the CFL condition is satisfied. In total we have 20000 time steps and 41 spatial steps. In Figure 6.2 we simulate both the Euler- and Weymouth equations with the Simple Upwind method and the Lax-Friedrich method with the initial conditions, constants and discretization steps described as above. We see that in this case the Simple Upwind Scheme performs bad for both Euler- and Weymouth equation. This scheme leads to unstable densities ρ and fluxes q as time progresses. In comparison to this, the Lax-Friedrich Method performs much more stable. This is quite reasonable as the scheme does not only make use from the preceding but also from the forerunning value which delivers more support for stability. Varying Δt in numerical experiments has shown that the graphical results obtained with Lax-Friedrich Method approach the results in Figure 6.1 the smaller Δt is chosen. It can also be observed that the Euler equation deviates only slightly from the Weymouth equation, when it is discretized

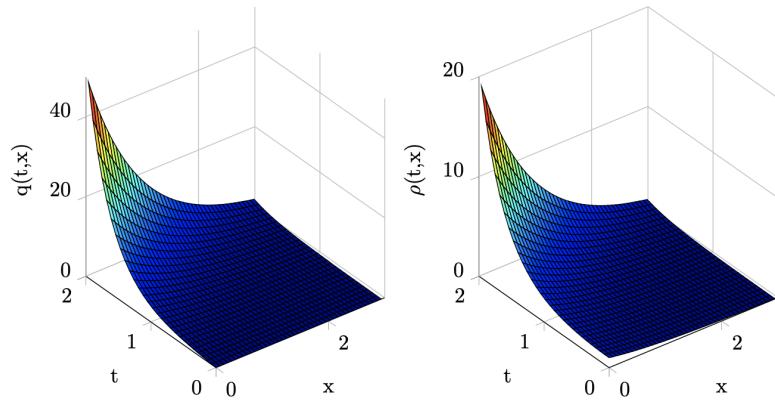


Figure 6.1: Graphs of $q(t, x)$ and $\rho(t, x)$ for the solution in Lemma 1 for $\frac{\lambda}{D} = 0.7$, $\beta = -1$, $a = 1$ ([28], p.7).

with Lax-Friedrich method.

If we take a closer look at the values ρ obtained by the Weymouth equation with Simple Upwind Scheme, we notice that ρ starts to get out of control after roughly 2030 time steps. Thus, when it comes to simulating the Euler and Weymouth equation for many time steps in a pipe, taking Lax-Friedrich Method for discretization seems to be the better choice.

Let us further consider a smaller model with $c = 340$ (m/s), 600 time steps and 6 spacial steps, where $T = 5$ sec and the length of pipe is 10000 m. Therefore $\Delta t = 1/120$ and $\Delta x = 1666.\bar{6}$. We choose $\lambda = 0.11$ and $D = 1$ m. Furthermore we set the following initial conditions in a discrete manner

$$q(0, x_j) = 550 - 10j \quad \text{and} \quad \rho(0, x_j) = 60 - 3j, \quad \forall j = 0, \dots, n \quad (6.3)$$

$$q(t_i, 0) = 550 \quad \text{and} \quad \rho(t_i, 0) = 60, \quad \forall i = 0, \dots, m. \quad (6.4)$$

We let ρ and q in (t_i, x_n) be equal for both Lax-Friedrich and Simple Upwind Scheme, meaning that at the end of the pipe we have the same values for both discretizations $\forall i = 0, \dots, m$ for ρ and q respectively. The goal of this investigation is to observe how Lax-Friedrich and Simple Upwind behave in a smaller model, if values for $q(0, x_j)$, $\rho(0, x_j)$ and $q(t_i, 0)$, $\rho(t_i, 0)$, $q(t_i, x_n)$, $\rho(t_i, x_n)$ are the same for both discretization methods. The results for the given initial conditions above are shown in Figure 6.3. Here we see that there is almost no difference in density for both the Euler and the Weymouth equations for each discretization method. This is due to the small influence of mass flow q on ρ (detailed discussion in Chapter 7.2).

For mass flow q things are different. It behaves more smooth with the Lax-Friedrich discretization, regarding the Simple Upwind Method the graph falls down in the middle. Thus for the mass flow the discretizations do not deliver the same results. Again we notice that there is hardly any difference between the Euler- and Weymouth equations, this time for any discretization method.

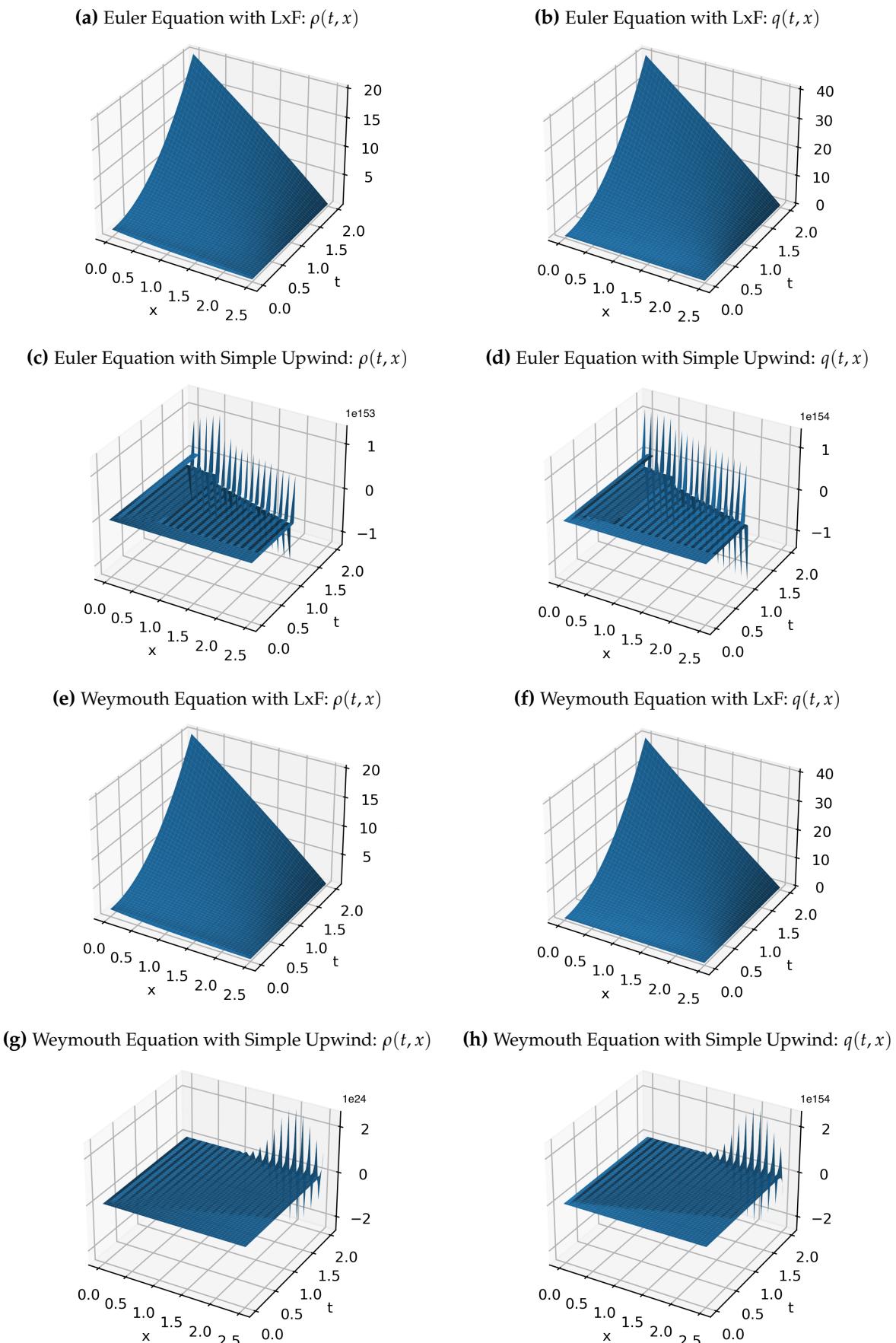
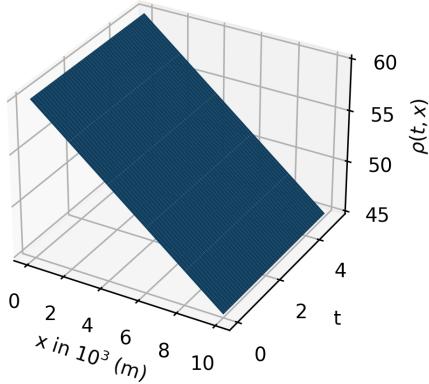
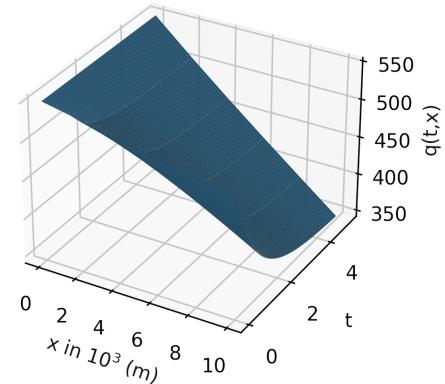
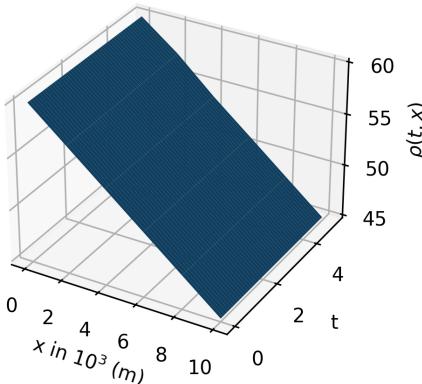
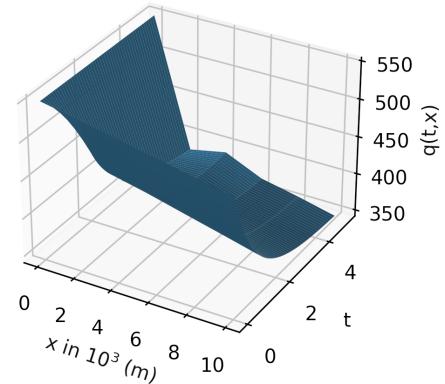
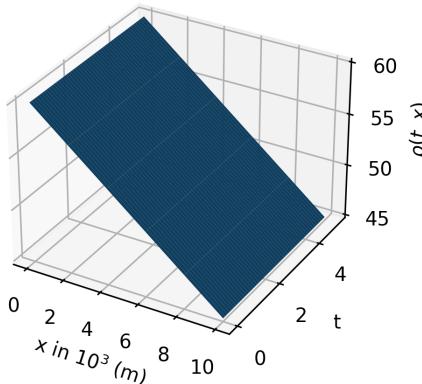
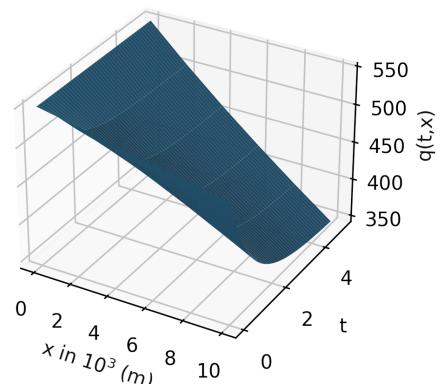
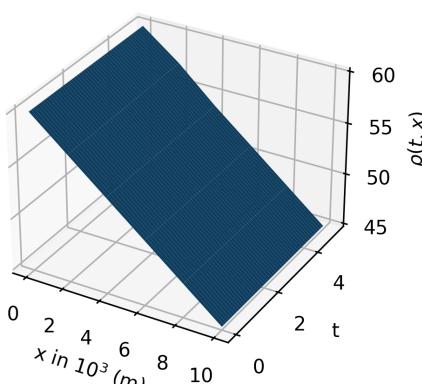
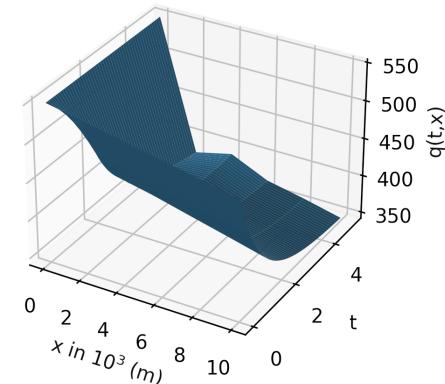


Figure 6.2.: Comparing different discretization methods for Weymouth and Euler equations

(a) Euler Equation with LxF: $\rho(t, x)$ (b) Euler Equation with LxF: $q(t, x)$ (c) Euler Equation with Simple Upwind: $\rho(t, x)$ (d) Euler Equation with Simple Upwind: $q(t, x)$ (e) Weymouth Equation with LxF: $\rho(t, x)$ (f) Weymouth Equation with LxF: $q(t, x)$ (g) Weymouth Equation with Simple Upwind: $\rho(t, x)$ (h) Weymouth Equation with Simple Upwind: $q(t, x)$ **Figure 6.3.: Comparing different discretization methods for Weymouth and Euler equations**

6.2. Problems Concerning Modeling Gas Networks with Lax-Friedrich Method

Although the simulation with Lax-Friedrich showed very good results for one pipe, a concern arises when it comes to modelling pipes in a network. For the linear advection equation in one space dimension $u_t + au_x = 0$ we remind that the Lax-Friedrich method is given as follows

$$u_j^{i+1} = \frac{1}{2}(u_{j-1}^i + u_{j+1}^i) - a \frac{\Delta x}{2\Delta t} (u_{j+1}^i - u_{j-1}^i) \quad (6.5)$$

with discrete grid points (x_j, t_i) , $j = 0, 1, 2, \dots$ and $i = 0, \dots, T$. This means that in order to calculate a value in point (x_j, t_{i+1}) one needs information at grid points (x_{j+1}, t_i) and (x_{j-1}, t_i) . In terms of modelling a pipe network with compressors one might stumble upon a problem when discretizing the model with Lax-Friedrich method. This can be seen in the following two pictures. The red line represents the location of the compressor. The orange grid cells are given

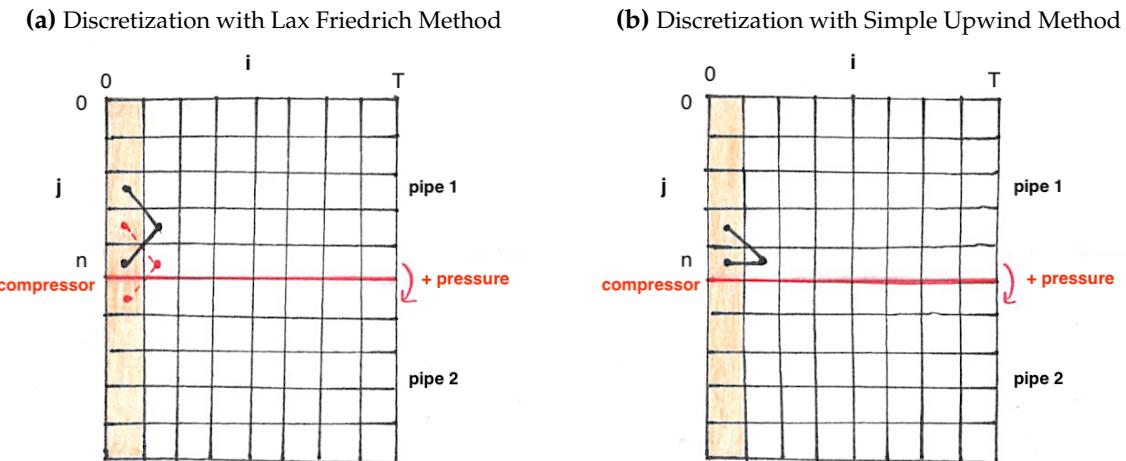


Figure 6.4.: Transition of two pipes with a compressor inbetween

as initial conditions in $t = 0$. On the left side we see a picture of Lax-Friedrich discretization. It uses the preceding and subsequent space points in time t in order to calculate a value at time $t + 1$. The calculation is shown by black lines where we calculate a value at grid point (x_n, t_1) . Theoretically we have to proceed with this approach and use it in point (x_n, t_1) . Otherwise our value in (x_n, t_1) becomes decoupled and the network will generate pressure and flux out of nowhere. This was an issue I stumbled upon in simulation. The problem that arises is that our compressor increases pressure between (x_n, t_1) in the first pipe and (x_0, t_1) in the second pipe. If the compressor is working, we will have a greater pressure in (x_0, t_1) than in point (x_n, t_1) . But this increasing pressure should have no influence on the pressure in the grid points of the first pipe (see operating principle of a piston compressor in Figure 5.3, Chapter 5.2. : Compressor Stations). Thus it distorts the models intention.

A discretization method which was also chosen for testing the three-step approach on

transmission lines in [1] is the Simple-Upwind Scheme, which is

$$u_j^{i+1} = u_j^i - a \frac{\Delta x}{\Delta t} (u_j^i - u_{j-1}^i) \quad (6.6)$$

for the advection equation with $a > 0$. This is shown on the right hand side of Figure 6.4. For calculating values in (x_n, t_1) only grid points in (x_n, t_0) and (x_{n-1}, t_0) are needed. So if the compressor increases pressure it has no affect on the value at point (x_n, t_1) .

But as we saw in the previous section, the Simple Upwind Scheme does not lead to promising results when many time steps, such as 20000, are considered. In our simulation we operate with around 300 total time steps and a total of 6 space steps, a simulation which lasts only around 2 sec. A smaller number of time and spatial steps however show decent results for the Simple Upwind scheme as is shown in the second example, Figure 6.3. However for real time approaches a simulation which lasts only 2 seconds is not enough and the Simple Upwind Scheme will fail.

Another discussion that we want to initiate, which also goes hand in hand with the debate above, at this point is: How do we choose our initial data? If we choose to work with the Lax-Friedrich method, we have to provide ghost-cells at each node which takes some degrees of freedom from our optimization model and might lead to a 'too tight' model (the optimization problem has at most one solution). For our further numerical analysis we choose to work with more degrees of freedom and go with the Simple Upwind Scheme since we consider only a small number of time and spatial steps and since we can just provide initial conditions in time $t = 0$ without having concerns on providing additionally values for pressure and flux at the gas net nodes in every time t .

6.3. Compressor Cost Optimization Problems

For our numerical analysis we are going to consider two different models. A MINLP with POC reformulation and a MINLP without the POC reformulation. Both belong to the class of nonlinear, nonconvex Mixed-Integer Problems. In both problems we consider $j = 0, \dots, n$ as our space index and $t = 0, \dots, m$ as our time index. We denote ρ as the density (kg/m^3), p as pressure (bar), q as mass flow (kg/s). Furthermore the relation $p(\rho) = c^2\rho$ holds with flux speed $c = 340(m/s)$. Our control variable u_{com} denotes the pressure increase (bar) at compressor com . ϵ^t is the flux which is taken by the slack connection node at time t and is calculated beforehand by solving the power-flow equations with the Levenberg-Marquardt algorithm. We model our network such that the compressor is located between two pipes (but not every pipe). We choose our objective function to be the sum of all additional pressure to the power of two at every time step and every compressor. As mentioned above we discretize the Weymouth equations with the Simple Upwind Scheme. Furthermore we choose a constant friction factor $\lambda = 0.11$. This factor is constant for every pipe section in every pipe at every time step t . We consider n_{com} as

the number of all compressors in the gas network. We refer to ${}^e p_n^t$ as pressure in edge e at spacial point n in time t . We denote V_c as the compressor nodes, V_p as sink node, V_q as source node and V as the set of all nodes. Node s is the gas node which is connected to the slack bus (we assume that there is only one such gas-net node). With h_{slack} we denote the edge which connects s and the slack bus.

6.3.1. Optimization Problem without POC

We introduce $\beta^t \in \{0, 1\}^{n_{com}}$ as a binary variable in time t . If it holds $\beta_v^t = 1$ for $v \in V_c$, then we say that compressor in node v is on in time t . If $\beta_v^t = 0$ for $v \in V_c$, then the compressor in node v is off in time t . We formulate our first MINLP Optimization Problem as follows

$$\min_u \quad \frac{1}{2} \sum_{t=0}^m \sum_{v \in V_c} (u_v^t)^2$$

subject to:

Weymouth Equation

$$\begin{aligned} 1. \quad & \rho_j^{t+1} = \rho_j^t - \frac{\Delta t}{\Delta x} (q_j^t - q_{j-1}^t) \\ & q_j^{t+1} = q_j^t - a^2 \frac{\Delta t}{\Delta x} (\rho_j^t - \rho_{j-1}^t) - \Delta t \frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t} \\ & \forall j = 1, \dots, n, \forall t = 0, \dots, m-1, \text{ for all pipes} \end{aligned}$$

2. gas coupling conditions

for a fix but arbitrary innode $v \in V \setminus (V_q \cup V_p \cup V_c)$ it holds:

$$\begin{aligned} {}^e p_n^t &= p(v, t), \quad \forall e \in \delta^-(v), \forall t = 0, \dots, m \\ {}^h p_0^t &= p(v, t), \quad \forall h \in \delta^+(v), \forall t = 0, \dots, m \end{aligned} \tag{6.7}$$

for every node $v \in V \setminus (V_q \cup V_p)$ it holds:

$$\sum_{e \in \delta^-(v)} {}^e q_n^t = \sum_{h \in \delta^+(v)} {}^h q_0^t, \quad \forall t = 0, \dots, m$$

3. compressor conditions

$$\begin{aligned} {}^e p_n^t &= {}^h p_0^t + u_v^t, \quad \forall v \in V_c, e \in \delta^-(v), h \in \delta^+(v), \forall t = 0, \dots, m \\ u_v^t &= \beta_v^t \cdot u_v^t, \quad \forall v \in V_c, \forall t = 0, \dots, m \end{aligned}$$

4. non - negativity constraints

$$u_v^t \geq 0, \quad \forall v \in V_c, \forall t = 0, \dots, m$$

$$p_j^t \geq 0, \quad \forall j = 0, \dots, n, \forall t = 0, \dots, m, \text{ for all pipes}$$

5. gas - power coupling at node s

$${}^e q_n^t = {}^h q_0^t + \epsilon_t, \quad e \in \delta^-(s), h \in \delta^+(s) \setminus h_{slack}, \forall t = 0, \dots, m$$

Initial conditions: $\rho_j^0, q_j^0, \forall j = 0, \dots, n$, for each pipe and $\epsilon^t, \forall t = 0, \dots, m$.

Solution: $\rho_j^t, q_j^t, \forall j = 0, \dots, n, \forall t = 0, \dots, m$, for each pipe, u_v^t and configuration $\beta_v^t, \forall v \in V_c$ and $\forall t = 0, \dots, m$.

Here condition 1 is the Weymouth equation discretized with Simple Upwind Method, Condition 2 are gas coupling conditions, where the first equation guarantees that the pressure in the pipe near the node is the same as the pressure at the node, the second equation makes sure that the outgoing and ingoing flows add up to zero. Condition 3 describes the properties which hold at compressor stations. The first equation in this formulation denotes the pressure increase. The second equation $u_v^t = \beta_v^t \cdot u_v^t$ states that pressure can only increase if the compressor is on. Condition 4 are non-negativity constraints for pressure. Equation 5 denotes the condition at the node which is connected to the slack bus and leads off flux to it.

6.3.2. Optimization Problem with POC

We will reformulate the last optimization problem using Partial Outer Convexification introduced in Chapter 3. Here we additionally include binary values (*convex multiplier controls*) w^t which fulfill the SOS1 condition with $w^t \in \{0, 1\}^{n_{oc}}$, where $n_{oc} = 2^{n_{com}}$ is the number of convex multiplier controls. Furthermore we denote c as the set of all possible switch combinations (see Chapter 3). With c_v^s we denote switch combination s at compressor node v . Then our POC reformulated Problem reads as follows:

$$\min_u \quad \frac{1}{2} \sum_{t=0}^m \sum_{v \in V_c} (u_v^t)^2$$

subject to:

Weymouth Equation

$$\begin{aligned} 1. \quad & \rho_j^{t+1} = \rho_j^t - \frac{\Delta t}{\Delta x} (q_j^t - q_{j-1}^t) \\ & q_j^{t+1} = q_j^t - a^2 \frac{\Delta t}{\Delta x} (\rho_j^t - \rho_{j-1}^t) - \Delta t \frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t} \\ & \forall j = 1, \dots, n, \forall t = 0, \dots, m-1, \text{ for all pipes} \end{aligned} \tag{6.8}$$

2. gas coupling conditions

for a fix but arbitrary innode $v \in V \setminus (V_q \cup V_p \cup V_c)$ it holds:

$${}^e p_n^t = p(v, t), \quad \forall e \in \delta^-(v), \forall t = 0, \dots, m$$

$${}^h p_0^t = p(v, t), \quad \forall h \in \delta^+(v), \forall t = 0, \dots, m$$

for every node $v \in V \setminus (V_q \cup V_p)$ it holds:

$$\sum_{e \in \delta^-(v)} {}^e q_n^t = \sum_{h \in \delta^+(v)} {}^h q_0^t, \quad \forall t = 0, \dots, m$$

3. compressor conditions

$$\begin{aligned}
 {}^e p_n^t &= {}^h p_0^t + u_v^t, \quad \forall v \in V_c, e \in \delta^-(v), h \in \delta^+(v), \forall t = 0, \dots, m \\
 u_v^t &= \sum_{s=1}^{n_{oc}} w_s^t c_v^s u_v^t, \quad \forall v \in V_c, \forall t = 0, \dots, m \\
 \sum_{s=1}^{n_{oc}} w_s^t &= 1 \quad (\text{SOS-Type1})
 \end{aligned} \tag{6.9}$$

4. non - negativity constraints

$$\begin{aligned}
 u_v^t &\geq 0, \quad \forall v \in V_c, \forall t = 0, \dots, m \\
 p_j^t &\geq 0, \quad \forall j = 0, \dots, n, \forall t = 0, \dots, m, \text{ for all pipes}
 \end{aligned}$$

5. gas - power coupling at node s

$${}^e q_n^t = {}^h q_0^t + \epsilon_t, \quad e \in \delta^-(s), h \in \delta^+(s) \setminus h_{slack}, \forall t = 0, \dots, m$$

Initial conditions: $\rho_j^0, q_j^0, \forall j = 0, \dots, n$, for each pipe and $\epsilon^t, \forall t = 0, \dots, m$

Solution: $\rho_j^t, q_j^t, \forall j = 0, \dots, n, t = 0, \dots, m$ for each pipe, $u_v^t, \forall v \in V_c$ and configuration $w_s^t, \forall t = 0, \dots, m$.

The optimization problem is the same as the optimization problem without POC reformulation, except for condition 3.

The second equation in condition 3 states that pressure can only increase if the compressor is on. The third equation in condition 3 states that only one of the possible multiplier controls is active at time t .

6.4. Additional Constraints

Additional Constraints: Type 1

In practice additional constraints have to be considered to achieve realistic results. The additional constraints that we will introduce in this section ensure that compressors hold their state for a minimum given time. These constraints have to be added to the optimization problems. These are two types of constraints, which were also used in [1]. The first inequality ensures that the compressor has to keep its state for at least a time of M_1 after switching from off to on

$$\sum_{e=k+1}^{k+\left\lfloor \frac{M_1}{\Delta t} \right\rfloor} \sum_{s=1}^{n_{oc}} c_v^s w_s^e \geq \sum_{s=1}^{n_{oc}} c_v^s \left[\frac{M_1}{\Delta t} \right] (-w_s^k + w_s^{k+1}) \tag{6.10}$$

for all $k \leq m - \left\lfloor \frac{M_1}{\Delta t} \right\rfloor$ for every compressor node $v \in V_c$.

The second inequality ensures that the compressor has to keep its state for at least a time of M_2 after switching from on to off

$$\sum_{e=k+1}^{k+\left\lfloor \frac{M_2}{\Delta t} \right\rfloor} \left(1 - \sum_{s=1}^{n_{oc}} c_v^s w_s^e \right) \geq \sum_{s=1}^{n_{oc}} c_v^s \left[\frac{M_2}{\Delta t} \right] (w_s^k - w_s^{k+1}) \quad (6.11)$$

for all $k \leq m - \left\lfloor \frac{M_2}{\Delta t} \right\rfloor$ for every compressor $v \in V_c$.

Additional Constraints: Type 2

We can also pose another type of additional constraints, where we demand that the compressor at node v can only switch maximum $r \in \mathbb{N}$ number of times in $t = 0, \dots, m$. This can be expressed by

$$\sum_{t=1}^m \sum_{s=1}^{n_{oc}} |w_s^t c_v^s - w_s^{t-1} c_v^s| \leq r, \quad \forall v \in V_c. \quad (6.12)$$

If additional constraints that couple over time are not present, we can solve CIAP by using Sum Up Rounding. Otherwise we have to solve a MILP

$$\begin{aligned} & \min_{\substack{\beta^t \in H \cap \{0,1\}^{n_{oc}}, \\ t=0, \dots, m-1, \\ \delta^0 \in \mathbb{R}^{n_{oc}}, \epsilon \in \mathbb{R}}} \epsilon \\ \text{subject to: } & \left\| \delta^0 + \sum_{t=0}^{k-1} \Delta t (\alpha^t - w^t) \right\|_{\Omega} \leq \epsilon \quad \text{for all } k = 0, \dots, m \\ & [+(6.10) + (6.11)] \text{ or } [+(6.12)] \end{aligned} \quad (6.13)$$

where α^t are the relaxed binary variables in time t .

Note that Theorem 3.1 presented in Chapter 3 does not hold if additional constraints that couple over time are present.

6.5. Showing Theorem 3.1 for Weymouth Equation discretized with Simple Upwind

In this section we consider the requirements of Theorem 3.1 to show its applicability on optimization problem 6.9. We reformulate the Simple Upwind discretization into the form $u^{t+1} = u^t + \Delta t \Phi(t, u)$:

$$\rho_j^{t+1} = \rho_j^t + \underbrace{\Delta t \left(\frac{1}{\Delta x} (q_{j-1}^t - q_j^t) \right)}_{:= \Phi(t, \rho)} \quad (6.14)$$

$$q_j^{t+1} = q_j^t + \underbrace{\Delta t \left(\frac{a^2}{\Delta x} (\rho_{j-1}^t - \rho_j^t) - \frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t} \right)}_{:= \Phi(t, q)}. \quad (6.15)$$

In order to verify the assumptions, we choose the following norms

$$\|u(\cdot)\|_X = \sum_{r \in A} \sum_{j=1}^n |\rho_j^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n |q_j^t| \Delta x \text{ and } \|\cdot\|_\Omega = \|\cdot\|_\infty. \quad (6.16)$$

We denote Ω as the set of all convex multiplier controls with $n_{oc} = |\Omega|$. Furthermore we define A as the set of all arcs.

We start by showing the **first requirement 3.7**:

- (i) to prove : $\|\Phi(t, u)v\|_X \leq M_{oc} \|v\|_\Omega, \quad \forall t \in \mathcal{I}, u = (\rho, q)^T, \rho \in \mathcal{D}^1, q \in \mathcal{D}^2,$
 $\mathcal{D}^1, \mathcal{D}^2$ bounded sets,
 $v \in \mathbb{R}^{n_{oc}}$ and $M_{oc} < \infty$

Proof.

$$\begin{aligned} \|\Phi(t, u)v\|_X &= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \Phi(t, \rho) v_c \right| \Delta x + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \Phi(t, q) v_c \right| \Delta x \\ &= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \frac{1}{\Delta x} (q_{j-1}^t - q_j^t) v_c \right| \Delta x \\ &\quad + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left(\frac{a^2}{\Delta x} (\rho_{j-1}^t - \rho_j^t) - \frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t} \right) v_c \right| \Delta x \end{aligned}$$

We pose a strong assumption on ρ_j^t and q_j^t , $\forall j = 0, \dots, n$ and $\forall t = 0, \dots, m$

$$\begin{aligned} 0 < \rho_{min} &\leq \rho_j^t \leq \rho_{max} < \infty \\ 0 &\leq q_{min} \leq q_j^t \leq q_{max} < \infty \end{aligned} \quad (6.17)$$

for each arc $r \in A$. Furthermore in practice it holds for the friction factor $\lambda \geq 0$. We use estimation 6.17 in the following

$$\begin{aligned}
\|\Phi(t, u)v\|_X &= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \frac{1}{\Delta x} \underbrace{\left(q_{j-1}^t - q_j^t \right)}_{\leq q_{max}} v_c \right| \Delta x \\
&\quad + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left(\frac{a^2}{\Delta x} (\rho_{j-1}^t - \rho_j^t) - \underbrace{\frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t}}_{\lambda \geq 0 \geq 0} \right) v_c \right| \Delta x \\
&\leq \sum_{r \in A} \sum_{j=1}^n \frac{q_{max}}{\Delta x} \sum_{c \in \Omega} |v_c| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{a^2 \rho_{max}}{\Delta x} \sum_{c \in \Omega} |v_c| \Delta x \\
&= \sum_{r \in A} n \frac{q_{max}}{\Delta x} \sum_{c \in \Omega} |v_c| \Delta x + \sum_{r \in A} n \frac{a^2 \rho_{max}}{\Delta x} \sum_{c \in \Omega} |v_c| \Delta x \\
&\leq \underbrace{\sum_{r \in A} n_{oc} n (q_{max} + a^2 \rho_{max})}_{:= M_{oc}} \|v\|_\Omega = M_{oc} \|v\|_\Omega
\end{aligned}$$

■

Next we show the **second requirement** 3.8:

(ii) to prove: $\|(\Phi(t, u) - \Phi(t, \tilde{u}))\alpha\|_X \leq L_{oc} \|u - \tilde{u}\|_X, \quad \forall t \in \mathcal{I}, u = (\rho, q), \tilde{u} = (\tilde{\rho}, \tilde{q})$

$$\text{with } \alpha \in H, \text{ where } H = \left\{ \alpha \in \mathbb{R}_{\geq 0}^{n_{oc}} \mid \sum_{c \in \Omega} \alpha_c = 1 \right\}$$

Proof.

$$\begin{aligned}
&\|(\Phi(t, u) - \Phi(t, \tilde{u}))\alpha\|_X \\
&= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} (\Phi(t, \rho) - \Phi(t, \tilde{\rho})) \alpha_c \right| \Delta x + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} (\Phi(t, q) - \Phi(t, \tilde{q})) \alpha_c \right| \Delta x \\
&= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left[\frac{1}{\Delta x} (q_{j-1}^t - q_j^t) - \frac{1}{\Delta x} (\tilde{q}_{j-1}^t - \tilde{q}_j^t) \right] \alpha_c \right| \Delta x \\
&\quad + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left[\frac{a^2}{\Delta x} (\rho_{j-1}^t - \rho_j^t) - \frac{a^2}{\Delta x} (\tilde{\rho}_{j-1}^t - \tilde{\rho}_j^t) - \frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t} + \frac{\lambda}{2D} \frac{\tilde{q}_j^t |\tilde{q}_j^t|}{\tilde{\rho}_j^t} \right] \alpha_c \right| \Delta x \\
&\leq \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left[\frac{1}{\Delta x} (q_{j-1}^t - \tilde{q}_{j-1}^t) + \frac{1}{\Delta x} (\tilde{q}_j^t - q_j^t) \right] \alpha_c \right| \Delta x \\
&\quad + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left[\frac{a^2}{\Delta x} (\rho_{j-1}^t - \tilde{\rho}_{j-1}^t) + \frac{a^2}{\Delta x} (\tilde{\rho}_j^t - \rho_j^t) \right] \alpha_c \right| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{\lambda}{2D} \left| \sum_{c \in \Omega} \left(\frac{\tilde{q}_j^t |\tilde{q}_j^t|}{\tilde{\rho}_j^t} - \frac{q_j^t |q_j^t|}{\rho_j^t} \right) \alpha_c \right| \Delta x
\end{aligned} \tag{6.18}$$

Using $\frac{1}{\Delta x} \leq \frac{a^2}{\Delta x}$ and $\alpha_c \leq 1$ we obtain

$$\begin{aligned}
& \|(\Phi(t, u) - \Phi(t, \tilde{u}))\alpha\|_X \\
& \leq \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} \left| \sum_{c \in \Omega} \left[(q_{j-1}^t - \tilde{q}_{j-1}^t) + (\tilde{q}_j^t - q_j^t) \right] \alpha_c \right| \Delta x \\
& + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} \left| \sum_{c \in \Omega} \left[(\rho_{j-1}^t - \tilde{\rho}_{j-1}^t) + (\tilde{\rho}_j^t - \rho_j^t) \right] \alpha_c \right| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{\lambda}{2D} \left| \sum_{c \in \Omega} \left(\frac{\tilde{q}_j^t |\tilde{q}_j^t|}{\tilde{\rho}_j^t} - \frac{q_j^t |q_j^t|}{\rho_j^t} \right) \alpha_c \right| \Delta x \\
& \stackrel{\alpha_c \leq 1}{\leq} \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |q_{j-1}^t - \tilde{q}_{j-1}^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |\tilde{q}_j^t - q_j^t| \Delta x \\
& + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |\rho_{j-1}^t - \tilde{\rho}_{j-1}^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |\tilde{\rho}_j^t - \rho_j^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{\lambda}{2D} n_{oc} \left| \frac{\tilde{q}_j^t |\tilde{q}_j^t|}{\tilde{\rho}_j^t} - \frac{q_j^t |q_j^t|}{\rho_j^t} \right| \Delta x
\end{aligned}$$

In order to find an upper bound for the last term in the previous equation, we have to pose another strong assumption: There exists $0 < C_1 < \infty$, $\forall j = 1, \dots, n$ such that

$$\left| \frac{\tilde{q}_j^t |\tilde{q}_j^t|}{\tilde{\rho}_j^t} - \frac{q_j^t |q_j^t|}{\rho_j^t} \right| \leq C_1 |(\tilde{q}_j^t - q_j^t) + (\tilde{\rho}_j^t - \rho_j^t)|.
\quad (6.19)$$

This assumption can be achieved if we choose ρ_{min}, ρ_{max} and q_{min}, q_{max} as given in 6.17, meaning ρ should not be zero or too close to zero and q_{max} should not be $+\infty$. With assumption 6.19 we obtain

$$\begin{aligned}
& \|(\Phi(t, u) - \Phi(t, \tilde{u}))\alpha\|_X \\
& \leq \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |q_{j-1}^t - \tilde{q}_{j-1}^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |\tilde{q}_j^t - q_j^t| \Delta x \\
& + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |\rho_{j-1}^t - \tilde{\rho}_{j-1}^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{a^2}{\Delta x} n_{oc} |\tilde{\rho}_j^t - \rho_j^t| \Delta x \\
& + \sum_{r \in A} \sum_{j=1}^n \frac{\lambda}{2D} n_{oc} C_1 |\tilde{q}_j^t - q_j^t| \Delta x + \sum_{r \in A} \sum_{j=1}^n \frac{\lambda}{2D} n_{oc} C_1 |\tilde{\rho}_j^t - \rho_j^t| \Delta x \\
& \leq \frac{2a^2}{\Delta x} n_{oc} \|u - \tilde{u}\|_X + \frac{\lambda}{2D} n_{oc} C_1 \|u - \tilde{u}\|_X = \underbrace{\left(\frac{2a^2}{\Delta x} n_{oc} + \frac{\lambda}{2D} n_{oc} C_1 \right)}_{:= L_{oc}} \|u - \tilde{u}\|_X.
\end{aligned}$$

■

Finally we show the **third requirement** 3.10:

$$\text{(iii) to prove : } \|(\Phi(t+1, u) - \Phi(t, u))v\|_X \leq \Delta t C_{oc} \|v\|_\Omega, \quad \forall t \in \mathcal{I}, v \in \mathbb{R}^{n_{oc}}, u = (\rho, q) \quad (6.20)$$

Proof.

$$\begin{aligned}
& \|(\Phi(t+1, u) - \Phi(t, u))v_c\| \Delta x \\
&= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} (\Phi(t+1, \rho) - \Phi(t, \rho))v_c \right| \Delta x + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} (\Phi(t+1, q) - \Phi(t, q))v_c \right| \Delta x \\
&= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \frac{1}{\Delta x} ((q_{j-1}^{t+1} - q_j^{t+1}) - (q_{j-1}^t - q_j^t))v_c \right| \Delta x \\
&\quad + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left[\frac{a^2}{\Delta x} ((\rho_{j-1}^{t+1} - \rho_j^{t+1}) - (\rho_{j-1}^t - \rho_j^t)) - \frac{\lambda}{2D} \left(\frac{q_j^{t+1}|q_j^{t+1}|}{\rho_j^{t+1}} - \frac{q_j^t|q_j^t|}{\rho_j^t} \right) \right] v_c \right| \Delta x \\
&= \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \frac{1}{\Delta x} \underbrace{((q_{j-1}^{t+1} - q_{j-1}^t) - (q_j^{t+1} - q_j^t))}_{= \Delta t \Phi(t, q)} v_c \right| \Delta x \\
&\quad + \sum_{r \in A} \sum_{j=1}^n \left| \sum_{c \in \Omega} \left[\underbrace{\frac{a^2}{\Delta x} ((\rho_{j-1}^{t+1} - \rho_{j-1}^t) - (\rho_j^{t+1} - \rho_j^t))}_{= \Delta t \Phi(t, \rho)} - \frac{\lambda}{2D} \left(\frac{q_j^{t+1}|q_j^{t+1}|}{\rho_j^{t+1}} - \frac{q_j^t|q_j^t|}{\rho_j^t} \right) \right] v_c \right| \Delta x
\end{aligned}$$

We need another assumption which is similiar to 6.19: There exists $0 < C_2 < \infty$, $\forall j = 1, \dots, n$ such that

$$\left| \frac{q_j^{t+1}|q_j^{t+1}|}{\rho_j^{t+1}} - \frac{q_j^t|q_j^t|}{\rho_j^t} \right| \leq C_2. \quad (6.21)$$

Thus assumption can be again achieved if we pose boundaries on ρ and q , such that ρ is not zero or close to zero and q is not $+\infty$. With the first requirement (shown in the first proof), assumption 6.21 and $\frac{1}{\Delta x} \leq \frac{a^2}{\Delta x}$ we obtain

$$\begin{aligned}
& \|(\Phi(t+1, u) - \Phi(t, u))v_c\| \Delta x \leq \left(\frac{2a^2}{\Delta x} \Delta t M_{oc} + \frac{\lambda}{2D} C_2 \right) \|v\|_\Omega \\
&= \Delta t \underbrace{\left(\frac{2a^2}{\Delta x} M_{oc} + \frac{\lambda}{\Delta t 2D} C_2 \right)}_{=: C_{oc}} \|v\|_\Omega
\end{aligned}$$

■

Notice that due to assumption 6.17, we could quiet easily proof the three theorem requirements. We will also provide boundries on ρ_j^t and q_j^t for all $j = 0, \dots, n$ and $t = 0, \dots, m$ in our implementation. Providing boundaries is a reasonable choice as some optimization solutions for q and ρ , that we obtained numerically, are too large to be used in practice. On the other hand restricting q to non-negativity is not always satisfied in practice, as flux can also be negative and flow into another direction.

7. Numerical Analysis

All computations in this chapter have been performed on a PC equipped with 16 GB RAM and Intel Quad-Core i5, 2,4 GHz CPU. The code has been run in Python 3.7.6.

All NLPs have been solved with IPOPT version 3.12.3 [18] using linear solver *ma27* integrated into CasADi [29], version 3.5.5. CasADi is an open-source software tool for numerical optimization. The linear solver *ma27* was obtained with a personal licence from the HSL website [30]. The reason for taking *ma27* instead of the default solver *mumps* is its better performance for large number of optimization variables according to user experience shared in the CasADi Google Group and by the CasADi developers themselves.

All MILPs were solved with Gurobi Optimizer obtained by an academic license, version 9.1.2. All MINLPs were solved with the open-source solver Bonmin [31], which is also integrated into CasADi and uses the NLP-based Branch-and-Bound algorithm (Bonmin denotes this algorithm as B-BB). The B-BB algorithm solves its NLPs again with IPOPT.

7.1. Solving the Power Model

We consider one fix power-network which we will couple with two different types of gas-networks. This network consists of one slack bus, two generator buses and six load buses. We

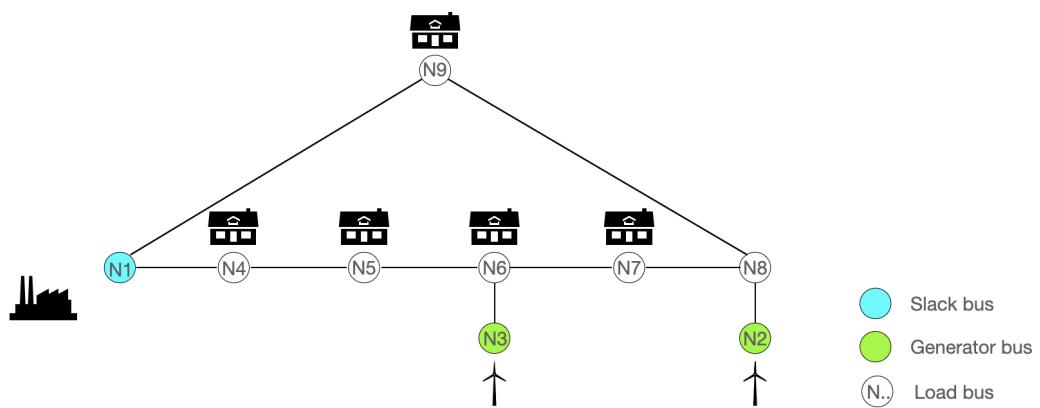


Figure 7.1.: Power Network with one slack bus, two generator buses and six load buses

choose B as our susceptance and G as our conductance matrix, where entries G_{jk} or B_{jk} denote the conductance or susceptance between bus j and bus k . G_{kk} or B_{kk} denote the self-conductance

or self-susceptance. Both matrices B and G are symmetric. The entries for these matrices are shown in Table 7.1 and Table 7.2. We use a per-unit system, whose base power and voltage are 100 MW and 345 kV respectively.

Node	1	2	3	4	5	6	7	8	9
1	-17.3611	0	0	17.3611	0	0	0	0	0
2	0	-16	0	17.3611	0	0	0	16	0
3	0	0	-17.0648	0	0	17.0648	0	0	0
4	17.3611	0	0	-39.3089	10.5107	0	0	0	11.6041
5	0	0	0	10.5107	-15.8409	5.5882	0	0	0
6	0	0	17.0648	0	5.5882	-32.1539	9.7843	0	0
7	0	0	0	0	0	9.7843	-23.3032	13.698	0
8	0	16	0	0	0	0	13.698	-35.4456	5.9751
9	0	0	0	11.6041	0	0	0	5.9751	-17.3382

Table 7.1.: Susceptance matrix B

Node	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	3.3074	-1.9422	0	0	0	-1.3652
5	0	0	0	-1.9422	3.2242	-1.282	0	0	0
6	0	0	0	0	-1.282	2.4371	-1.1551	0	0
7	0	0	0	0	0	-1.1551	2.7722	-1.6171	0
8	0	0	0	0	0	0	-1.6171	2.8047	-1.1876
9	0	0	0	-1.3652	0	0	0	-1.1876	2.5528

Table 7.2.: Conductance matrix G

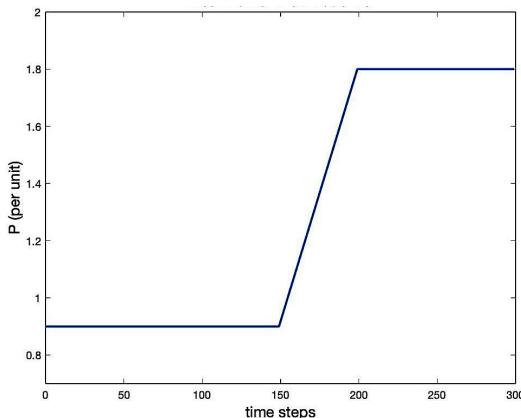
As mentioned in Chapter 5.3, real and reactive power are known at load bus. At generator bus real power and voltage amplitude are known. At slack bus voltage amplitude and phase ϕ are known. In Table 7.3 we provide initial conditions for the power-network 7.1 in $t = 0$.

Node	P	Q	V	ϕ
N1	-	-	1	0
N2	163	-	1	-
N3	85	-	1	-
N4	0	0	-	-
N5	-90	-30	-	-
N6	0	0	-	-
N7	-100	-35	-	-
N8	0	0	-	-
N9	-125	-50	-	-

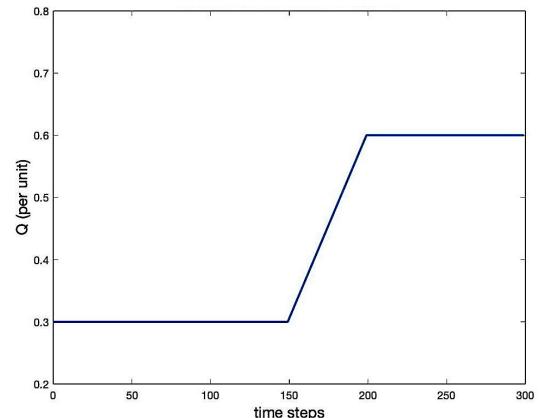
Table 7.3.: Initial Values at $t = 0$

All values for the power grid, including susceptance (Table 7.1) and conductance (Table 7.2) matrices as well the initial values (Table 7.3) are taken from the Paper *Optimal Control of Compressor Stations in a Coupled Gas-to-Power Network* [32].

Except for real and reactive power P, Q at node N5 the initial values presented in Table 7.3 remain constant over time. In order to simulate a sudden increase in power demand we increase the real power from 0.3 p.u. (per unit) to 0.6 p.u. and the reactive power from 0.9 p.u. to 1.8 p.u at node N5. Given time steps $t = 0, \dots, m$, we let the real and reactive power increase in the interval $[m/2, m/2 + 50]$. The change in real and reactive power at node N5 is shown in the following figure.



(a) Real power P at node N5



(b) Reactive power Q at node N5

Inserting all initial values into the power flow equations presented in Chapter 5.3 we obtain a non-linear system of 9 equations and 9 unknowns for each time step t . We solve these equations in Matlab with the help of the build-in function *fsolve* using Levenberg-Marquardt Algorithm (see Chapter 4.1.2). Then, for every time step and every bus we obtain the values $P, Q, |V|$ and ϕ . The result for real power in N1 obtained with Levenberg-Marquardt is presented in Figure 7.3.

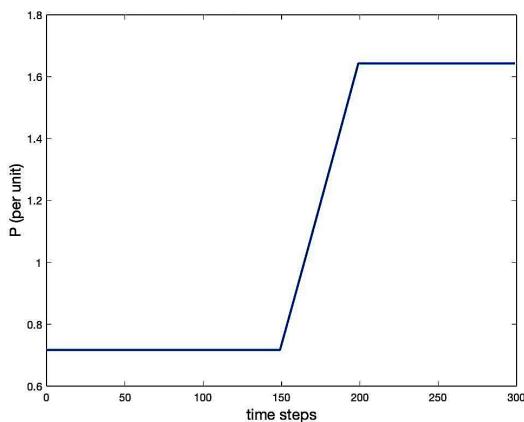


Figure 7.3.: Results after applying *fsolve*: Real power P in node N1

In the last step power need to be converted gas. For conversion we use, as in [32], the quadratic function

$$\epsilon = a_0 + a_1 P + a_2 P^2 \quad (7.1)$$

where P is again the real power at slack bus (here: N1), ϵ is the mass flow and a_0, a_1, a_2 are conversion constants. The larger the constants a_0, a_1 and a_2 the more gas is being taken from the gas network.

7.2. Simple Model: One Compressor

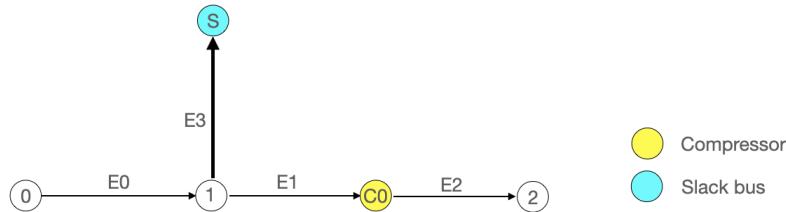


Figure 7.4.: Simple Model

We start with a simple model where only one compressor is integrated into the gas net. For the sake of simplicity we choose first $\epsilon^t = 0, \forall t = 0, \dots, m$ - thus there is no outflow from the gas network to the power grid. We choose $\lambda = 0.11$ as a constant friction factor in every pipe. We set $D = 1 \text{ m}$ as the diameter of the pipe and $a = 340 \text{ m/s}$ as flux speed. Furthermore we choose the length of one pipe to be 12000 m (12 km) and a total execution time of 5 sec with time step size $\Delta t = 1/60$ and spatial step size $\Delta x = 2000$. The CFL condition is fulfilled since

$$\Delta t \cdot a^2 \leq \Delta x \iff \frac{1}{60} \cdot (340)^2 = 1926.6 \leq 2000. \quad (7.2)$$

We choose our initial values to be $p(0, x) = 60 \text{ bar}$ and $q(0, x) = 500 \text{ (kg/s)}$ for each pipe at every pipe section. Note that for the initial values we have provided pressure instead of density. The Weymouth equation however works with density ρ instead of pressure p . Thus we recalculate our initial pressure into density according to the following example.

Consider a pressure of 60 bar . It holds $60 \text{ bar} = 6000000 \text{ Pascal} = 6000000 \text{ kg/(m} \cdot \text{s}^2\text{)}$. With the relation $p(\rho) = a^2 \rho$ and $a^2 = 340 \text{ m/s}$ we obtain the density:

$$\rho = \frac{6000000 \text{ kg/(m} \cdot \text{s}^2\text{)}}{a^2} = \frac{6000000 \text{ kg/(m} \cdot \text{s}^2\text{)}}{340 \text{ m}^2/\text{s}^2} \approx 51.9 \frac{\text{kg}}{\text{m}^3}. \quad (7.3)$$

Before starting to solve the MINLPs we run an NLP model without binary constraints in order to test if our initial conditions are reasonable and a solution can be found. This is indeed the case for the initial conditions provided above. We compare three different models:

the three-step approach and the direct solver bonmin, which solves a POC reformulated and a 'non-POC' reformulated optimization problem. Here, no additional constraints that couple over time are present. In all three cases an optimal solution is found, where both the Kirchhoff-Type conditions and the Weymouth equation are satisfied. While testing the three-step approach I noticed that this method converges either to an infeasible solution or produces very large values for p and q if pressure p and mass flow q are not restricted with boundaries. Therefore the following additional constraints are set: $\rho \in [0, 100]$, $q \in [-1000, 1000]$.

Graphical results for mass flux and pressure at source node 0 and sink node 2 as well pressure increase u_0 at compressor C0 and the binary variable β_0 are presented in Figure 7.5 - 7.7. The execution time and the objective function value for all three approaches can be found in Table 7.4 - 7.6. For all three examples we can see that the compressor is barely working, since no additional pressure is needed.

OFV	time total (sec)
$1.6350264 \cdot 10^{-7}$	107.35

Table 7.4.: Results for direct solver Bonmin (no POC Reformulation)

OFV	time total (sec)
$1.64764 \cdot 10^{-7}$	176.13

Table 7.5.: Results for direct solver Bonmin (POC Reformulation)

Step 1		Step 2		Step 3		
OFV	time (sec)	time (sec)	OFV	time (sec)	total time (sec)	
$6.94157 \cdot 10^{-19}$	6.5365	0.006	$3.1675 \cdot 10^{-7}$	1.05	7.59	

Table 7.6.: Results for three-step approach

In terms of execution time, the three-step approach was the fastest as it took only 7.59 sec, the direct solver Bonmin with POC reformulation in comparison was 20-times slower with a total execution time of 176.13 sec. Bonmin without POC reformulation was 14-times slower than the three-step approach with a total execution time of 107.13 sec.

We observe that the direct solver without POC reformulation delivered the lowest objective function value with $1.6350264 \cdot 10^{-7}$ which is not significantly smaller than optimization results obtained by the two other problems which are also in the range of 10^{-7} .

Another important observation is the behaviour of the compressor and the control variable u_0 for the three-step method. Here, the compressor switches its state from on to off every time step, which also lets u_0 jump in every time step. In reality, we cannot afford to have the

compressor switching so often. By posing additional constraints on the compressors state that couple over time, we hope to bypass this behaviour. We will work on this in Chapter 7.4.

We notice that pressure and mass flow at sink node 2 are the same for all three approaches. Pressure and mass flow at the source node however are only equal for direct solver bonmin with POC reformulation and the three-step method. We observe that the mass flow at source node in bonmin (no POC) case is different from the one obtained by approaches with POC reformulation. An explanation for this difference can be found when taking again a closer look at the Weymouth equations

$$\rho_j^{t+1} = \rho_j^t - \frac{\Delta t}{\Delta x} (q_j^t - q_{j-1}^t) \quad (7.4)$$

$$q_j^{t+1} = q_j^t - a^2 \frac{\Delta t}{\Delta x} (\rho_j^t - \rho_{j-1}^t) - \Delta t \frac{\lambda}{2D} \frac{q_j^t |q_j^t|}{\rho_j^t}. \quad (7.5)$$

Assume that we want to determine mass flow in time step 2, spacial step 1 denoted as q_1^2 . According to equation 7.5 no point in q_0^1 is required. However q_0^1 is needed when calculating density ρ_1^2 in equation 7.4. Consider the same step sizes $\Delta t = 1/60$, $\Delta x = 2000$ as above. Then $\frac{\Delta t}{\Delta x} = 8,33 \cdot 10^{-6}$ and considering that q is in range -1000 to 1000 (kg/s) we notice that for determining ρ_1^2 mass flow q_0^1 plays almost no role. This affects the simulation in the sense that we cannot see a proper effect of the mass flow on pressure in the short time span of 2 seconds that we simulate.

In our case it seems that no additional flux at source node is needed to satisfy the optimization conditions. In particular (in our case) ρ_j^{t+1} only depends on the ρ_j^t . Thus considering a fix j the pressure values ρ_j^t for $t = 1, \dots, m$ do not vary too much from its initial conditions ρ_j^0 .

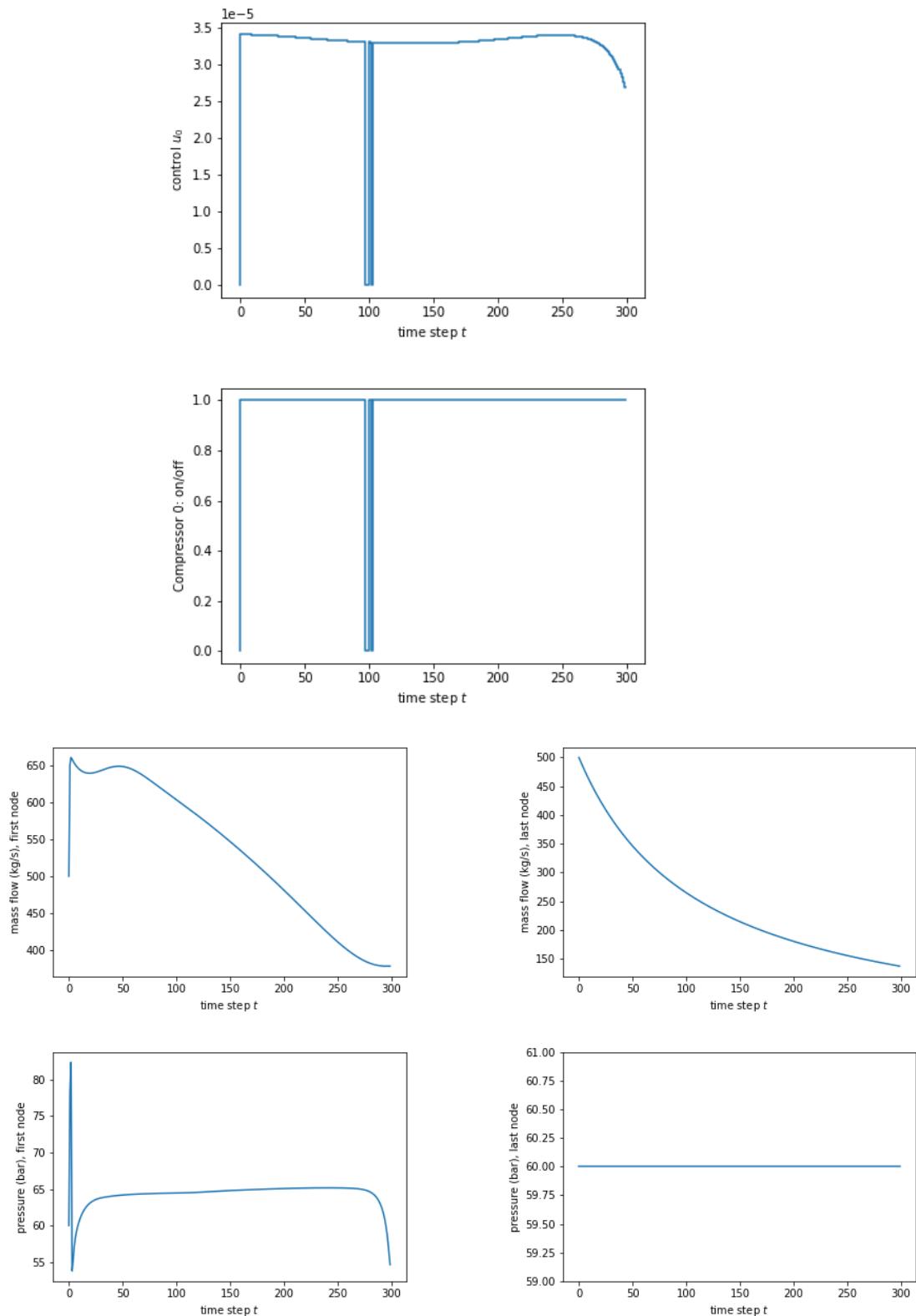


Figure 7.5.: Results for direct solver Bonmin (no POC Reformulation)

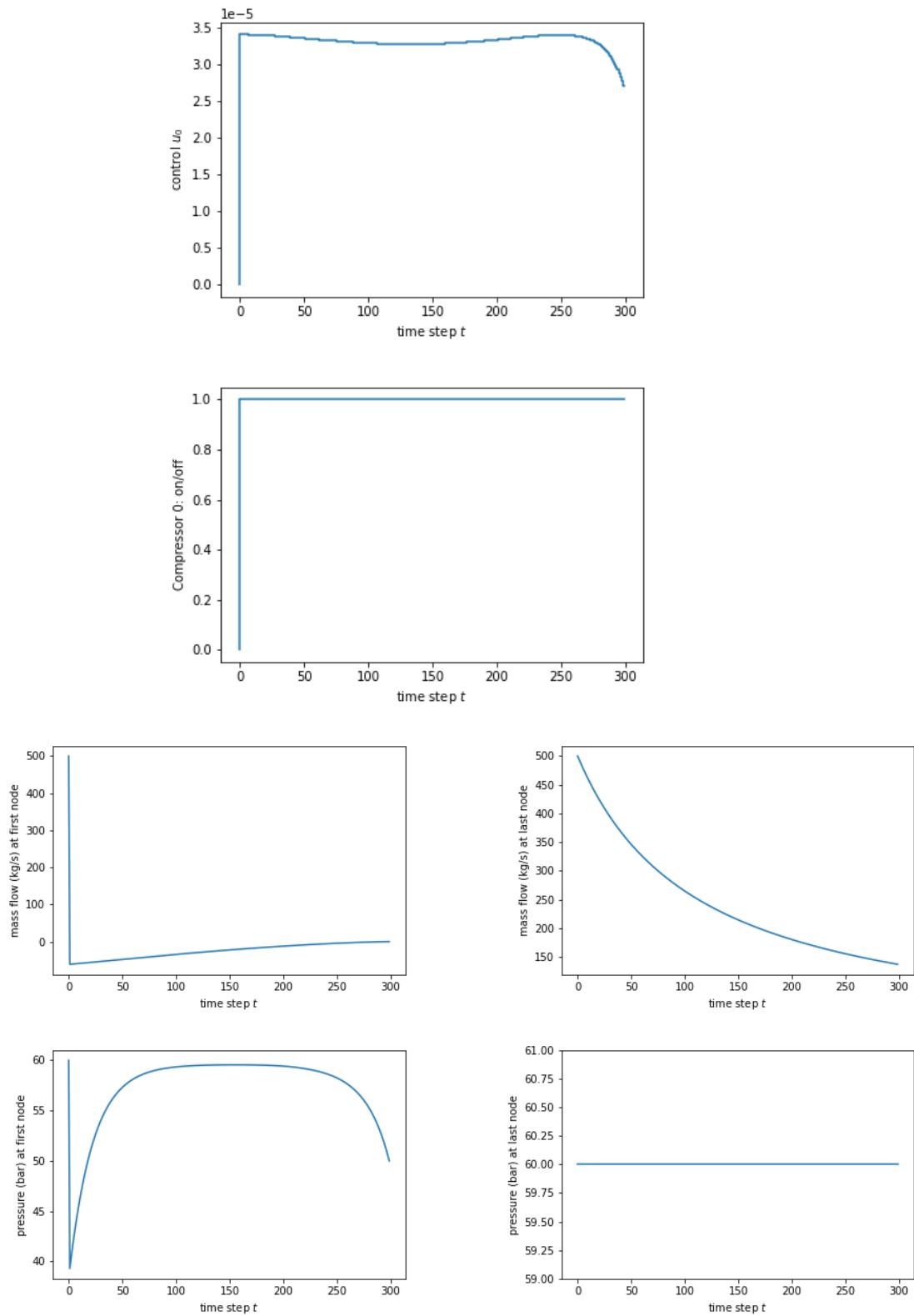


Figure 7.6.: Results for direct solver Bonmin (POC Reformulation)

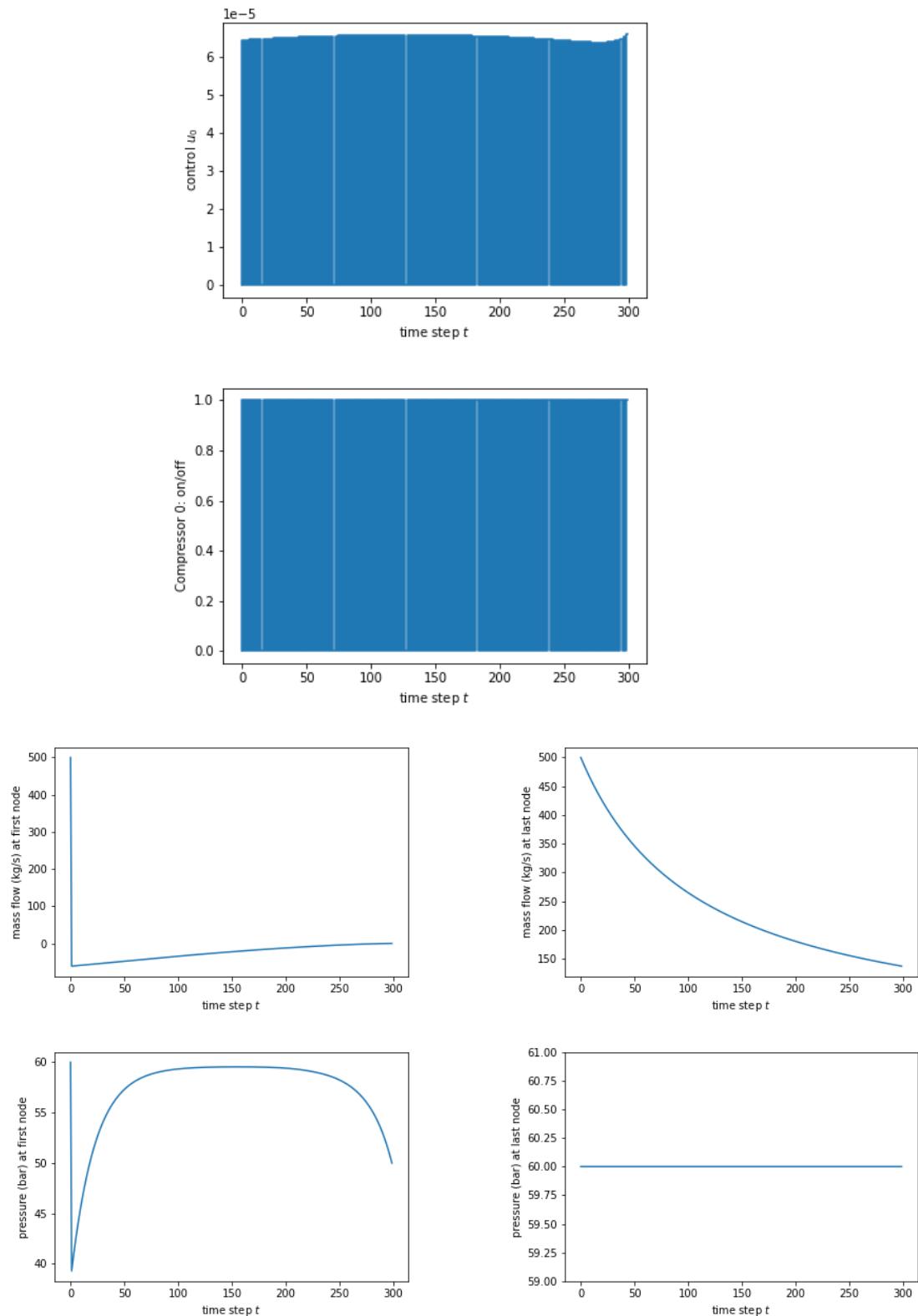


Figure 7.7.: Results for three-step approach (POC Reformulation)

7.3. Advanced Model: Two Compressors

We move on to a more advanced network which contains two compressors. We choose $\lambda = 0.11$

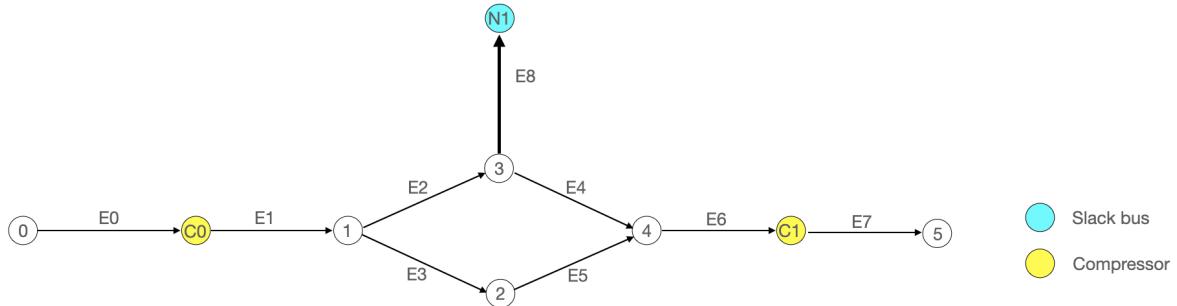


Figure 7.8.: Advanced Model

as the friction factor, diameter $D = 1 \text{ m}$ and flux speed $a = 340 \text{ m/s}$. Same as before the time step size is $\Delta t = 1/60$ and $\Delta x = 2000$ with 12000 m pipe length and a total simulation time of 5 sec . Consider again the real power in node $N1$ in Figure 7.3 and the quadratic equation 7.1. We set $a_0 = 2$, $a_1 = 5$ and $a_2 = 5$ and calculate the mass flow ϵ taken from the grid network using the quadratic function displayed in Figure 7.9.

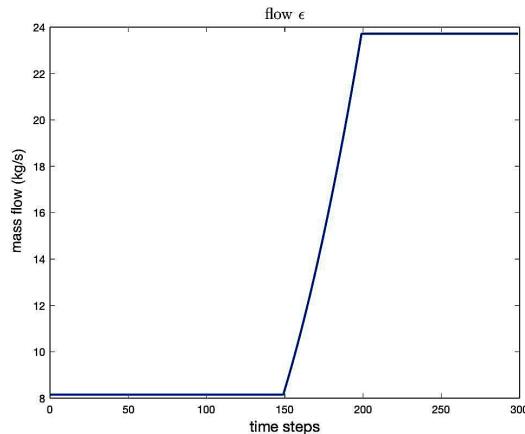


Figure 7.9.: Mass flow ϵ taken out of gas grid

We provide initial conditions for pressure and flux in $t = 0$ as follows:

	E0	E1	E2, E3	E4	E5	E6, E7
$p(x, 0) (\text{bar})$	60	62	62	62	62	62
$q(x, 0) (\text{kg/s})$	500	500	250	241.847	250	491.847

Table 7.7.: Initial Values for Advanced Model at $t = 0$

Note that we choose the initial conditions such that in $t = 0$ the compressor $C0$ is forced to increase pressure by 2 bar . Notice also that mass flow $q(x, 0)$ in $E4$ is uneven because gas is

taken out of the network in time $t = 0$. Same as for the simple model we first test a NLP model without integers, to ensure that our initial conditions are reasonable and a feasible solution can be found. The NLP model was solved "*to acceptable level*". This indicates that the problem did not converge to the "desired" tolerance (10^{-8} in Equation 4.42), but that it was able to obtain a point satisfying the "acceptable" tolerance level (10^{-6}). We start with the three-step approach without using additional constraints that couple over time. For the three step approach we restrict IPOPT to a maximum number of 1100 iterations. The optimization problem with the three-step approach is also being solved "*to acceptable level*". The objective function value for the

Step 1		Step 2		Step 3		
OFV	time (sec)	AC	time (sec)	OFV	time (sec)	total time (sec)
2.4189253	1182.81	no	0.006	2.0000	76.212	1259.03

Table 7.8.: Results for three-step approach

three-step approach as well as the time duration of each step is shown in Table 7.8.

Trying to solve the advanced model with bonmin turned out to be a difficult task, since no sign of reaction was shown after 60 *min* of running (in this time bonmin has not even started to run). Now this "freezing behaviour" did also appear in the simple model but lasted only 3 – 5 *min*. Setting different margins on p and q wasn't helpful either. So both for the POC and the non-POC reformulated model we unfortunately did not obtain any results with the direct solver. From the numerical results we can deduce, that the direct solver bonmin integrated into CasADI copes quite well with small networks (like simple model in the previous section) but its computational time starts to grow superlinear when the network expands.

The numerical results obtained with the three-step approach are shown in Figure 7.10. As we can see the compressor switches its state around 7 times within 5 *sec*. Furthermore observing our data we remark that if the compressor is off there is no pressure increase. This is clearly visible in the interaction between compressor C1 and u_1 . For C0 and u_0 the figure might be misleading, since u_0 is not 0 at all time steps $t = 1, \dots, 300$. Further we notice that compressor 0 bypasses the pressure difference of 2 *bar* which we have provided for the initial data (see Table 7.7).

In the next section we are going to investigate how additional constraints that couple over time can prevent the compressor from switching too often and how our results change for both the simple and advanced model.

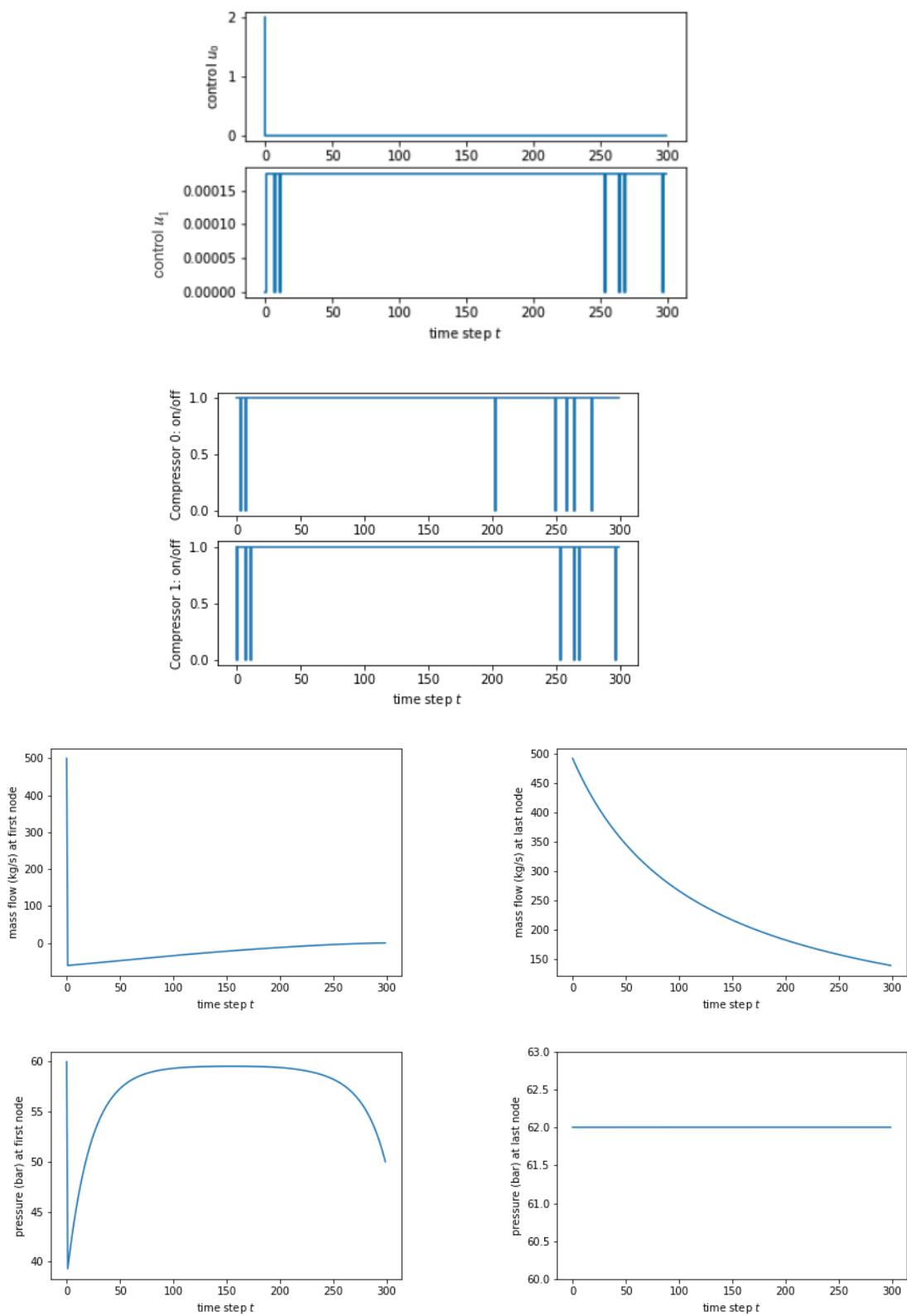


Figure 7.10.: Results for three-step method (POC Reformulation)

7.4. Additional Constraints

When solving the three-step approach with additional constraints that couple over time we cannot solve CIAP with Sum Up Rounding anymore. Instead we need to solve a MILP as given in Problem 6.13. We solve the MILP Problem with gurobipy which uses linear-programming branch-and-bound algorithm.

7.4.1. Simple Model: One Compressor

Additional Constraints in Equation 6.12

We start with additional constraints in 6.12 that demand that our compressor can only switch maximum $r \in \mathbb{N}$ number of times in $t = 0, \dots, m$. We obtain the same results for all $r = 5, r = 10$ and $r = 50$, namely that the compressor should be off all the time. The graphical results for

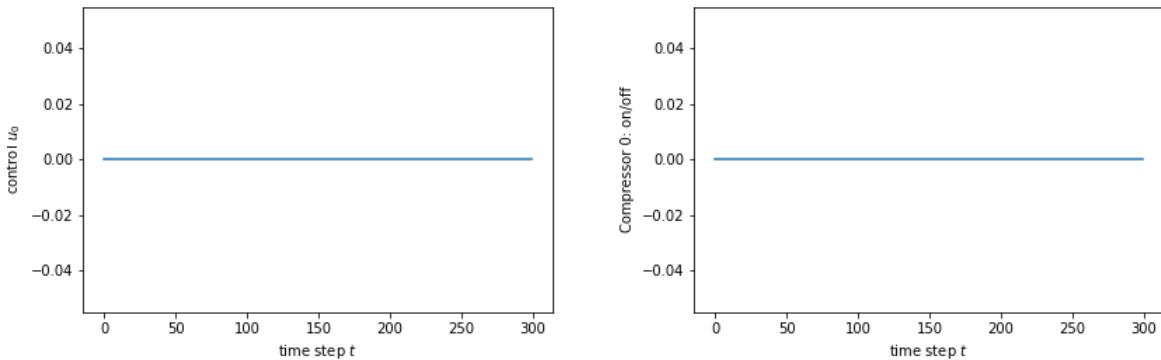


Figure 7.11.: Results for three-step approach with additional constraints 6.12

control variable u_0 and compressors state are displayed in Figure 7.11. Comparing these results to Figure 7.7 we obtain the same pressure and mass flow graphics at source and sink node. At the same time we prevent the compressor from switching every time step. In this case the compressor is off for all time steps. The compressor state calculated with bonmin however was on for almost all time steps. The execution time and the objective function value can be found in Table 7.9.

Step 1		Step 2		Step 3		
OFV	time (sec)		time (sec)	OFV	time (sec)	total time
$6.94157 \cdot 10^{-19}$	6.078		5.663	$1.49996 \cdot 10^{-14}$	0.6654	12.4066

Table 7.9.: Results for three-step approach with additional constraints 6.12

Comparing these results to the results obtained for direct solver bonmin in Table 7.4 - 7.5 we see that with additional constraints for the three-step approach we have found the lowest

objective function value which is $1.49996 \cdot 10^{-14}$. The lowest objective function value found in previous chapter was $1.6350264 \cdot 10^{-7}$ for bonmin with no POC reformulation. We did not only obtain the smallest objective function value, but also the smallest execution time. The execution time here is 12.4066 sec which is still at least 9 times faster than the time needed to run bonmin. The three-step approach with additional constraints performs 9 times faster than bonmin with POC - (see Table 7.4) and 15 times faster than bonmin without POC reformulation (see Table 7.5).

Additional Constraints in Equation (6.10) + (6.11)

We move on to additional constraints presented in (6.10 + 6.11). These constraints require that the compressor keeps its state for at least a time M_1 (after switching from off to on) or M_2 (after switching from on to off). We set $M_1 = M_2 = 1/20$, thus we demand the compressor to keep its state for at least 3 steps. Here we receive the same control and compressor state results as in Figure 7.11. We obtain also the same OFV as in 7.9. Considering execution time this optimization is 6 sec longer than the case with additional constraints 6.12.

Step 1		Step 2		Step 3		
OFV	time (sec)	time (sec)		OFV	time (sec)	total time
$6.94157 \cdot 10^{-19}$	8.5182	8.7786		$1.49996 \cdot 10^{-14}$	0.9155	18.2124

Table 7.10.: Results for three-step approach with additional constraints (6.10) + (6.11)

7.4.2. Advanced Model: Two Compressors

Additional Constraints in Equation 6.12

We proceed with an advanced model containing two compressors. We choose the same initial conditions as given in Table 7.7 and same ϵ as is given in Figure 7.9. We demand that our compressor can only switch $r = 10$ number of times.

Now, the case that we obtain here is very interesting. Since we have to abort the optimization after 1100 iterations, the relaxed control variables w^t are not always the same. We obtain two cases. In the first case, Step 3 leads to a solution with *acceptable level*. In the second case, Step 3 converges to a point of local infeasibility. There is a significant difference between the two cases when we take a closer look at the relaxed control variables in Step 1. In the first, feasible case the w_s^t variables are in general distributed towards the c_4 configuration (both compressors are on) whereas in the second, infeasible case the control variables are distributed more towards the c_1 (both compressor are off) and c_3 (compressor one is on, compressor two is off) configuration. However the case where a feasible solution is found to be more frequent.

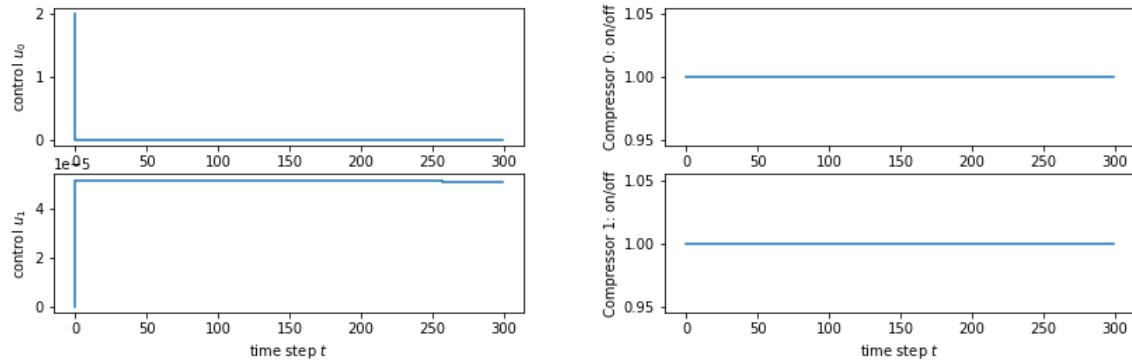


Figure 7.12.: Results for three-step approach with additional constraint 6.12

In the case where the problem is solved to *acceptable level*, we obtain the values for control variable u_0 and compressors state as shown in Figure 7.12. Both compressors are on for every time step t . The pressure and mass flow values at sink and source node are the same as shown in Figure 7.10. We have obtained the same OFV as in the case where no additional constraints were present but in this case the execution time was about 700 sec longer.

Step 1		Step 2		Step 3		
OFV	time (sec)		time (sec)	OFV	time (sec)	total time
2.4189253	1798.61		19.86	2.00000	123.16	1941.63

Table 7.11.: Results for three-step approach with additional constraint 6.12

Additional Constraints in Equation (6.10) + (6.11)

We will also consider a second type of additional constraints which demand that the compressor keeps its state for at least 3 time steps before switching to another state. Therefore we set $M1, M2 = 1/20$. The objective function value and execution time for these type of additional constraints are presented in Table 7.12. We obtain again the same objective function values as before. The optimal control values u_0, u_1 are shown in Figure 7.13.

Step 1		Step 2		Step 3		
OFV	time (sec)		time (sec)	OFV	time (sec)	total time
2.4189253	1775.24		40.19	2.00000	104.73	1920.15

Table 7.12.: Results for three-step approach with additional constraint (6.10) + (6.11)

The compressor in this case is on for all t except the last 3 time steps where it is off. All in all, for the advanced model we obtain the same objective function values for the three-step

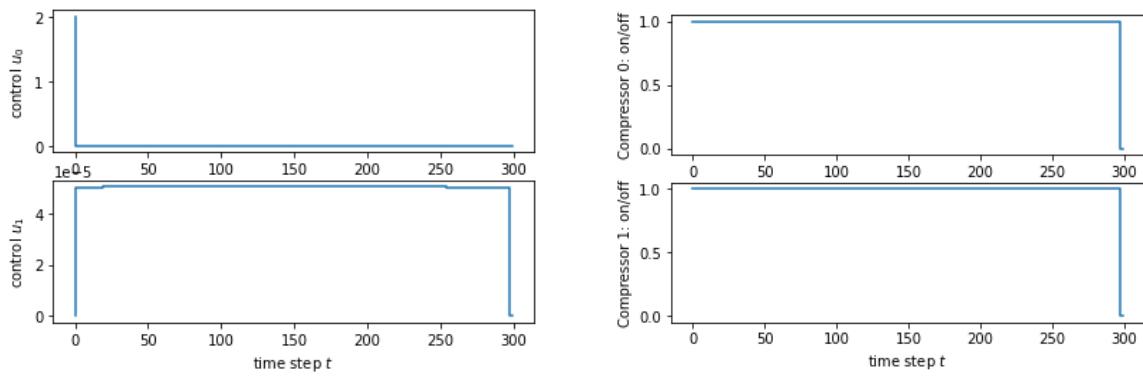


Figure 7.13.: Results for three-step approach with additional constraint 6.12

approach with and without additional time constraints.

For all our optimization models (simple and advanced) solved with additional constraints by the three-step approach, the objective function value either remained the same or even improved compared to the optimization model without additional constraints.

8. Conclusion and Outlook

In this thesis we compared the Euler equations and its simplified version, the Weymouth equations, discretized with Simple Upwind and Lax-Friedrich Scheme. We have shown that for a large amount of time steps the Lax-Friedrich Scheme delivered the best results but also mentioned some concerns regarding gas-network modeling with this scheme. We have also found that the Euler equations are not different from the Weymouth equations for high pressures.

We have derived a gas-to-power model with compressor optimization that uses an explicit discretization of the Weymouth equation. The power model was solved in advance. In order to apply a three-step heuristic based on partial outer convexification we proved the applicability of an approximation result for positive density and positive mass flow. We compared optimization results obtained by the three-step approach with optimization results obtained by direct solver Bonmin for both POC and non-POC reformulated problems. Numerical results for a simple gas network model have shown that the three-step approach works up to 14 times faster in comparison to the direct solver and delivers good objective function values which can be even improved with additional constraints that couple over time. When considering an advanced model we could not find any solution within 1 *h* with a direct solver. Using the three-step approach however an optimal solution could be found within 20 *min*. With additional constraints the three-step approach found an optimal solution within 30 *min*.

In this thesis we have simulated the gas-to-power network for 5 *seconds*. Considering the model from a more practical side an issue arises when we model gas networks with explicit methods. Due to the CFL condition we have to restrict our time step size to a very small value (here $\Delta t = \frac{1}{60}$ sec). If we want to consider realistic simulations for, say, one day (which yields 5184000 time steps), we get a very large model with many optimization variables. Therefore, when modeling gas networks for long time horizons, it may be more appropriate to use an implicit rather than an explicit discretization method. The implicit method, for which the three-step approach provides no approximation results, does not require stability constraints on the time and spatial step size, which also reduces the number of simulation variables.

A. Appendix: Discrete Gronwall Lemma

Lemma A.1 (Discrete Gronwall Lemma([33], p.478)). *Let the numbers ξ_i satisfy estimates of the form*

$$|\xi_{i+1}| \leq (1 + \delta)|\xi_i| + B, \quad \delta > 0, \quad B \geq 0, \quad i = 0, 1, 2, \dots, n$$

then it holds

$$|\xi_n| \leq e^{n\delta} |\xi_0| + \frac{e^{n\delta} - 1}{\delta} B.$$

Proof. For $|\xi_1|$ we receive according to assumption

$$|\xi_1| \leq (1 + \delta)|\xi_0| + B.$$

Then inserting the inequality above into $|\xi_2|$ we obtain

$$\begin{aligned} |\xi_2| &\leq (1 + \delta)|\xi_1| + B \\ &\leq (1 + \delta)((1 + \delta)|\xi_0| + B) + B \\ &= (1 + \delta)^2 |\xi_0| + B(1 + \delta) + B. \end{aligned}$$

For the n -th iterative we obtain

$$\begin{aligned} |\xi_n| &\leq (1 + \delta)^n |\xi_0| + \underbrace{B[1 + (1 + \delta) + (1 + \delta)^2 + \dots + (1 + \delta)^{n-1}]}_{(*)} \\ &= (1 + \delta)^n |\xi_0| + B \frac{(1 + \delta)^n - 1}{\delta} \\ &\leq e^{n\delta} |\xi_0| + \frac{e^{n\delta} - 1}{\delta} B \end{aligned}$$

where $(*)$ is the n -th partial sum of the geometric series. In the last equation we use the estimation: $0 < 1 + \delta \leq e^\delta$ for $\delta > 0$. ■

B. Appendix: Global and Local Minima

Consider the Problem

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subject to: } & x \in X \end{aligned} \tag{B.1}$$

where $X \subseteq \mathbb{R}^n$ is the **feasible set** and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is the **objective function**.

Definition B.1 (Local and Global Minimizer ([12], p.6)). *The feasible point $\bar{x} \in X$ is called*

- a) *local minimizer of Problem B.1, if there exists an $\epsilon > 0$ such that*

$$f(\bar{x}) \leq f(x), \quad \forall x \in X \cap B_\epsilon(\bar{x}) \tag{B.2}$$

with $B_\epsilon(\bar{x}) = \{x \in \mathbb{R}^n : \|x - \bar{x}\| < \epsilon\}$.

- b) *global minimizer of Problem B.1, if*

$$f(\bar{x}) \leq f(x), \quad \forall x \in X. \tag{B.3}$$

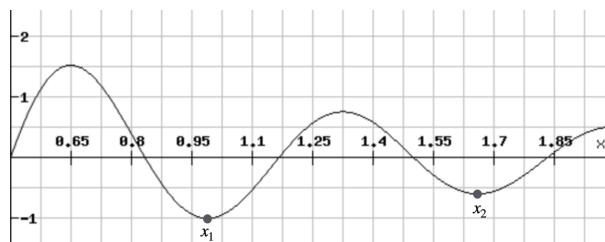


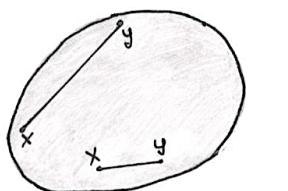
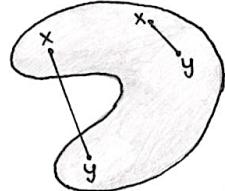
Figure B.1.: in the interval $[0, 2]$ the function has a global minima in x_1 and a local minima in x_2

An important Theorem holds true for convex optimization problems. We will first provide two definitions of convexity, one for functions and one for sets.

Definition B.2 (Convex Sets ([12], p.10)). *A set $X \subseteq \mathbb{R}^n$ is called convex, if*

$$\forall x, y \in X, \forall \lambda \in (0, 1) : (1 - \lambda)x + \lambda y \in X. \tag{B.4}$$

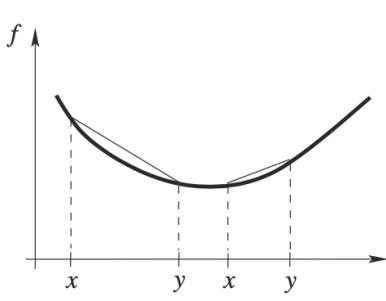
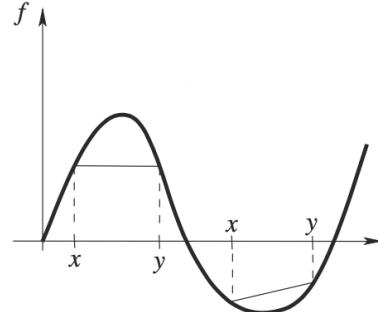
In other words: Points on the line connecting any two points in X belong again to X .

(a) X convex(b) X non-convex**Figure B.2.:** Convexity of sets in \mathbb{R}^2

Definition B.3 (Convex Functions ([12], p.10)). *Let $X \subseteq \mathbb{R}^n$ be a convex set. A function $f : X \rightarrow \mathbb{R}$ is called convex, if*

$$\forall x, y \in X, \forall \lambda \in (0, 1) : f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y). \quad (\text{B.5})$$

For convex functions this means that the graph lies below a line connecting any two points of the function.

(a) f convex(b) f non-convex**Figure B.3.:** Convexity of functions in \mathbb{R} ([17], p.49)

With the two definitions above we can formulate the next Theorem.

Theorem B.4 (Convex Optimization Problems ([12], p.12-13)). *Let $f : X \rightarrow \mathbb{R}$ be a continuously differentiable and convex function and $X \subseteq \mathbb{R}^n$ be a convex set. Then for Problem B.1, every local minimizer of f on X is also a global minimizer of f on X .*

C. Appendix: Karush-Kuhn-Tucker Conditions

Consider the following constrained optimization problem

$$\begin{aligned} & \min_{x \in \mathbb{R}^n} f(x) \\ \text{subject to: } & c_i(x) = 0 \quad \text{for all } i \in \mathcal{E} \\ & c_i(x) \geq 0 \quad \text{for all } i \in \mathcal{I} \end{aligned} \tag{C.1}$$

with \mathcal{E} and \mathcal{I} two finite index sets. We assume that $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $c_i : \mathbb{R}^n \rightarrow \mathbb{R}, i \in \mathcal{E} \cup \mathcal{I}$ are at least once continuously differentiable. First we provide a definition of a *feasible set* and an *active set*.

Definition C.1 (Feasible set ([16], p.40)). *We denote the set*

$$\Omega = \{x \in \mathbb{R}^n : c_i(x) = 0, i \in \mathcal{E}; c_i(x) \geq 0, i \in \mathcal{I}\} \subseteq \mathbb{R}^n \tag{C.2}$$

as the *feasible set* of Problem C.1.

Definition C.2 (Active set ([16], p.40)). *Let $x \in \Omega$ be a point in the feasible set. The set*

$$\mathcal{A}(x) = \mathcal{E} \cup \{i \in \mathcal{I} : c_i(x) = 0\} \tag{C.3}$$

is called *active set* of Problem C.1 in point x .

The feasible set contains all points that satisfy the constraints $i \in \mathcal{E} \cup \mathcal{I}$, the active set contains all points which satisfy all constraints with equality.

Definition C.3 (Linear Independence Constraint Qualification - LICQ ([16], p.41)). *Let $x \in \Omega$ be a feasible point and $\mathcal{A}(x)$ the associated active set. The **Linear Independence Constraint Qualification (LICQ)** holds true in point x , if the set of gradients*

$$\{\nabla c_i(x) : i \in \mathcal{A}(x)\} \subseteq \mathbb{R}^n \tag{C.4}$$

is linearly independent.

One can also formulate this condition in terms of **tangent cone** and **linearized tangent cone** by setting both equal to each other as was done in ([12], p.42). However, with the definition

provided above, we can proof quite easily if the LICQ-condition is satisfied, as we just have to decide if a matrix of gradients as column-vectors has full rank.

Theorem C.4 (KKT: First Order Optimality Conditions ([16], p.42)). *Let x^* be a local solution of Problem C.1. Let the functions f and $c_i, i \in \mathcal{I} \cup \mathcal{E}$ be continuously differentiable. Furthermore let the LICQ condition hold in x^* . Then there exists a vector of Lagrange multipliers $\lambda^* \in \mathbb{R}^{|\mathcal{E}|+|\mathcal{I}|}$, such that the following conditions are satisfied in (x^*, λ^*)*

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) - \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla c_i(x^*) = 0 \quad (\text{C.5})$$

$$c_i(x^*) = 0 \quad \text{for all } i \in \mathcal{E} \quad (\text{C.6})$$

$$c_i(x^*) \geq 0 \quad \text{for all } i \in \mathcal{I} \quad (\text{C.7})$$

$$\lambda_i^* \geq 0 \quad \text{for all } i \in \mathcal{I} \quad (\text{C.8})$$

$$\lambda_i^* c_i(x^*) = 0 \quad \text{for all } i \in \mathcal{E} \cup \mathcal{I}. \quad (\text{C.9})$$

where $\mathcal{L}(x, \lambda) = f(x) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(x)$ is the Lagrange function.

Conditions C.5 - C.9 in Theorem C.4 are known as *Karush-Kuhn Tucker (KKT)* conditions. Here, condition C.5 is called the *dual admissibility* condition, conditions C.6 and C.7 are called *primal admissibility* conditions, condition C.8 is the *non-negativity* condition and condition C.9 is the *complementary* conditions. A point, which satisfies conditions C.5 - C.9 is denoted as *KKT-point*.

D. Appendix: Code

Code files used for simulation and numerical analysis can be found in this git-repository:

<https://github.com/K-Enin/CompressorCostOptimization>.

Acknowledgement

First, I would like to thank Prof. Dr. Simone Göttlich and Eike Fokken for enabling and supervising this Master Thesis and supporting me with theoretical and implementation questions.

I would also like to thank Felix Benning, Kiril Dik, Hashim Hafiz Imtiaz, Hélène Peter and Michelle Stichweh for proofreading this thesis and giving me feedback. Furthermore I am very thankful to Marcel Deutschel and Kiril Dik for the inspiring discussions on gas networks.

Finally, I would like to thank my parents and grandparents for their support during my final study years and the pandemic.

Glossary

Electrolysis	chemical process of passing an electric current through a substance in order to produce chemical changes in the substance
Energy conversion efficiency	ratio of the useful energy to the supplied energy. It is denoted by the letter η with property $\eta \in [0, 1]$. E.g. the energy conversion efficiency of a light bulb is around 5%, since a lot of power is being lost through heat
Inertia	property of a physical object with a tendency to stay in its existing state, unless that state is changed by an external force
Isentropic process	process, in which entropy remains unchanged
Isothermal process	process, in which temperature remains unchanged
Standard Volume (V_n)	unit of volume measure for gases under certain standard conditions according to DIN 1343 (1.01325 bar and 0 °Celsius)

Bibliography

- [1] Simone Göttlich, Andreas Potschka, and Claus Teuber. A partial outer convexification approach to control transmission lines. *Computational Optimization and Applications* (2019) 72:431–456, 28.11.2018. https://www.researchgate.net/publication/329250136_A_partial_outer_convexification_approach_to_control_transmission_lines.
- [2] Simone Göttlich, Andreas Potschka, and Ute Ziegler. Partial outer convexification for traffic light optimization in road networks. *SIAM Journal on Scientific Computing* 39(1):B53-B75, 01.02.2017. https://www.researchgate.net/publication/313254666_Partial_Outer_Convexification_for_Traffic_Light_Optimization_in_Road_Networks.
- [3] Peiran R. Liu and Adrian E. Raftery. Country-based rate of emissions reductions should increase by 80 % beyond nationally determined contributions to meet the 2 °c target. *Nature, Communications & Earth Environment*, 09.02.2021. <https://www.nature.com/articles/s43247-021-00097-8>.
- [4] German Aerospace Center (DLR). Concentrating solar power for the mediterranean region - final report. *German Aerospace Center (DLR), Institute of Technical Thermodynamics, Section Systems Analysis and Technology Assessment*, 16.04.2005. https://www.dlr.de/tt/Portaldata/41/Resources/dokumente/institut/system/publications/MED-CSP_complete_study.pdf.
- [5] Sebastian Schiebahn, Thomas Grube, Martin Robinius, Vanessa Tietze, Bhunesh Kumar, and Detlef Stolten. Power to gas: Technological overview, systems analysis and economic assessment for a case study in germany. *International Journal of Hydrogen Energy, Volumen 40*, 06.04.2015. <https://www.sciencedirect.com/science/article/abs/pii/S0360319915001913?via%3Dihub>.
- [6] IRENA. Hydrogen from renewable power: Technology outlook for the energy transition. *International Renewable Energy Agency*, September 2018. https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2018/Sep/IRENA_Hydrogen_from_renewable_power_2018.pdf.
- [7] Simone Göttlich. *Vorlesungsnotizen, Numerik partieller Differentialgleichungen, Hyperbolische Erhaltungsgleichungen*. Universität Mannheim, Department of Mathematics, Februar 2020.

- [8] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2004.
- [9] Randall J. LeVeque. *Numerical Methods for Conservation Laws, Second Edition*. Springer Basel AG, Lectures in Mathematics, ETH Zürich, 1992.
- [10] Sebastian Sager, Hans Georg Bock, and Moritz Diehl. The integer approximation error in mixed-integer optimal control. 2009. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.174.8287&rep=rep1&type=pdf>.
- [11] Sebastian Sager. Numerical methods for mixed-integer optimal control problems. *Der andere Verlag Tönning*, 2005, Lübeck.
- [12] Andreas Sommer. *Nonlinear Optimization, Lecture Notes*. Universität Mannheim, Department of Mathematics, Spring Semester 2019.
- [13] Bastian von Harrach. *Vorlesungsskript: Optimierung und Inverse Probleme*. Goethe-Universität Frankfurt am Main, Institut für Mathematik, Winter Semester 2020/2021. https://www.math.uni-frankfurt.de/~harrach/lehre/Optimierung_und_inverse_Probleme.pdf.
- [14] Peter Deuflhard. *Newton Methods for Nonlinear Problems, Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics, 2011.
- [15] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. *Mixed-integer nonlinear optimization*. Cambridge University Press, 2013. https://www.researchgate.net/publication/259432162_Mixed-integer_nonlinear_optimization.
- [16] Lars Schewe und Martin Schmidt. *Optimierung von Versorgungsnetzen: Mathematische Modellierung und Lösungstechniken*. Springer Spektrum, 2019.
- [17] Oliver Stein. *Grundzüge der Nichtlinearen Optimierung*, 2.Auflage. Springer Spektrum, 2021.
- [18] Andreas Wächter. Short tutorial: Getting started with ipopt in 90 minutes. *IBM T.J. Watson Research Center, Department of Business Analytics and Mathematical Sciences*, 2009. <https://projects.coin-or.org/CoinBinary/export/837/CoinAll/trunk/Installer/files/doc/Short%20tutorial%20Ipopt.pdf>.
- [19] Andreas Wächter and Lorenz Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program., Ser. A*, 2005. <http://cepac.cheme.cmu.edu/pasilectures/biegler/ipopt.pdf>.
- [20] World Meteorological Organization. Weather-related disasters increase over past 50 years, causing more damage but

- fewer deaths. <https://public.wmo.int/en/media/press-release/weather-related-disasters-increase-over-past-50-years-causing-more-damage-fewer>. Accessed: 2021-11-09.
- [21] Frank Graf, René Schoof, and Markus Zdrallek. *Power-to-Gas: Grundlagen-Konzepte-Lösungen*. Vulkan Verlag GmbH, Dezember 2020.
- [22] Michael Sternner, Mareike Jentsch, and Uwe Holzhammer. Energiewirtschaftliche und ökologische Bewertung eines Windgas Angebotes. *Fraunhofer Institut für Windenergie und Energiesystemtechnik (IWES)*, February 2021. https://www.greenpeace-energy.de/fileadmin/docs/sonstiges/Greenpeace_Energy_Gutachten_Windgas_Fraunhofer_Sternner.pdf.
- [23] Thomas Schmidt. *Wasserstofftechnik - Grundlagen, Systeme, Anwendung, Wirtschaft*. Carl Hanser Verlag München, 2020.
- [24] Electrical4u. <https://www.electrical4u.com/admittance/>.
- [25] John J.Grainger and William D. Stevenson. *Power System Analysis*. McGraw-Hill Series in Electrical and Computer Engineering, 1994.
- [26] Armin Fügenschuh, Björn Geißler, Ralf Gollmer, Antonio Morsi, Marc Pfetsch, Jessica Rövekamp, Martin Schmidt, Klaus Spreckelsen, and Marc Steinbach. *Evaluating Gas Network Capacities* (pp.17-43). Society for Industrial and Applied Mathematics, SIAM, March 2015. <https://pubs.siam.org/doi/book/10.1137/1.9781611973693?mobileUi=0>.
- [27] Michael Herty, Joachim Mohring, and Veronika Sachers. A new model for gas flow in pipe networks. *Mathematical Methods in the Applied Sciences*, Wiley InterScience, 17.07.2009.
- [28] Martin Gugat and Stefan Ulbrich. The isothermal euler equations for ideal gas with source term: Product solutions, flow reversal and no blow up. *Journal of Mathematical Analysis and Applications*, 01.10.2016. <https://www.sciencedirect.com/science/article/abs/pii/S0022247X17304274>.
- [29] Joel A. E. Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 2018.
- [30] HSL for IPOPT. Science and technology facilities council. <https://www.hsl.rl.ac.uk/ipopt/>. accessed July 2021.
- [31] Open-source solver bonmin. <https://projects.coin-or.org/Bonmin>. accessed 25.10.2021.

- [32] Eike Fokken, Simone Göttlich, and Oliver Kolb. Optimal control of compressor stations in a coupled gas-to-power network. *Advances in Energy System Optimization*, Springer International Publishing, January 2020. https://www.researchgate.net/publication/337742720_Optimal_Control_of_Compressor_Stations_in_a_Coupled_Gas-to-Power_Network.
- [33] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis, Third Edition*. Springer Science+Business Media, 2002.
- [34] Gerda Gahleitner. Hydrogen from renewable electricity: An international review of power-to-gas pilot plants for stationary applications. *International Journal of Hydrogen Energy*, 31.12.2012. https://www.researchgate.net/publication/257175385_Hydrogen_from_renewable_electricity_An_international_review_of_power-to-gas_pilot_plants_for_stationary_applications.
- [35] Yue Qui, Sara Grundel, Martin Stoll, and Peter Benner. Efficient numerical methods for gas network modeling and simulation. *Networks and heterogeneous media* 15(4), January 2020. https://www.researchgate.net/publication/344061913_Efficient_numerical_methods_for_gas_network_modeling_and_simulation.
- [36] Eike Fokken, Simone Göttlich, and Oliver Kolb. Modeling and simulation of gas networks coupled to power grids. *Springer Nature*, 23.11.2019. <https://link.springer.com/article/10.1007/s10665-019-10026-6>.
- [37] Oliver Kolb. Dissertation on simulation and optimization of gas and water supply networks. *TU Darmstadt*, 2011.
- [38] Martin Zapf. *Stromspeicher und Power-to-Gas im deutschen Energiesystem: Rahmenbedingungen, Bedarf und Einsatzmöglichkeiten*. Springer Vieweg, 03.02.2017.