Solving Least-Squares Collocated Differential Algebraic Equations by Successive Abs-Linear Minimization

A Case Study on Gas Network Simulation

Timo Kreimeier* Henning Sauter† Tom Streubel‡ Caren Tischendorf§
Andrea Walther¶

January 19, 2022

Abstract

This paper studies the numerical simulation of gas networks with regulating elements using differential algebraic equations (DAEs) in combination with least-squares collocation. In contrast to classical collocation methods, more collocation points than degrees of freedom for the collocation polynomials are used. Recently, it has been shown that such a least-squares collocation has a regularizing effect for DAEs, in particular for DAEs with higher index. In each time step of the numerical integration, one has to solve a system of nonlinear equations that is nonsmooth due to the regulating elements in the gas networks. We consider four solvers one of which explicitly exploits the inherent nonsmooth nature. Numerical results are given for three different test cases with increasing complexity illustrating the feasibility of the proposed approach to approximate a solution of the DAE and the advantageous performance of the nonsmooth solver that is based on the concept of abs-linearization.

Keywords – Numerical Integration, Least-Squares Collocation Method, Simulation of Gas Transportation Networks, Target or Set Point Values, Controlling Regulators and Compressors, Abs-smooth Algorithmic Differentiation, SALMIN, Nonsmooth Optimization

2010 Mathematics Subject Classification – 65L80, 65H10, 65L05, 65M22, 65F20, 65L20, 65Z05

1 Introduction

Given the ongoing and substantial transformation of the energy market, the simulation of energy systems becomes more and more important. This also includes the gas industry in its entirety. In the future one may either utilize gas network infrastructures for the dynamical energy storage to smoothen the highly volatile generation of renewable energy sources [Fed21] or one may want to shift from mostly methane to pure hydrogen transport [Eur20] or both. Either way this will increase the complexity and dynamics for controlling gas networks. More variability of demand and supply has to be considered given the different properties of hydrogen compared to natural gas [Hop+20].

Here, we study industry relevant modellings for various types of gas network components, also called elements, from recent research and combine them into a comprehensive modelling framework for the description of gas networks using differential algebraic equations (DAEs). We include advanced features such as transient target value control, also known as set-point value control, for actively regulating network elements, i.e., regulators and compressors. The coverage of the additional capabilities leads to nonsmoothness in the overall mathematical model rendering it to

*Humboldt-Universität zu Berlin, Germany, orcid.org/0000-0002-6268-6402
†Humboldt-Universität zu Berlin, Germany, orcid.org/0000-0003-4603-3943, sauter@math.hu-berlin.de
‡Humboldt-Universität zu Berlin, Germany, orcid.org/0000-0003-4917-7977, streubel@math.hu-berlin.de
§Humboldt-Universität zu Berlin, Germany, orcid.org/0000-0002-8421-7199
¶Humboldt-Universität zu Berlin, Germany, orcid.org/0000-0002-3516-4641

become piecewise differentiable. The resulting gas network DAEs may have a varying number of dynamic components. Furthermore, such DAEs may change their index when valves switch from open to close and vice versa. Therefore, we study the least-squares collocation method for such network DAEs. This approach shows a surprising well behavior for the numerical solution of higher index DAEs [Han+17; HM21c]. Convergence theory for this class of numerical integration methods and for numerous types of DAEs has been established [HM21b; HM21a; HMT19], but is still under ongoing research and investigation.

We aim to contribute numerical verification from computational experiments via the following setup: Numerical integrators for DAEs typically generate a sequence of root problems, which in our case are both overdetermined and nonsmooth. These generated subproblems correspond to a minimization problem of a scalar-valued objective function, once some norm has been applied onto it. Such a nonsmooth minimization is still a challenging problem. Therefore, sometimes the nonsmoothness is simply ignored when solving such problems, see, e.g., [LO13]. Other approaches are based on bundle methods that employ numerous parameters, see, e.g., [Bag+20] for a recent overview. In contrast to that, we embed a dedicated algorithm, called SALMIN, as proposed in [FWG19]. SALMIN exploits explicitly the structure provided by the nonsmoothness of so-called abs-smooth objectives defined below in more detail. The nonsmooth optimization problems stemming from the approximate solution of DAEs lead exactly to abs-smooth target functions and hence we use SALMIN as an alternative to other solvers. We compare SALMIN with the least-squares solver of scipy. Either solver is employed to solve the nonsmooth subproblems from a sequence of overdetermined root problems generated by the least-squares collocation method.

The paper is structured as follows: In Section 2 we introduce and cite a version of isothermal Euler equations suited for the modelling of gas flow in pipes and pipeline segments. In the next step, Section 3 presents an elaborated mathematical modelling framework for the formulation of gas networks as nonsmooth differential algebraic equations, capable to describe entire national gas grids. Towards that goal and still within the same section, we gather and describe industry-relevant modellings for valves, control valves and compressors, but also reenact a topology-adaptive discretization approach for pipes modelled with the aforementioned variant of Euler equations. Subsequently, Section 4 introduces and substantiates the least-squares collocation approach for the numerical integration of differential algebraic equations arising from gas network modelling. In Section 5, the SALMIN approach already mentioned above is described briefly. It will be used to solve the nonsmooth optimization problems which are sequentially generated by the least-squares collocated integration. Numerical results comparing the performance of SALMIN with the least-squares solver of scipy are contained in Section 6. For this purpose, we apply the solvers within the least-squares collocated integration to approximate the gas flow for three distinct gas network instances. The three gas network instances range from two consecutive pipes, over a single control valve up to a gas network of 70 nodes with a complex gas network station the hearth of the network. Hence, these examples cover elements of all types that were introduced in Section 3. Finally, a summary and an outlook are given in Section 7.

2 Pipe Physics

The physics of gas flowing through a pipe can be described by the Euler equations. In [Dom+21] a variety of simplified models have been derived, ranging from the original Euler equations via temperature-independent *isothermal* models up to purely algebraic ones. Here, we focus on the ISO2 system of equations, reading

$$\partial_t\big(\varphi(p)\big) + \kappa \partial_x q(t) = 0, \tag{1a}$$

$$\partial_t q + A\partial_x p + \frac{\lambda\kappa}{2D}\frac{q|q|}{\varphi(p)} + \frac{g}{\kappa}\sin(\theta)\varphi(p) = 0, \tag{1b}$$

with the function

$$\varphi(p) = \frac{p}{z(p)}.$$

On this modelling level, all terms are included that significantly contribute to the real gas network flow. The intrinsic quantities are the pressure $p \equiv p(x,t)$ and the flow $q \equiv q(x,t)$, each being a function in two arguments. The first argument is a one-dimensional space variable $x \in \Omega = [0, L] \subset \mathbb{R}$ parameterizing the longitudinal axis of the pipe. We refer to $x = 0 \in \Omega$ as the left end and to $x = L \in \Omega$ as the right end of the pipe, where $L > 0$ denotes the length of the pipe. Note that this just indicates the topological orientation of the pipe and does not necessarily coincide with the flow direction of the gas, which can go either way. The second argument of both pressure and flow is the time $t \in [t_0, T]$. The other parameters appearing in (1) are described in Table 1.

Parameter	Description
D	Pipe diameter
$A = \frac{1}{4}\pi D^2$	Pipe cross-sectional area
θ	Pipe slope angle (compared to the ground)
g	Gravitational acceleration
$\kappa = \frac{R_s T}{A}$	Fraction of the specific gas constant, the temperature and the pipe cross-sectional area
λ	Friction factor (see subsection 2.1)
$z(p)$	Real gas factor depending on the pessure (see subsection 2.2)

Table 1: Parameters appearing in the ISO2 system of equations (1)

2.1 Friction Factor Model

The friction factor is a crucial part of the empirical Darcy-Weißbach model for friction of transported fluid media within a pipe [Bro02]. It is crucial as it allows a granular specialization of that model as a trade-off between relatively high accuracy versus faster computable approximations. Correspondingly, literature offers quite a variety of models, each with different emphasis. Decent summaries beyond industrial norm defining documentations can be found, e.g., in [Dom+21] as well as in [Ben+19]. For our purpose we will use the *Nikuradse* formula [Nik50] which is given by

$$\lambda = \left[-2\log_{10}\left(r/(3.71D)\right)\right]^{-2} \tag{2}$$

and which is an explicit simplification of the otherwise implicit *Colebrook-White* [CW37; Bro02] formula. With r we denote the roughness of the pipe wall surface that should be provided in [m].

2.2 Real Gas Factor Model

The real gas factor adjusts the behaviour of an actual composition of gas from the expected behaviour of idealized gas, which is an abstract concept in chemistry. This factor is a quantity that can actually only be measured in experiments. However, it is assumed that it has an equivalent power series representation referred to as *Virial expansion* [Onn91]. Partial sums, i.e., polynomials, can be estimated by best fitting approaches and are typically accepted and used as model. Here, we will make use of the AGA formula [SL21; Dom+21] from the American Gas Association which is linear in p and considered accurate up to pressures $p \leq 70$[bar]. Other potential options can be taken from further literature, e.g., [Pap68; Sal02; Dom+21; Ben+19].

3 Gas Network Modelling

We follow the approach in [Dom+21] and model a gas network as an oriented graph $G = (\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. Each edge $e \in \mathcal{E}$ has a fixed orientation and so we may denote the two nodes belonging to an edge as left node $v_l \in \mathcal{V}$ and right node $v_r \in \mathcal{V}$, with the convention that the edge is directed from left to right. Note that the orientation of the edge does not imply gas flowing only in that direction. In particular, positive flow values correspond to gas flowing from left to right whereas negative flow values correspond to gas flowing from right to left.

We distinguish nodes with respect to their modelling. In $\mathcal{V}_{\text{pset}}$ we collect all nodes with a fixed pressure. Fixed pressure means the existence of a time-dependent function chosen in advance, yielding the respective pressure value at each point in time. We call them *pressure nodes*. All other nodes are collected in $\mathcal{V}_{\text{qset}}$. Note that $\mathcal{V} = \mathcal{V}_{\text{pset}} \dot{\cup} \mathcal{V}_{\text{qset}}$ holds.

Regarding the edge elements, we focus on networks consisting of pipes, valves and regulating elements (regulators and compressors). Thus, we impose the partition $\mathcal{E} = \mathcal{E}_{\text{pip}} \dot{\cup} \mathcal{E}_{\text{val}} \dot{\cup} \mathcal{E}_{\text{reg}}$.

3.1 Nodal equations

For pressure nodes, i.e., nodes $u \in \mathcal{V}_{\mathrm{pset}}$, the pressure is given by the pressure boundary condition

$$\begin{pmatrix} \vdots \\ p_u^{\mathrm{set}}(t) \\ \vdots \end{pmatrix}_{u \in \mathcal{V}_{\mathrm{pset}}}.$$

For all other nodes, i.e., $u \in \mathcal{V}_{\mathrm{qset}}$, a Kirchhoff-type flow balance equation

$$A_{\mathrm{pip,l}} q_{\mathrm{pip,l}} + A_{\mathrm{pip,r}} q_{\mathrm{pip,r}} + A_{\mathrm{val}} q_{\mathrm{val}} + A_{\mathrm{reg}} q_{\mathrm{reg}} = q^{\mathrm{set}}(t) \tag{3}$$

is given. For junction nodes, there is neither an incoming nor an outgoing flow, i.e., $q^{\mathrm{set}} = 0$. For the demand nodes, we have an outgoing flow $q^{\mathrm{set}} \geq 0$. For source nodes with a fixed inflow, we have $q^{\mathrm{set}} \leq 0$. The flows $q_{\mathrm{pip,l}}$, $q_{\mathrm{pip,r}}$, q_{val} and q_{reg} represent the flows at the left end of pipes, at the right end of pipes, through valves and through regulating elements, respectively. The matrices used in (3) are (constant) incidence matrices given by

$$A_{\mathrm{pip,l}} \coloneqq \left(a_{ij}^{\mathrm{pip,l}}\right)_{\substack{i=1,\dots,|\mathcal{V}_{\mathrm{qset}}| \\ j=1,\dots,|\mathcal{E}_{\mathrm{pip}}|}}, \qquad a_{ij}^{\mathrm{pip,l}} \coloneqq \begin{cases} -1 & \text{if node } i \text{ is the left node of pipe } j, \\ 0 & \text{else,} \end{cases}$$

$$A_{\mathrm{pip,r}} \coloneqq \left(a_{ij}^{\mathrm{pip,r}}\right)_{\substack{i=1,\dots,|\mathcal{V}_{\mathrm{qset}}| \\ j=1,\dots,|\mathcal{E}_{\mathrm{pip}}|}}, \qquad a_{ij}^{\mathrm{pip,r}} \coloneqq \begin{cases} 1 & \text{if node } i \text{ is the right node of pipe } j, \\ 0 & \text{else,} \end{cases}$$

$$A_{\mathrm{val}} \coloneqq \left(a_{ij}^{\mathrm{val}}\right)_{\substack{i=1,\dots,|\mathcal{V}_{\mathrm{qset}}| \\ j=1,\dots,|\mathcal{E}_{\mathrm{val}}|}}, \qquad a_{ij}^{\mathrm{val}} \coloneqq \begin{cases} -1 & \text{if node } i \text{ is the left node of valve } j, \\ 1 & \text{if node } i \text{ is the right node of valve } j, \\ 0 & \text{else,} \end{cases}$$

$$A_{\mathrm{reg}} \coloneqq \left(a_{ij}^{\mathrm{reg}}\right)_{\substack{i=1,\dots,|\mathcal{V}_{\mathrm{qset}}| \\ j=1,\dots,|\mathcal{E}_{\mathrm{reg}}|}}, \qquad a_{ij}^{\mathrm{reg}} \coloneqq \begin{cases} -1 & \text{if node } i \text{ is the left node of regulating element } j, \\ 1 & \text{if node } i \text{ is the right node of regulating element } j, \\ 0 & \text{else,} \end{cases}$$

3.2 Edge element equations

3.2.1 Pipes

Since we are looking for numerical solutions the equations (1) need to be discretized. We follow the approach in [Ben+19] of discretizing first in space to obtain a differential algebraic equation system and make use of a topology-adapted left-right discretization scheme ([Huc18], [Ben+19]) resulting in an index-1 DAE. Then, for each pipe $e \in \mathcal{E}_{\mathrm{pip}}$, the spatially discretized equations have the form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\varphi(p_{\mathrm{r},e}) + \frac{\kappa_e}{L_e}(q_{\mathrm{r},e} - q_{\mathrm{l},e}) &= 0, \\ \frac{\mathrm{d}}{\mathrm{d}t} q_{\mathrm{l},e} + \frac{A_e}{L_e}(p_{\mathrm{r},e} - p_{\mathrm{l},e}) + \frac{\lambda_e \kappa_e}{2D_e} \frac{q_{\mathrm{l},e}|q_{\mathrm{l},e}|}{\varphi(p_{\mathrm{r},e})} + \frac{g}{\kappa_e}\sin(\theta_e)\varphi(p_{\mathrm{r},e}) &= 0, \end{aligned}$$

where L_e is the length of pipe e, $\varphi(p) = \frac{p}{z(p)}$ and

$$p_{\mathrm{l},e}(t) \coloneqq p_e(0,t), \qquad q_{\mathrm{l},e}(t) \coloneqq q_e(0,t), \qquad p_{\mathrm{r},e}(t) \coloneqq p_e(L_e,t), \qquad q_{\mathrm{r},e}(t) \coloneqq q_e(L_e,t).$$

Collecting all left pressures $p_{\mathrm{l},e}$ of pipes in the vector $p_{\mathrm{pip,l}}$, all right pressures $p_{\mathrm{r},e}$ of pipes in the vector $p_{\mathrm{pip,r}}$, all left flows $q_{\mathrm{l},e}$ of pipes in the vector $q_{\mathrm{pip,l}}$ and all right flows $q_{\mathrm{r},e}$ of pipes in the vector $q_{\mathrm{pip,r}}$, we obtain the pipe equation system

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\bar{\varphi}(p_{\mathrm{pip,r}}) + D_q(q_{\mathrm{pip,r}} - q_{\mathrm{pip,l}}) &= 0, \\ \frac{\mathrm{d}}{\mathrm{d}t} q_{\mathrm{pip,l}} + D_p(p_{\mathrm{pip,r}} - p_{\mathrm{pip,l}}) + f_{\mathrm{fric}}(p_{\mathrm{pip,r}}, q_{\mathrm{pip,l}}) + f_{\mathrm{grav}}(p_{\mathrm{pip,r}}) &= 0 \end{aligned}$$

with $D_q \coloneqq \operatorname{diag}\{..., \frac{\kappa_e}{L_e}, ...\}_{e\in\mathcal{E}_{\text{pip}}}$, $D_p \coloneqq \operatorname{diag}\{..., \frac{A_e}{L_e}, ...\}_{e\in\mathcal{E}_{\text{pip}}}$,

$$\bar{\varphi}(p) \coloneqq \begin{pmatrix} \vdots \\ \varphi(p_e) \\ \vdots \end{pmatrix}_{e\in\mathcal{E}_{\text{pip}}}, \qquad f_{\text{fric}}(p,q) \coloneqq \begin{pmatrix} \vdots \\ \frac{\lambda_e\kappa_e}{2D_e}\frac{q_e|q_e|}{\varphi_e(p_e)} \\ \vdots \end{pmatrix}_{e\in\mathcal{E}_{\text{pip}}}, \qquad f_{\text{grav}}(p) \coloneqq \begin{pmatrix} \vdots \\ \frac{g}{\kappa_e}\sin(\theta_e)\varphi_e(p_e) \\ \vdots \end{pmatrix}_{e\in\mathcal{E}_{\text{pip}}}.$$

Collecting all pressures of nodes $u \in \mathcal{V}_{\text{qset}}$ in the vector p, we have

$$p_{\text{pip,l}} = -A_{\text{pip,l}}^\top p - A_{\text{pip,s}}^\top p^{\text{set}}, \quad p_{\text{pip,r}} = A_{\text{pip,r}}^\top p.$$

Here, $A_{\text{pip,s}}$ is the incidence matrix describing the (left) connections of pipes to pressure nodes, i.e., the definition coincides with the definition of $A_{\text{pip,l}}$ but the rows represent the nodes $u \in \mathcal{V}_{\text{pset}}$ instead of the nodes $u \in \mathcal{V}_{\text{qset}}$.

Note that we used here the assumption that pressure nodes are not located at right ends of pipes. It is always possible to give all pipes an orientation satisfying this assumption unless there is one pipe that connects two pressure nodes. The latter case is not really useful in practice. Furthermore, such a pipe could be modelled as a connection of two pipes of half length having the additional connecting node as right node.

Consequently, we can summarize the pipe equations as

$$A_{\text{pip,r}}^\top \frac{\mathrm{d}}{\mathrm{d}t}\phi(p) + D_q(q_{\text{pip,r}} - q_{\text{pip,l}}) = 0, \tag{4a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}q_{\text{pip,l}} + D_p(A_{\text{pip,r}}^\top p + A_{\text{pip,l}}^\top p) + f_{\text{pip}}(p, q_{\text{pip,l}}, t) = 0 \tag{4b}$$

with

$$f_{\text{pip}}(p,q,t) \coloneqq f_{\text{fric}}(A_{\text{pip,r}}^\top p, q) + f_{\text{grav}}(A_{\text{pip,r}}^\top p) + D_p A_{\text{pip,s}}^\top p^{\text{set}}(t) \text{ and } \phi(p) \coloneqq \begin{pmatrix} \vdots \\ \varphi(p_u) \\ \vdots \end{pmatrix}_{u\in\mathcal{V}_{\text{qset}}}.$$

3.2.2 Valves

Valves are represented as edges in the overall graph. The mathematically idealized valve knows two definitive states. There is the *ideal-open* state which allows a bidirectional, unresisted and drag-free flow of gas, always balancing the adjacent node pressures. The other or *ideal-closed* state does not allow any exchange of gas or equivalently enforces $q = 0$. The main purpose from a modelling point of view is the manipulation of the underlying topology at simulation runtime. In between, there may or may not be a short-lived transition phase. Hence, we require a piecewise linear, continuous profile function $\Theta : [t_0, T] \to [0, 1]$ to determine the state. Thus, the idealized valve on edge $e \in \mathcal{E}_{\text{val}}$ may be modelled via

$$0 = \begin{cases} p_{r,e} - p_{l,e} & \text{if } \Theta_e(t) = 1, \\ q_e & \text{else.} \end{cases}$$

Another, but continuous version derived from the one above is provided by

$$0 = \mu_e\,\dot{q}_e + \Theta_e(t)\cdot(p_{r,e} - p_{l,e}) + \nu_e(1 - \Theta_e(t))q_e,$$

with $\mu_e = 0$ to achieve an algebraic constraint or $\mu_e > 0$ to achieve a differential constraint. The factors $\mu_e > 0$ and $\nu_e > 0$ can be used for suitable scaling. Collecting all left pressures $p_{l,e}$ of valves in the vector $p_{\text{val,l}}$, all right pressures $p_{r,e}$ of valves in the vector $p_{\text{val,r}}$ and all flows q_e through valves in the vector q_{val}, we obtain the valve equation system

$$D_{\text{val}}\dot{q}_{\text{val}} + D_\Theta(t)\cdot(p_{\text{val,r}} - p_{\text{val,l}}) + D_\nu(I - D_\Theta(t))q_{\text{val}} = 0$$

with $D_{\text{val}} \coloneqq \operatorname{diag}\{..., \mu_e, ...\}_{e\in\mathcal{E}_{\text{val}}}$, $D_\nu \coloneqq \operatorname{diag}\{..., \nu_e, ...\}_{e\in\mathcal{E}_{\text{val}}}$ and $D_\Theta(t) \coloneqq \operatorname{diag}\{..., \Theta_e(t), ...\}_{e\in\mathcal{E}_{\text{val}}}$. Furthermore, we know that

$$p_{\text{val,r}} - p_{\text{val,l}} = A_{\text{val}}^\top p + A_{\text{val,s}}^\top p^{\text{set}}(t).$$

Here, $A_{\mathrm{val,s}}$ is again an incidence matrix. It is defined as A_{val} but the rows represent the nodes $u \in \mathcal{V}_{\mathrm{pset}}$ instead of the nodes $u \in \mathcal{V}_{\mathrm{qset}}$. Hence, we may summarize the valve equations as

$$D_{\mathrm{val}}\dot{q}_{\mathrm{val}} + f_{\mathrm{val}}(p, q_{\mathrm{val}}, t) = 0 \tag{5}$$

with

$$f_{\mathrm{val}}(p, q_{\mathrm{val}}, t) \coloneqq D_{\Theta}(t)A_{\mathrm{val}}^{\top}p + D_{\nu}\big(I - D_{\Theta}(t)\big)q_{\mathrm{val}} + A_{\mathrm{val,s}}^{\top}p^{\mathrm{set}}(t).$$

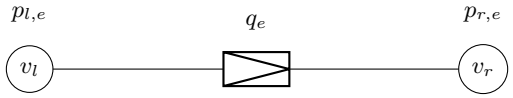
3.2.3 Actively Regulating Elements – Regulators & Compressors

In order to actively control the gas transportation process, so called *regulators* (also referred to as *control valves*) and *compressors* are in use for the up- and down-regulation of gas flow q. Regulators one-way down regulate gas by creating resistance, whereas the up-regulation is accomplished by *compressors*. The amount or intensity of regulation is not determined by hand. Instead, so-called target or set point values are used to define desired states. Regulators and compressors adjust themselves to get as close as possible. There are many different potential kinds of target values, we will present five. Target values are represented by time-dependent profile functions, which we assume to be piecewise linear in the sense of [Sch12]. They are part of the overall scenario description and are hence supposed to be provided upfront. The first four target values define two desired ranges for the left p_ℓ right pressure p_r. Those are $\underline{p_l} : [t_0, T] \to \mathbb{R}_{\geq 0}$, i.e., the desired minimal target pressure on the left and $\overline{p_l} : [t_0, T] \to \mathbb{R}_{\geq 0}$, the desired maximal target pressure on the left as well as $\underline{p_r}, \overline{p_r} : [t_0, T] \to \mathbb{R}_{\geq 0}$ the minimal and maximal target pressures on the right. The fifth and final target value is $q^{\mathrm{set}} : [t_0, T] \to \mathbb{R}$ defining a desired level of gas flow. These five target values are by no means bounds, since maintaining all five target values simultaneously may sometimes result in conflicts. Instead, there is a prioritizing hierarchy among target values that is condensed and summarized in Table 2, which itself is merged and derived from Table 1 in [HPS21] as well as Table 5.2 in [Str21].

affected target value	priority	violation scenario	response to violation
lower left pressure $\underline{p_l}$	2	$\underline{p_l} > p_\ell$	closing
upper right pressure $\overline{p_r}$	2	$\overline{p_r} < p_r$	closing
upper left pressure $\overline{p_l}$	1	$\overline{p_l} < p_\ell$	opening
lower right pressure $\underline{p_r}$	1	$\underline{p_r} > p_r$	opening
flow set q^{set}	0	$q^{\mathrm{set}} < q$	closing
flow set q^{set}	0	$q^{\mathrm{set}} > q$	opening

Table 2: Table of all target value violations, their priority of being treated and their responses to violation

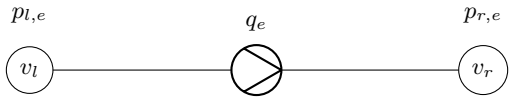
A single *regulator* $e \in \mathcal{E}_{\mathrm{reg}}$ is represented by the following circuit symbol



and a target value compatible model as suggested by [Str21] and [HPS21] reads

$$\left.\begin{aligned} 0 = \mu\dot{q}_e - \max(-\nu q_e, \min(p_{l,e} - \max(\underline{p_l}, p_{r,e}), \min(\overline{p_r}, p_{l,e}) - p_{r,e}, \\ \max(\nu(q^{\mathrm{set}} - q_e), p_{l,e} - \overline{p_l}, \underline{p_r} - p_{r,e}))), \end{aligned}\right\} \tag{6}$$

with $\mu = 0$ to achieve an algebraic or $\mu > 0$ for a differential constraint. Again, the factors $\mu > 0$ and $\nu > 0$ can be used for suitable scaling. Likewise, an idealized *compressor* $e \in \mathcal{E}_{\mathrm{reg}}$ is represented by the following circuit symbol



and a target value compatible model is suggested by [Str21] as

$$\left.\begin{aligned} 0 = \mu\dot{q}_e - \max(-\nu q_e, p_{l,e} - p_{r,e}, \min(p_{l,e} - \underline{p_l}, \overline{p_r} - p_{r,e}, \\ \max(\nu(q^{\mathrm{set}} - q_e), p_{l,e} - \overline{p_l}, \underline{p_r} - p_{r,e}))), \end{aligned}\right\} \tag{7}$$

with $\mu = 0$ to achieve an algebraic or $\mu > 0$ for a scalable differential constraint.

Consequently, the regulating element on edge e can be modelled as

$$\mu_e \dot{q}_e - f_{\mathrm{reg,e}}(p_{l,e}, p_{r,e}, q_e) = 0$$

with the continuous and piecewise differentiable function

$$f_{\mathrm{reg,e}}(p_l, p_r, q) \coloneqq \max\Big\{-\nu_e q,\ \min\big\{p_l - \max\{\underline{p_l}, p_r\},\ \min\{\overline{p_r}, p_l\} - p_r, \max\{\nu_e(q^{\mathrm{set}} - q), p_l - \overline{p_l}, \underline{p_r} - p_r\}\big\}\Big\}$$

in case of a regulator $e \in \mathcal{E}_{\mathrm{reg}}$ and

$$f_{\mathrm{reg,e}}(p_l, p_r, q) \coloneqq \max\Big\{-\nu_e q, p_l - p_r, \min\big\{p_l - \underline{p_l},\ \overline{p_r} - p_r, \max\{\nu_e(q^{\mathrm{set}} - q), p_l - \overline{p_l}, \underline{p_r} - p_r\}\big\}\Big\}$$

in case of an idealized compressor $e \in \mathcal{E}_{\mathrm{reg}}$.

Collecting all left pressures $p_{l,e}$ of regulating elements in the vector $p_{\mathrm{reg,l}}$, all right pressures $p_{r,e}$ of regulating elements in the vector $p_{\mathrm{reg,r}}$ and all flows q_e of regulating elements in the vector q_{reg}, we obtain the regulating element equation system

$$D_{\mathrm{reg}} \dot{q}_{\mathrm{reg}} - f_r(p_{\mathrm{reg,l}}, p_{\mathrm{reg,r}}, q_{\mathrm{reg}}) = 0$$

with $D_{\mathrm{reg}} \coloneqq \mathrm{diag}\{..., \mu_e, ...\}_{e \in \mathcal{E}_{\mathrm{reg}}}$,

$$f_r(p_{\mathrm{reg,l}}, p_{\mathrm{reg,r}}, q_{\mathrm{reg}}) \coloneqq \begin{pmatrix} \vdots \\ f_{\mathrm{reg,e}}(p_{l,e}, p_{r,e}, q_e) \\ \vdots \end{pmatrix}_{e \in \mathcal{E}_{\mathrm{reg}}}.$$

The left and right pressures $p_{\mathrm{reg,l}}$ and $p_{\mathrm{reg,r}}$ can again be expressed by the nodal pressures p and p^{set} as

$$p_{\mathrm{reg,l}} = -A_{\mathrm{reg,l}}^\top p - A_{\mathrm{reg,ls}}^\top p^{\mathrm{set}}, \quad p_{\mathrm{reg,r}} = A_{\mathrm{reg,r}}^\top p + A_{\mathrm{reg,rs}}^\top p^{\mathrm{set}},$$

where $A_{\mathrm{reg,l}}$ and $A_{\mathrm{reg,r}}$ are incidence matrices defined as $A_{\mathrm{pip,l}}$ and $A_{\mathrm{pip,r}}$, but the columns represent the regulating elements instead of pipes. Note that $A_{\mathrm{reg}} = A_{\mathrm{reg,r}} + A_{\mathrm{reg,l}}$. Analogously, $A_{\mathrm{reg,ls}}$ and $A_{\mathrm{reg,rs}}$ are incidence matrices. They are defined as $A_{\mathrm{reg,l}}$ and $A_{\mathrm{reg,r}}$, but the rows represent the nodes $u \in \mathcal{V}_{\mathrm{pset}}$ instead of the nodes $u \in \mathcal{V}_{\mathrm{qset}}$. Hence, we may summarize the regulating element equations as

$$D_{\mathrm{reg}} \dot{q}_{\mathrm{reg}} - f_{\mathrm{reg}}(p, q_{\mathrm{reg}}, t) = 0 \tag{8}$$

with

$$f_{\mathrm{reg}}(p, q_{\mathrm{reg}}, t) \coloneqq f_r(-A_{\mathrm{reg,l}}^\top p - A_{\mathrm{reg,ls}}^\top p^{\mathrm{set}}(t), A_{\mathrm{reg,r}}^\top p + A_{\mathrm{reg,rs}}^\top p^{\mathrm{set}}(t), q_{\mathrm{reg}}).$$

3.3 DAE network model

Collecting the edge element equations (4), (5) and (8) and the nodal equations (3), we obtain the gas network DAE

$$A_{\mathrm{pip,r}}^\top \frac{\mathrm{d}}{\mathrm{d}t}\phi(p) + D_q(q_{\mathrm{pip,r}} - q_{\mathrm{pip,l}}) = 0, \tag{9a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} q_{\mathrm{pip,l}} + D_p(A_{\mathrm{pip,r}}^\top + A_{\mathrm{pip,l}}^\top)p + f_{\mathrm{pip}}(p, q_{\mathrm{pip,l}}, t) = 0, \tag{9b}$$

$$D_{\mathrm{val}} \frac{\mathrm{d}}{\mathrm{d}t} q_{\mathrm{val}} + f_{\mathrm{val}}(p, q_{\mathrm{val}}, t) = 0, \tag{9c}$$

$$D_{\mathrm{reg}} \frac{\mathrm{d}}{\mathrm{d}t} q_{\mathrm{reg}} - f_{\mathrm{reg}}(p, q_{\mathrm{reg}}, t) = 0, \tag{9d}$$

$$A_{\mathrm{pip,l}} q_{\mathrm{pip,l}} + A_{\mathrm{pip,r}} q_{\mathrm{pip,r}} + A_{\mathrm{val}} q_{\mathrm{val}} + A_{\mathrm{reg}} q_{\mathrm{reg}} = q^{\mathrm{set}}(t). \tag{9e}$$

Additionally, we have technical bounds of the form

$$f_{\mathrm{pb}}(p) = 0, \qquad f_{\mathrm{qb}}(q_{\mathrm{pip,l}}, q_{\mathrm{pip,r}}, q_{\mathrm{val}}, q_{\mathrm{reg}}) = 0 \tag{9f}$$

with continuous, piecewise smooth functions f_{pb} and f_{qb}.

The next section describes a numerical solution approach for such network DAEs.

4 Least-squares Collocation Method for Differential Algebraic Equations

The gas network DAE (9) can be written as

$$A\frac{\mathrm{d}}{\mathrm{d}t}\psi(x(t)) + b(x(t),t) = 0 \tag{10}$$

with $x = (p, q_{\mathrm{pip,l}}, q_{\mathrm{pip,r}}, q_{\mathrm{val}}, q_{\mathrm{reg}})$,

$$A = \begin{pmatrix} A_{\mathrm{pip,r}}^\top & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & D_{\mathrm{val}} & 0 \\ 0 & 0 & 0 & D_{\mathrm{reg}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad \psi(x) := \begin{pmatrix} \phi(p) \\ q_{\mathrm{pip,l}} \\ q_{\mathrm{val}} \\ q_{\mathrm{reg}} \end{pmatrix}$$

and

$$b(x,t) := \begin{pmatrix} D_q(q_{\mathrm{pip,r}} - q_{\mathrm{pip,l}}) \\ D_p(A_{\mathrm{pip,r}}^\top + A_{\mathrm{pip,l}}^\top)p + f_{\mathrm{pip}}(p, q_{\mathrm{pip,l}}, t) \\ f_{\mathrm{val}}(p, q_{\mathrm{val}}, t) \\ -f_{\mathrm{reg}}(p, q_{\mathrm{reg}}, t) \\ A_{\mathrm{pip,l}}q_{\mathrm{pip,l}} + A_{\mathrm{pip,r}}q_{\mathrm{pip,r}} + A_{\mathrm{val}}q_{\mathrm{val}} + A_{\mathrm{reg}}q_{\mathrm{reg}} - q^{\mathrm{set}}(t) \\ f_{\mathrm{pb}}(p) \\ f_{\mathrm{qb}}(q_{\mathrm{pip,l}}, q_{\mathrm{pip,r}}, q_{\mathrm{val}}, q_{\mathrm{reg}}) \\ f_{\mathrm{cap}}(p) \end{pmatrix}.$$

Suppose the function $z(p)$ describing the real gas factor is differentiable, then also $\psi(x)$ is differentiable and the gas network DAE (10) can be written in standard form as

$$f_s(x'(t), x(t), t) = 0 \tag{11}$$

with

$$f_s(y, x, t) := A\psi'(x)y + b(x,t).$$

Notice that we also may equivalently formulate the gas network DAE (10) with a properly stated derivative term. For that purpose, we collect the flows of all valves and regulating elements that are modelled with a dynamic term (corresponding to $\mu_e \neq 0$) in $\bar{q}_{\mathrm{val}}$ and $\bar{q}_{\mathrm{reg}}$, respectively. Additionally, we delete the (zero) columns in D_{val} and D_{reg} corresponding to the flows of valves and regulating elements that are modelled without a dynamic term ($\mu_e = 0$) and denote them by $\bar{D}_{\mathrm{val}}$ and $\bar{D}_{\mathrm{reg}}$. Furthermore, we delete all node entries from $\phi(p)$ for each node that does not have a right end of a pipe connected to it. We denote the result as $\bar{\phi}(p)$. Correspondingly, we delete all zero columns of $A_{\mathrm{pip,r}}^\top$ and $\bar{A}_{\mathrm{pip,r}}^\top$ represents the remaining matrix. Then, the gas network DAE (10) is equivalent to

$$f_p\Big(\frac{\mathrm{d}}{\mathrm{d}t}d_p(x(t)), x(t), t\Big) = 0 \tag{12}$$

with $f_p(y, x, t) := A_p y + b(x,t)$ and

$$A_p = \begin{pmatrix} \bar{A}_{\mathrm{pip,r}}^\top & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & \bar{D}_{\mathrm{val}} & 0 \\ 0 & 0 & 0 & \bar{D}_{\mathrm{reg}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad d_p(x) := \begin{pmatrix} \bar{\phi}(p) \\ q_{\mathrm{pip,l}} \\ \bar{q}_{\mathrm{val}} \\ \bar{q}_{\mathrm{reg}} \end{pmatrix}.$$

Both representations of the gas network equations (10) with or without the technical bounds (9f) are special cases of general nonlinear differential algebraic equations of the form

$$f\Big(\frac{\mathrm{d}}{\mathrm{d}t}d(x(t)), x(t), t\Big) = 0, \tag{13}$$

where $f : \mathbb{R}^k \times \mathbb{R}^m \times [a,b] \to \mathbb{R}^n$. With a proper choice of the pipe directions and some additional conditions to the positions of valves and regulators, the resulting DAE system has index 1 [HT17; Huc18]. Next, we introduce the least-squares collocation approach introduced in [HMT19; HM21a] for (13) combined with the initial value condition

$$x(a) = x_0. \tag{14}$$

Note that x_0 is assumed to be consistent, that means there is a continuous solution x of (13) on $[a,b]$ that satisfies $x(a) = x_0$. The least-squares collocation approach computes continuous, piecewise polynomial solutions x_p, i.e., solutions belonging to the space

$$X_p := \{x \in C([a,b], \mathbb{R}^m) \mid x_{|_{I_k}} \in P_N(I_k)\ \forall k = 1, ..., K\}$$

where $[a,b]$ is decomposed into subintervals I_k with $k = 1, ..., K$ and $P_N(I_k)$ is the set of polynomials on I_k with degree at most N. We speak of a least-squares collocation solution of (13) if $x_p \in X_p$ with $x_p(a) = x_0$ is a least-squares solution of

$$f\Big(\frac{\mathrm{d}}{\mathrm{d}t}d(x_p(t_p)), x_p(t_p), t_p\Big) = 0, \qquad t_p \in T_p \tag{15}$$

with the finite set $T_p \subset [a,b]$ of collocation points. For simplicity, we choose the same number M of collocation points in each subinterval. In case of classical collocation, we have $M = N$. Here, we also allow $M > N$.

Let $\{b_{k0}(t), ..., b_{kN}(t)\}$ be a set of polynomial basis functions spanning $P_N(I_k)$. Then, we can write

$$x_p(t) = \sum_{i=0}^{N} b_{ki}(t) c_{ki} \quad \text{on } I_k. \tag{16}$$

We consider two possible choices, firstly the monomial basis and secondly the Lagrange basis. For the monomial basis representation we choose $t_k \in I_k$ for $k = 1, \ldots, K$ and have

$$b_{ki}(t) = (t - t_k)^i.$$

For the Lagrange basis we choose $t_{k0}, \ldots, t_{kN} \subset I_k$ for $k = 1, \ldots, K$ and have

$$b_{ki}(t) := \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{t - t_{kj}}{t_{ki} - t_{kj}}.$$

Note that $c_{ki} = x_p(t_{ki})$ for the monomial basis whereas $c_{ki} = x_p^{(i)}(t_k)/(i!)$ for the Lagrange basis. The collocation equations have then the form

$$f\Big(g_t'(t_p, c), \sum_{i=0}^{N} b_{ki}(t_p) c_{ki}, t_p\Big) = 0, \qquad t_p \in T_p \cap I_k, \qquad k = 1, ..., K, \tag{17}$$

with $c := (c_{ki})_{k=1,\ldots K, i=0,\ldots,N}$ and

$$g(t,c) := d(x_p(t)) = d\Big(\sum_{i=0}^{N} b_{ki}(t) c_{ki}\Big).$$

Consequently, we have to solve the overdetermined nonlinear root problem

$$\hat{F}(c) = 0 \tag{18}$$

with

$$\hat{F}(c) := \begin{pmatrix} \vdots \\ f\Big(g_t'(t_p, c), \sum\limits_{i=0}^{N} b_{ki}(t_p) c_{ki}, t_p\Big) \\ \vdots \end{pmatrix}, \quad t_p \in T_p \cap I_k, \quad k = 1, ..., K.$$

5 Solving Overdetermined Nonlinear Root Problems using Nonsmooth Optimization

Overdetermined nonlinear and possibly nonsmooth systems of equations like the one given in Eq. (18) can be formulated as optimizations problems using an appropriate norm of the resulting vector. Then, an argument that yields zero as value of the objective function corresponds to a solution of the system of equations. The employed norm as well as the equations considered may lead to a nonsmooth target function. In this paper, we will consider the specific target function

$$F(c) = \|\hat{F}(c)\|_2^2 \tag{19}$$

originating from the least-squares collocation as described in the last section. Note that the norm that we use here does not cause additional nonsmoothness.

The solution of nonsmooth optimization problems is still a challenging task. When the objective is convex, subgradient methods can be used, but they converge only with a sublinear rate, see, e.g., [Sho98, Chapter 2]. A better rate of convergence can be shown for bundle methods and gradient sampling, see, e.g., [KM10] or [BLO05]. However, these methods require the choice of various parameters and the performance is somewhat erratic. For a large class of piecewise smooth function, a completely new solution approach called Successive Abs-Linear MINimization (SALMIN) was developed in recent years, see, e.g., [FWG19]. Its convergence analysis is complete and even quadratic convergence can be achieved under suitable circumstances [GW19b]. For the SALMIN approach, it is assumed that the nonsmoothness of the target function $F(.)$ is caused by the evaluation of the absolute value function only. Then, at a given argument c, a local, piecewise linear model denoted by $\Delta F(c;.)$ can be derived using the technique of abs-linearization as introduced in [Gri13]. In the same paper, it was shown that $\Delta F(c;.)$ provides a second order approximation of $F(.)$, a property that serves as one of the main building blocks in the convergence analysis of the SALMIN approach. Another desirable property of these piecewise linearizations is that their representation can be factored and normalised into a specialized block matrix-based representation called *abs-linear form* (ALF). Hence, ALFs are operator representations of piecewise linear functions that can be realized or implemented as a slight as well as modular update on top of any existing modern matrix or linear algebra software package [Gri+15; Str+14]. Only small changes of existing tools for algorithmic differentiation like ADOL-C [WG12], cycADa [cyc21] or Tapenade [HP13] were required to obtain the local abs-linear model $\Delta F(c;.)$ in ALF in a completely automated way. Hence, this ingredient required by SALMIN is available once a code to evaluate $F(.)$ is available. For the numerical experiments in the subsequent section we used cycADa as tool for algorithmic piecewise linearization and numpy [Har+20] as package for linear algebra and matrix manipulation to generate the ALF representations of the local abs-linear model $\Delta F(c;.)$. Finally, cython [Beh+11] allowed us to connect the python parts of our implementations with the `C++` kernel of SALMIN.

Keeping it very brief and denoting the vector of optimization variables in iteration k with c^k, the optimization approach of SALMIN can be described by the iteration

$$c^{k+1} = c^k + \underset{\Delta c}{\arg\min} \left\{ \Delta F(c^k; \Delta c) + \frac{\omega^k}{2} \|\Delta c\|^2 \right\}. \tag{20}$$

As can be seen, only one parameter, namely the penalty factor ω^k, occurs. Starting from an initial choice ω^0, the convergence analysis of SALMIN also provides an automatic adaptation of this parameter. Given the structure as stated in Eq. (20), SALMIN can be interpreted as a quadratic overestimation method, where the error between the local, piecewise linear model $\Delta F(c;.)$ and the real function $F(.)$ is bounded by a power of the distance, see, e.g., [Gri81]. The SALMIN approach is in some sense also related to a proximal point method. However, in Eq. (20) the local abs-linear model of the function $F(.)$ to be minimized at the current iterate c^k is used instead of the original nonlinear function $F(.)$. This fact makes the inner arg min problem of Eq. (20) considerably easier to solve in comparison to usual proximal point approaches. An adapted algorithm with provable finite convergence to determine a solution Δc^* of the corresponding piecewise linear objective with a quadratic penalty term was proposed and analysed in [GW19a]. Furthermore, the solution Δc^* provides a very simple stopping criterion for the SALMIN approach in that the outer iteration terminates as soon as the objective function reduction promised by the solution of the inner problem falls below a user supplied tolerance. Then, an at least approximately stationary point is reached. Hence, no generalized gradients or $\varepsilon-$subdifferentials have to be provided and evaluated.

6 Numerical Experiments and Data Availability

In this section we present three distinct instances of gas network simulations. An instance contains all necessary information for the creation of a network model (9). These information are usually provided in the form of data sheets using `csv`, `xml`, `json` or similar data formats and markup languages. Furthermore, this data is typically divided and grouped into static information mostly regarding the network's topology or dynamic information making up a scenario. This is a useful separation as it is possible to have different scenarios addressing the same network topology. The GasLib data base [Sch+17a; Sch+17b], although offering instances for time-stationary optimization of gas networks only, follows the same data separation paradigm.

The details on all three instances presented here are publicly available. All necessary data regarding the first instance is presented in this paper. The data regarding the second instance which we referred to as *the single control valve* can be found in [HPS21]. The data of the 3rd or *modified GasLib-40* is contained in [Str21]. Being called *modified* the third instance is derived from the *GasLib-40* instance from the aforementioned GasLib data base (see [Sch+17a; Sch+17b] or `gaslib.zib.de`). Additional dynamical instances can also be found within [Ben+19].

The focus of our numerical experiments is on the employed routines to solve the overdetermined root problems, i.e., (18) or alternatively (19), created by the outer least-squares collocation integrator as described in Section 4. More precisely, we want to compare SALMIN with the *trust region reflective algorithm mode* of `scipy.optimize.least_squares` from scipy [Vir+20]. We will henceforth refer to the latter simply as scipy-ls-trf for short. The scipy-ls-trf solver is dedicated to solve differentiable overdetermined root problems in a least-squares sense. The robustness and accessibility of scipy methods and routines turned them into a quasi-standard in python for numerical and computational science research to solve various optimization and root solving tasks. Hence, the two central questions we aim to contribute to are:

1. Is scipy-ls-trf a viable option within the simulation of piecewise differentiable gas network instances although its convergence theory does not apply?
2. How does SALMIN perform in comparison?

For each instance we will compare in total four configurations of the two solvers. The four configurations either address (18) directly or solve (19) instead. Furthermore, some configurations will be fed with automatically computed derivative information whilst other rely on their build-in finite differences scheme. By the term *derivative information* we refer to the Jacobian matrix for scipy-ls-trf or the abs-linear form (ALF, see Section 5) for SALMIN yielding the following four configurations:

- *SALMIN* solving (19) and provided with ALFs computed by algorithmic differentiation,
- *scipy-ls-trf-vec-jac* solving (18) and provided with Jacobian matrices computed by algorithmic differentiation,
- *scipy-ls-trf-jac* solving (19) and provided with Jacobian matrices computed by algorithmic differentiation,
- *scipy-ls-trf-nojac* solving (19) and approximating Jacobian matrices internally by finite differences.

Furthermore, we will compare the performance of all four configurations while using monomial as well as Lagrange basis functions for the representation for all polynomials from each subinterval of the piecewise polynomial $x_p \in X_p$. Hence, we deal with results from eight simulation runs per instance, i.e., all four configuration times the two polynomial bases in consideration. We will use a constant time resolution or constant time step width $h > 0$ in the minute range per instance. Finally, and to conclude the documentation of the experiment parameters, the collocation points $T_p \cap I_k$ per subinterval $I_k = [a_k, b_k]$ will be chosen as

$$T_p \cap I_k \equiv \{a_k + \tfrac{i+1}{M} \cdot (b_k - a_k) \mid i = 0, 1, \ldots, M-1\},$$

where $M = N + 1$ (in the sense of Section 4).

Ultimately we will compare all configurations in terms of the number of function evaluations (*nfev*) as well as how many generations of derivative operators (*njev*) have been invoked per solver configuration to fully integrate the corresponding instance numerically. The shorthands *nfev* and *njev* are part of a nomenclature taken over from scipy directly. Of course the letter j within *njev* were supposed to refer to term *Jacobian matrix* which makes sense in context of scipy methods and solvers. In our experiments, however, the phrase derivative operator refers to the ALF for the first configuration *SALMIN*, to the algorithmically computed Jacobian for the *scipy-ls-trf-vec-jac* as well as *scipy-ls-trf-jac* configurations or to the finite difference approximation of a Jacobian in context of the last or *scipy-ls-trf-nojac* configuration.

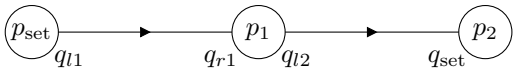


Figure 1: Two pipes example net topology

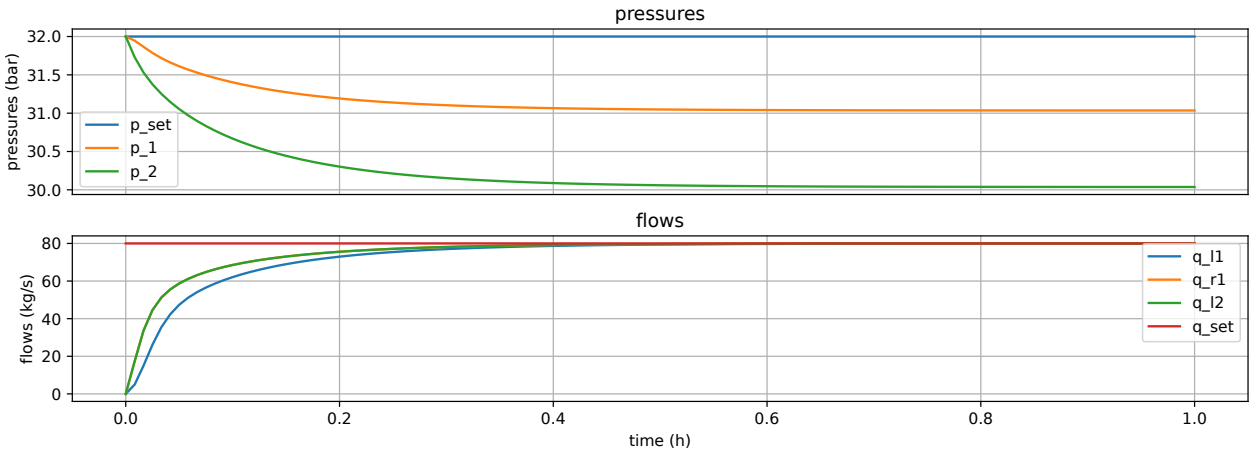


Figure 2: Simulation result for the *two pipes* instance

6.1 Two Pipes

The first instance serves as a validation of the functionality of the SALMIN solver. We consider the simple network of two connected pipes as depicted in Figure 1.

The pressure at the source node on the left is fixed at $p_{\text{set}} = 32$[bar], while the flow going into the sink node on the right is fixed at $q_{\text{set}} = 80\left[\frac{\text{kg}}{\text{s}}\right]$. The initial values of the other pressures are given by the same value of 32[bar], the other flows are starting with a value of $0\left[\frac{\text{kg}}{\text{s}}\right]$. The simulation has been run with a step size of $h = 30$s.

The simulation results can be seen in Figure 2. The observed results are nearly identical to those obtained by an implicit Euler scheme and the residuals of the system equations, starting at an order of 10^{-3}, quickly tend to zero, indicating a good approximation of the true solution.

The results also can be interpreted quite nicely. The network dynamics determine the gradient of the pressure p_2 to be proportional to the difference of flows $q_{r2} - q_{l2}$. Since q_{r2} is fixed by q_{set} and q_{l2} starts at $0\left[\frac{\text{kg}}{\text{s}}\right]$, a large flow difference leads to a quickly dropping pressure p_2. Likewise, the increasing pressure difference $p_2 - p_1$ implies a surge of the flow q_{l2}. Similar arguments hold for p_1 and q_{l1}. After about 35 minutes, as the dynamic flows align to the fixed flow q_{set}, the pressures also stabilize. The network dynamic enters the steady state that is uniquely determined by the boundary data.

Figures 3 and 4 demonstrate a particular dependence on the choice of polynomial basis functions already in this simple example. While the convergence of the different configurations is almost perfectly steady in the case of Lagrange basis functions, the configuration *scipy-ls-trf-vec-jac* shows a clearly unstable behavior when using monomial basis functions. Furthermore, *scipy-ls-trf-vec-jac* causes 30 to 60 times more function and Jacobian evaluations than *SALMIN* despite the fact that the resulting solution is far from the true one. *SALMIN* comes out ahead as the clear winner with the least amount of function and Jacobian evaluations and a stable convergence behavior in the monomial basis case. This is also true in the Lagrange basis case, albeit by a much more narrow margin. Table 3 aggregates the results.

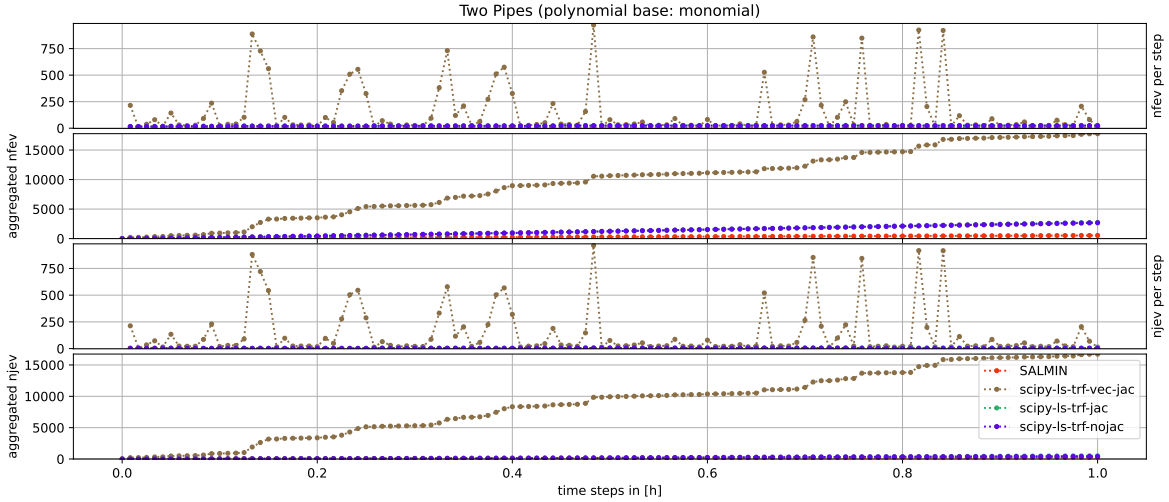


Figure 3: The *two pipes* instance – number of function evaluations *nfev* as well as number of Jacobian or ALF evaluations *njev* per configuration and per time step utilizing monomial basis functions

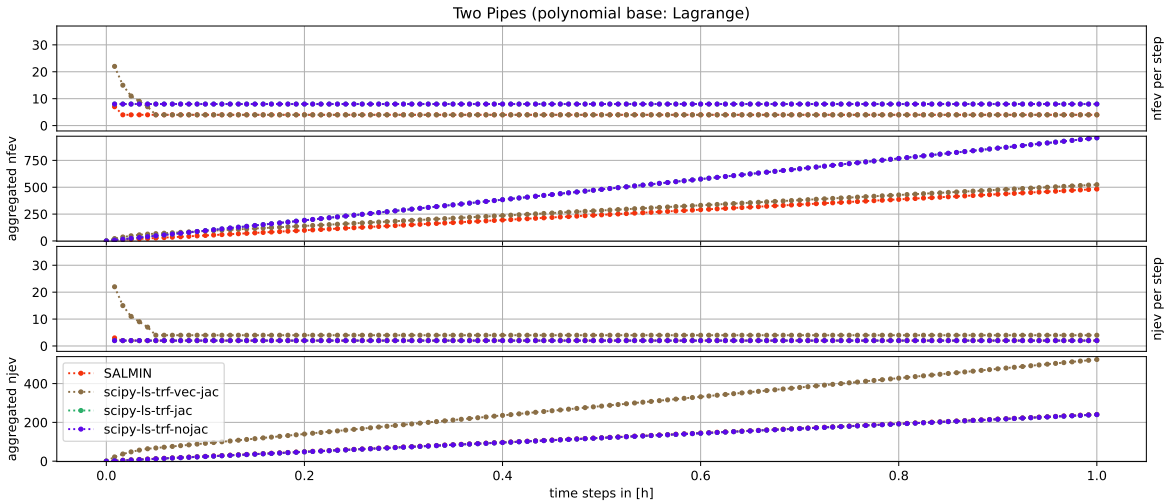


Figure 4: The *two pipes* instance – number of function evaluations *nfev* as well as number of Jacobian or ALF evaluations *njev* per configuration and per time step utilizing Lagrange basis functions

Configuration			Monomial base		Lagrange base	
label	solves	get derivative	*nfev*	*njev*	*nfev*	*njev*
SALMIN	(19)	yes	531	257	483	241
scipy-ls-trf-vec-jac	(18)	yes	**17759**	**16727**	524	524
scipy-ls-trf-jac	(19)	yes	**2745**	**507**	960	240
scipy-ls-trf-nojac	(19)	no	2678	400	960	240

Table 3: Aggregated table of total numbers per configuration, extracted from Figures 3 and 4

6.2 The Single Control Valve

The second instance should be considered a nonlinear, nonsmooth modelling stress test as well as a benchmark on simulating the active regulation behaviour of control valves. Although it is rather small in terms of its topology its complexity stems from many changes of the target value controls. This instance had been hand crafted by experts from the gas transportation industry solely as a demonstration of the hierarchical logic behind the many different target values in use for control valves from gas networks [HPS21]. Hence, it never was nor is supposed to resemble or reflect a desired control in the day-to-day transport operation. On the contrary, this instance, i.e., the scenario, should be considered extreme in that it keeps itself artificially busy throughout. However, simulation technology has to be able to withstand and solve it successfully.

The network comprises two pipes and a single control valve in a sequential configuration (see Figure 5). At any point in time, $10\left[\frac{\text{kg}}{\text{s}}\right]$ are supplied at the source node IN0 on the left but simultaneously also taken from the sink node OUT0 on the right. In other words, the networks supply and demand are constant and always balanced. An optimal control of the control valve in the meaning of efficient network operation would be to do nothing and let the flow pass unhindered. However, network efficient operation is not the goal here.

Instead, the following changes in control take place over time: Initially the target value for the flow q^{set} (or q_set in Figure 5) forces the passing gas flow down onto $9\left[\frac{\text{kg}}{\text{s}}\right]$, which the control valve actually can realize by creating resistance. Correspondingly, the control valve builds up a pressure difference, as the left pressure p_ℓ (or p_l in Figure 5) is rising and the right pressure p_r (or p_r in Figure 5) is falling. After the first hour, the target flow lifts up and pulls the actual flow which leads to a rapid decline of the pressure difference until it gets balanced. After balancing the input-to-output pressure ratio the actual flow returns and stabilizes at $10\left[\frac{\text{kg}}{\text{s}}\right]$, although the target flow still demands a higher flow, but control valves are not allowed to compress gas. At the beginning of the second hour the target flow drops down to $6\left[\frac{\text{kg}}{\text{s}}\right]$ and pulls the actual flow with it. Once more, this leads to a build up in pressure difference as well. Half an hour later, the target value for the flow as well as the actual flow will return together to $10\left[\frac{\text{kg}}{\text{s}}\right]$ and hence the pressure difference becomes constant. Within the fourth hour, the upper limiting target value for the right pressure $\overline{p_r}$ (or p_r_upper from Figure 5) suddenly demands the actual right pressure to be reduced down to 47[bar]. The control valve is able to achieve this by shutting tightly momentarily. It is important to note here that any target value for pressures overrides the target value for flow. In the transition from the fifth into the sixth hour, basically the same happens again but driven by a sudden rise in the lower limiting target value for the left pressure $\underline{p_l}$ (or p_l_lower in Figure 5). The next three incidents take place between the sixth and the eighth hour, where we can observe three spikes in the flow as a reaction of the right pressure p_r (p_r) being raised three times by lowering the limiting target value for the right pressure $\underline{p_r}$ (or p_r_lower in Figure 5) in rather small incremental steps. However, the right pressure does not follow $\underline{p_r}$ (p_r_lower) all the way up to 47[bar] during the third jump, because the left pressure p_ℓ hits its lower limiting target value $\underline{p_l}$ (p_l_lower). Thus, as a second important fact we point out that target values for pressures that potentially decrease the gas flow overrule those that potentially increase it.

One first clear observation that can be obtained from the Figures 6 and 7 or Table 4 is the substantial impact of the choice of the polynomial base. The various scipy related configurations are affected to various degrees. When using monomial basis functions both Jacobian-fed scipy based configurations, i.e., *scipy-ls-trf-jac* as well as *scipy-ls-trf-vec-jac*, demonstrate a very unstable convergence behaviour. They generate outlier statistics in Table 4 requiring 20 up to a 100 times more function evaluations and derivative operator computations. Furthermore, their demand is fluctuating considerably from time step to time step. In contrast to this the same two configurations prove to be quite competitive when using Lagrange basis functions. A rather strange side effect which we have double- and triple-checked is that the pair of *scipy-ls-trf-jac* and *scipy-ls-trf-nojac* suddenly have an identical performance under the use of the Lagrange basis. However, the single *SALMIN* configuration appears almost unaffected and hence rather robust against the choice of the set of basis functions. Using either basis *SALMIN* always requires the least number of function evaluations as well as the second and third lowest amount of derivative, i.e., ALF, computations of all configurations. There is one noteworthy feature shared by *SALMIN* and *scipy-ls-trf-vec-jac* while using the Lagrange base, namely the appearance of one spike in *nfev* and *njev* on a single time spot when the target value $\underline{p_r}$ pulls the pressure curve p_r to rise up, which in turn causes the flow to spike up, too. The magnitude of the flow spike correlates directly with the time step width or resolution h. This may be considered a deficiency of the underlying control valve model which could be easily fixed by introducing slope limiter $\text{abs}(\dot{q}) \leq c_{\dot{q}}$ to cap the speed of change of the flow. Such a slope limiter would also improve the behaviour of the *SALMIN* and *scipy-ls-trf-vec-jac* configuration.

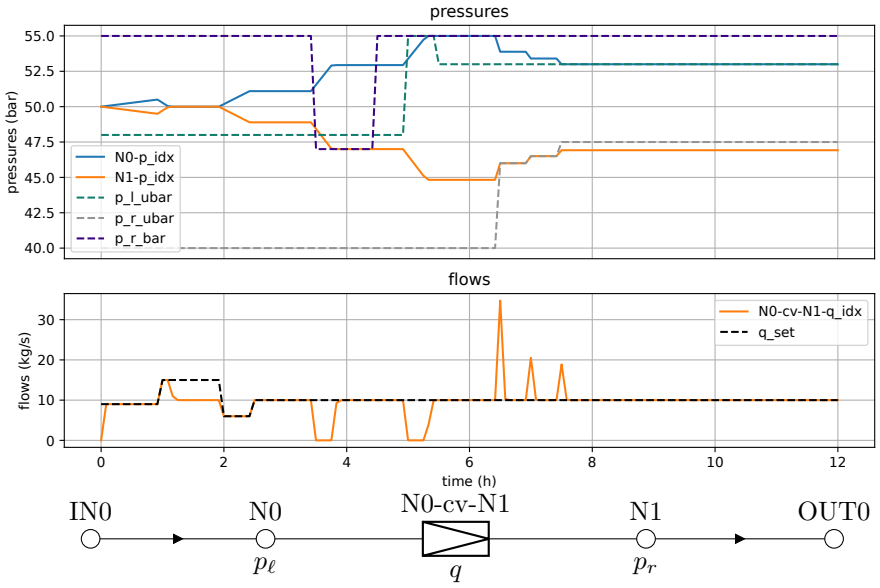


Figure 5: The *single control valve* instance – (solid) solution and (dashed) target values to pressures (in the top) and the flow (in the bottom) of the control valve; net topology plotted in between the graphs

Configuration			Monomial base		Lagrange base	
label	solves	get derivative	*nfev*	*njev*	*nfev*	*njev*
SALMIN	(19)	yes	786	366	849	402
scipy-ls-trf-vec-jac	(18)	yes	**17891**	**17836**	901	867
scipy-ls-trf-jac	(19)	yes	**85426**	**82475**	1152	288
scipy-ls-trf-nojac	(19)	no	3851	509	1152	288

Table 4: Aggregated table of total numbers per configuration, extracted from Figures 6 and 7

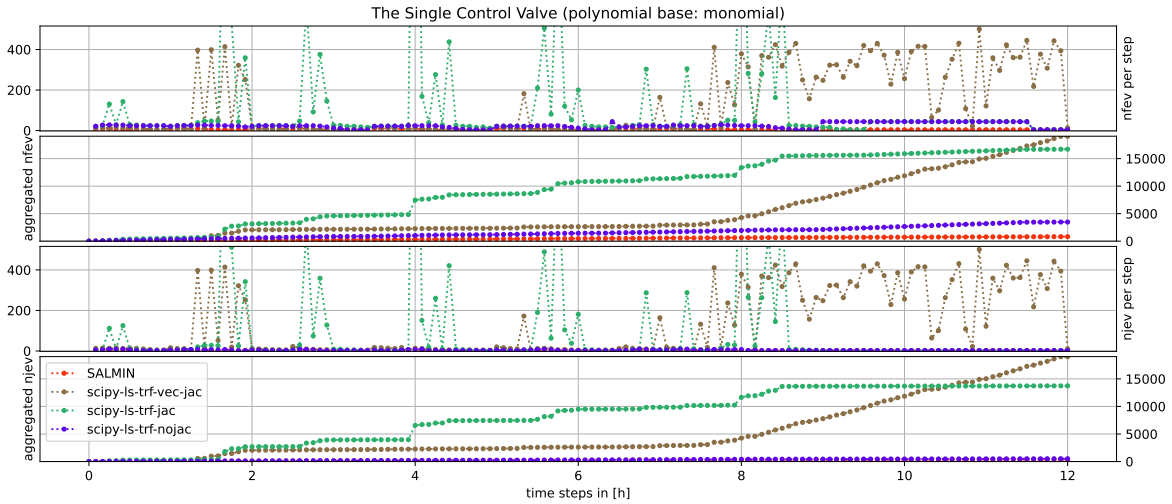


Figure 6: The *single control valve* instance – number of function evaluations *nfev* as well as number of Jacobian or ALF evaluations *njev* per configuration and per time step utilizing monomial basis functions

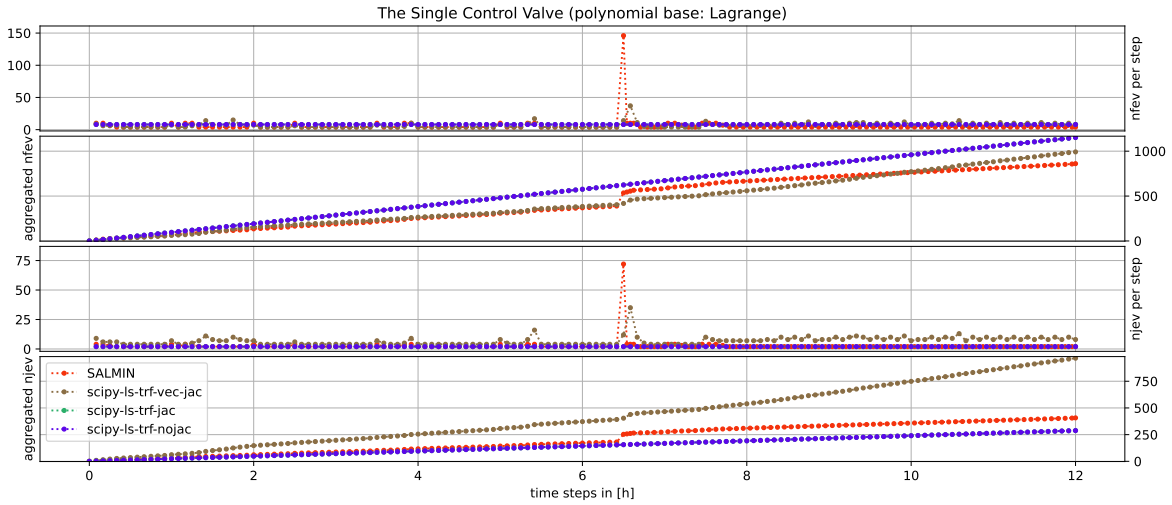


Figure 7: The *single control valve* instance – number of function evaluations *nfev* as well as number of Jacobian or ALF evaluations *njev* per configuration and per time step utilizing Lagrange basis functions

6.3 The Modified GasLib-40

The final instance brings everything together into a small-sized network of 70 nodes that had been derived from the GasLib-40 instance contained in `gaslib.zib.de` (see [Sch+17a; Sch+17b; Sch+17c]) which originally does consist of 40 nodes. The full list of differences between the original and the derived modified version can be found in [Str21]. However, the most relevant differences include

- relocation of all entry and exit nodes, where gas can be supplied and taken,
- introduction of a complex gas network station at the hearth of the network, allowing complex routing strategies to take control over transport operations,
- addition of a time-dynamical, i.e., transient scenario.

The gas network station consists of a multitude of valves, a single control valve and three ideal compressors. The transient scenario makes use of three distinguished principal configurations of the gas network station. All three principal configurations are at display within Figure 8. As a brief reminder the scenario details are provided in [Str21]. Within the first ten hours, using the principal configuration A, gas is pulled from the north west and north east, gets compressed and pushed into the south. Within the next ten hours, using the principal configuration B, gas from the north east gets distributed and pushed to the north west as well as the south. In the last chunk of about ten hours, using the principal configuration C, the north east gets disconnected from the gas network station while gas from the south is supplied to the north west.

Once more, we observe a distinct behaviour comparing simulations when using monomial versus Lagrange basis functions. Another similarity between this instance and the single control valve instance is that the simulations from the configurations *scipy-ls-trf-jac* and *scipy-ls-trf-nojac* when using Lagrange basis functions share an identical performance according to Table 5. However, in many other aspects the results draw a different picture in comparison to the single control valve instance. This time *scipy-ls-trf-jac* and *scipy-ls-trf-nojac* appear to suffer in terms of *nfev* from choosing the monomial basis according to Table 5. On the other hand, *scipy-ls-trf-nojac* requires the least amount of Jacobian approximations, i.e., *njev*, although *SALMIN* also requires relatively few computations of ALFs. More precisely, *SALMIN* requires the second and third smallest amount of derivative computations, i.e., *njev*, across both polynomial basis options. In other words, while different options and parameters fed to scipy have their sweet spots here and there it is actually *SALMIN* offering consistently stable and low costs for solving this instance.

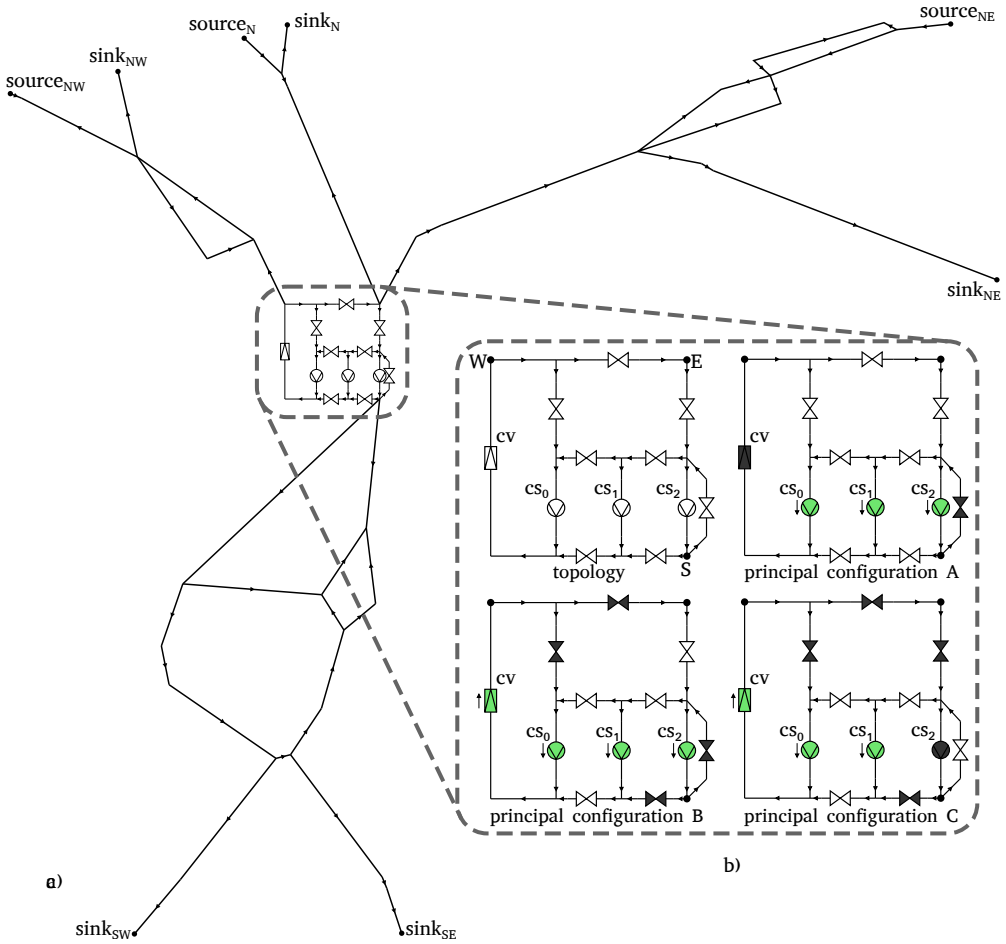


Figure 8: The *modified GasLib*-40 instance – figure derived from Figure 5.20 from [Str21]; network display of mod-g40 a network derived from GasLib-40 (see [Sch+17a; Sch+17b; Sch+17c] as well as `gaslib.zib.de`): **a)** general network topology as gas network circuit; **b)** enlargement of network station and three principal configurations A, B and C, where white elements allow unmanaged bi-directional flow, dark grey elements blocking flow in any direction, light green elements allow and actively control one-directional flow

Configuration			Monomial base		Lagrange base	
label	solves	get derivative	*nfev*	*njev*	*nfev*	*njev*
SALMIN	(19)	yes	588	277	435	209
scipy-ls-trf-vec-jac	(18)	yes	394	362	503	484
scipy-ls-trf-jac	(19)	yes	**15816**	**13869**	744	186
scipy-ls-trf-nojac	(19)	no	**2712**	186	744	186

Table 5: Aggregated table of total numbers per configuration, extracted from Figures 9 and 10

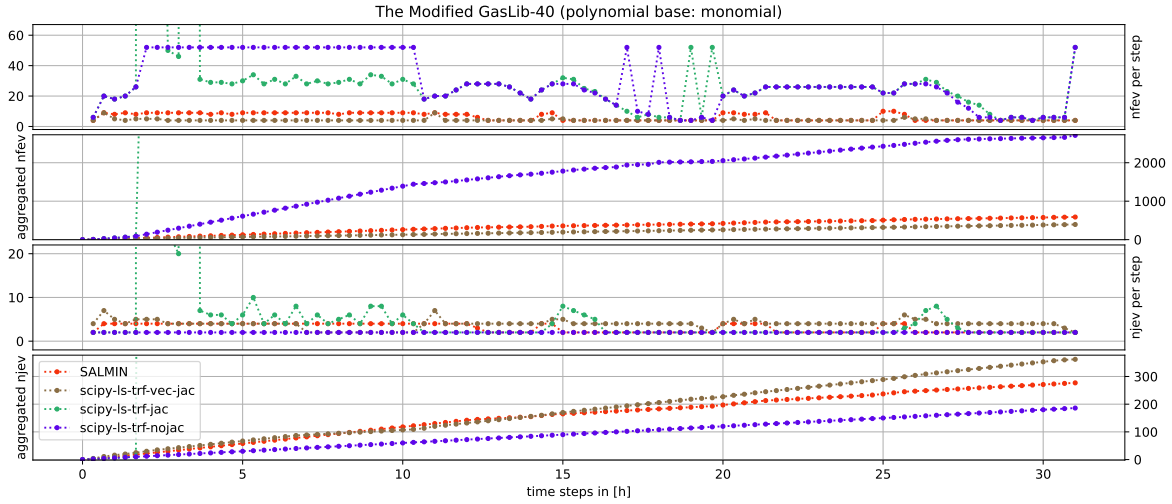


Figure 9: The *modified GasLib*-40 instance – number of function evaluations *nfev* as well as number of Jacobian or ALF evaluations *njev* per configuration and per time step utilizing monomial basis functions

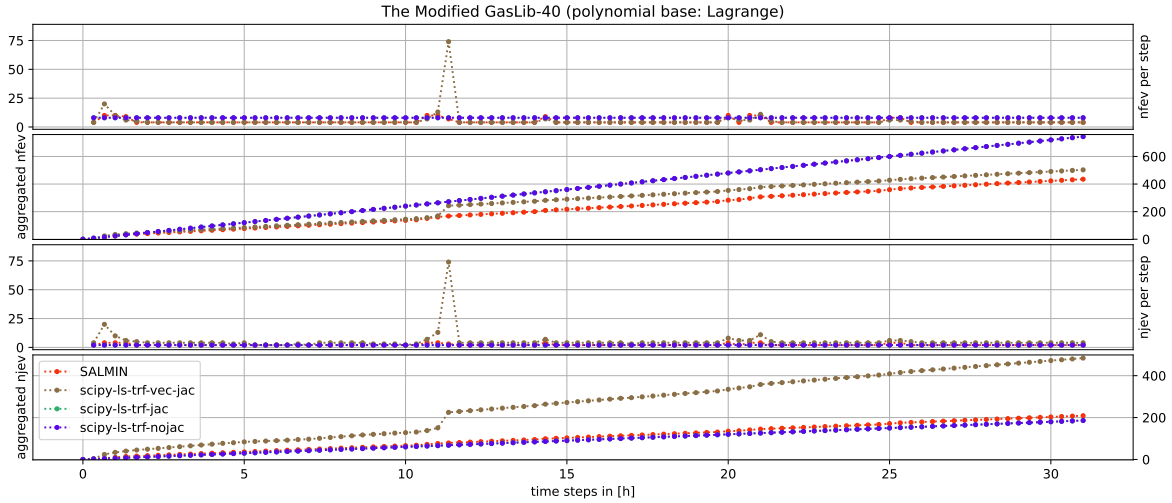


Figure 10: The *modified GasLib*-40 instance – number of function evaluations *nfev* as well as number of Jacobian or ALF evaluations *njev* per configuration and per time step utilizing Lagrange basis functions

7 Conclusion and Outlook

We presented a modelling of gas networks with advanced features using differential algebraic equations including models for valves, regulators, compressors and a topologically adaptive discretization of the isothermal Euler equations for pipes and pipelines. In addition, the models for control valves and compressors allow their control via target values reflecting their real world use within the dispatching of gas networks. For the actual numerical simulation, a least-squares collocation approach is presented, where for each time step an overdetermined system of nonsmooth nonlinear equations has to be solved. Besides targeting the system of equations directly, we also propose to reformulate this task as the solution of a nonsmooth nonlinear optimization problem.

Four different variants for solving either the system of equations or the reformulated optimization problem using either algorithmically computed or approximated derivatives are considered. Three of these approaches are already widely used but ignore the inherent nonsmoothness of the underlying problems. The fourth approach relies on SALMIN, which is a dedicated optimization algorithm for nonsmooth problems. This new approach explicitly exploits the nonsmooth character of the optimization problem.

The obtained numerical results validate the proposed approach based on a modelling using DAEs in combination with a least-squares collocation and the nonsmooth solver SALMIN. Furthermore, it is illustrated that this nonsmooth optimization approach using abs-linearization outperforms the already established solvers in several ways. For once, the numbers of function evaluations and derivative computations, gathered to measure the computational work, as required by SALMIN range among the smallest, hence most efficient of all four choices across all gas network simulation instances. In addition to that, all three other options are very sensitive regarding the choice of the polynomial basis functions used to represent the numerical solution. Although we can achieve good results when using the Lagrange basis, the three classical approaches do suffer from choosing the monomial basis instead and without hinting a clear pattern in which way or why exactly. In total contrast the SALMIN approach does not suffer from the choice of either polynomial basis in consideration, instead always achieving good performance at equally good results.

The three gas network instances in consideration have been carefully designed with the ideal of bringing mathematics into application in mind. Each one of these instances is relevant as it adopts real world aspects. Hence, this feasibility study proves the potential of implementing and integrating our approaches, the overall framework and its models into the gas transportation industry. Furthermore, we pave the way for further theoretical studies comprising for example the theory concerning the proposed integration approach like error estimators. Here, one could also take the inexact solution of the optimization problems in each time step into account. Furthermore, nonsmooth generalization of collocation incorporating abs-polynomial expansions as developed within [Str21; STG21; GST21] should be considered next.

Acknowledgments

The authors thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for their support within subproject B10 and subproject C02 in the CRC TRR154-2 “Mathematical Modelling, Simulation and Optimization using the Example of Gas Networks” (Project-ID 239904186).

References

[Bag+20] A. Bagirov, M. Gaudioso, N. Karmitsa, M. Mäkelä, and S. Taheri, eds. *Numerical nonsmooth optimization. State of the art algorithms*. Cham: Springer, 2020. DOI: 10.1007/978-3-030-34910-3.

[Beh+11] S. Behnel et al. “Cython: The best of both worlds”. In: *Computing in Science & Engineering* 13.2 (2011), pp. 31–39.

[Ben+19] P. Benner et al. “Gas Network Benchmark Models”. en. In: ed. by S. Campbell, A. Ilchmann, V. Mehrmann, and T. Reis. Differential-Algebraic Equations Forum. Cham: Springer International Publishing, 2019, pp. 171–197. DOI: 10.1007/11221_2018_5.

[Bro02] G. O. Brown. “The History of the Darcy-Weisbach Equation for Pipe Flow Resistance”. In: *Proc. Environ. Water Resour. Hist.* 38 (2002), pp. 34–43. DOI: 10.1061/40650(2003)4.

[BLO05] J. Burke, A. Lewis, and M. Overton. "A robust gradient sampling algorithm for nonsmooth nonconvex optimization". In: *SIAM Journal on Optimization* 15.3 (2005), pp. 751–779. DOI: 10.1137/030601296.

[CW37] C. F. Colebrook and C. M. White. "Experiments with Fluid Friction in Roughened Pipes". In: *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* 161.906 (1937), pp. 367–381. DOI: 10.1098/rspa.1937.0150.

[cyc21] cycADa. *cycADa - cython based algorthmic differentiation*. https://gitlab.com/berlinade/cycada-algorithmic-differentiation-in-python. (online; last accessed 2021-08-22). 2021.

[Dom+21] P. Domschke et al. "Gas Network Modeling: An Overview". 2021. URL: https://opus4.kobv.de/opus4-trr154/frontdoor/index/index/docId/411.

[Eur20] European Commission. *COMMUNICATION FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT, THE COUNCIL, THE EUROPEAN ECONOMIC AND SOCIAL COMMITTEE AND THE COMMITTEE OF THE REGIONS - A hydrogen strategy for a climate-neutral Europe*. English. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020DC0301. Accessed: 2021-11-22. 2020.

[Fed21] Federal Ministry for Economic Affairs and Energy. *Still Indispensable for a Reliable Energy Supply*. English. https://www.bmwi.de/Redaktion/EN/Dossier/conventional-energy-sources.html. Accessed: 2021-11-22. 2021.

[FWG19] S. Fiege, A. Walther, and A. Griewank. "An algorithm for nonsmooth optimization by successive piecewise linearization". In: *Mathematical Programming* 177.1-2 (A) (2019), pp. 343–370.

[Gri81] A. Griewank. *The modification of Newton's method for unconstrained optimization by bounding cubic terms*. Tech. rep. NA/12. University of Cambridge, 1981.

[Gri13] A. Griewank. "On Stable Piecewise Linearization and Generalized Algorithmic Differentiation". In: *Optimization Methods and Software* 28.6 (2013), pp. 1139–1178. DOI: 10.1080/10556788.2013.796683.

[Gri+15] A. Griewank, J.-U. Bernt, M. Radons, and T. Streubel. "Solving piecewise linear systems in abs-normal form". In: *Linear Algebra and its Applications* 471.0 (2015), pp. 500–530. DOI: https://doi.org/10.1016/j.laa.2014.12.017.

[GST21] A. Griewank, T. Streubel, and C. Tischendorf. "On the abs-polynomial expansion of piecewise smooth functions". In: *Optimization Methods and Software* 36.2-3 (May 2021), pp. 301–315. ISSN: 1055-6788. DOI: 10.1080/10556788.2020.1817448.

[GW19a] A. Griewank and A. Walther. "Finite convergence of an active signature method to local minima of piecewise linear functions". In: *Optimization Methods and Software* 34.5 (2019), pp. 1035–1055. DOI: 10.1080/10556788.2018.1546856.

[GW19b] A. Griewank and A. Walther. "Relaxing kink qualifications and proving convergence rates in piecewise smooth optimization." In: *SIAM Journal on Optimization* 29.1 (2019), pp. 262–289.

[HM21a] M. Hanke and R. März. "Convergence analysis of least-squares collocation methods for nonlinear higher-index differential–algebraic equations". In: *Journal of Computational and Applied Mathematics* 387 (2021), p. 112514. DOI: 10.1016/j.cam.2019.112514.

[HMT19] M. Hanke, R. März, and C. Tischendorf. "Least-squares collocation for higher-index linear differential-algebraic equations: Estimating the instability threshold". In: *Math. Comp.* 88 (2019), pp. 1647–1683. DOI: 10.1090/mcom/3393.

[Han+17] M. Hanke, R. März, C. Tischendorf, E. Weinmüller, and S. Wurm. "Least-squares Collocation for Linear Higher-index Differentialalgebraic Equations". In: *J. Comput. Appl. Math.* 317.C (2017), pp. 403–431.

[HM21b] M. Hanke and R. März. "A reliable direct numerical treatment of differential–algebraic equations by overdetermined collocation: An operator approach". In: *Journal of Computational and Applied Mathematics* 387 (2021). Numerical Solution of Differential and Differential-Algebraic Equations. Selected Papers from NUMDIFF-15, p. 112520. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2019.112520.

[HM21c] M. Hanke and R. März. "Towards a reliable implementation of least-squares collocation for higher index differential-algebraic equations-Part 1: basics and ansatz function choices". en. In: *Numerical Algorithms* (June 2021). ISSN: 1572-9265. DOI: 10.1007/s11075-021-01140-7.

[Har+20] C. R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[HP13] L. Hascoët and V. Pascual. "The Tapenade Automatic Differentiation tool: Principles, Model, and Specification". In: *ACM Transactions on Mathematical Software* 39.3 (2013), 20:1–20:43.

[HPS21] F. Hennings, M. Petkovic, and T. Streubel. "On the Numerical Treatment of Interlaced Target Values - Modeling, Optimization and Simulation of Regulating Valves in Gas Networks". eng. (unpublished manuscript). 2021.

[Hop+20] K. Hoppmann-Baum et al. *From Natural Gas towards Hydrogen - A Feasibility Study on Current Transport Network Infrastructure and Its Technical Control*. English. Tech. rep. 20-27. Takustr. 7, 14195 Berlin: ZIB, 2020.

[Huc18] C. Huck. "Perturbation analysis and numerical discretisation of hyperbolic partial differential algebraic equations describing flow networks". Dissertation. Humboldt Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät, 2018. DOI: 10.18452/19596.

[HT17] C. Huck and C. Tischendorf. "Transient Modeling and Simulation of Gas Pipe Networks with Characteristic Diagram Models for Compressors". In: *PAMM* 17.1 (2017), pp. 707–708. DOI: 10.1002/pamm.201710322.

[KM10] N. Karmitsa and M. M. Mäkelä. "Limited memory bundle method for large bound constrained nonsmooth optimization: convergence analysis". In: *Optimization Methods and Software* 25.6 (2010), pp. 895–916.

[LO13] A. Lewis and M. Overton. "Nonsmooth optimization via quasi-Newton methods". In: *Mathematical Programming Series A* 141.1-2 (2013), pp. 135–163.

[Nik50] J. Nikuradse. *Laws of Flow in Rough Pipes*. National Advisory Committee for Aeronautics Washington, 1950.

[Onn91] H. K. Onnes. "Expression of the equation of state of gases and liquids by means of series". In: *Through Measurement to Knowledge: The Selected Papers of Heike Kamerlingh Onnes 1853–1926*. Ed. by K. Gavroglu and Y. Goudaroulis. Dordrecht: Springer Netherlands, 1991, pp. 146–163. ISBN: 978-94-009-2079-8. DOI: 10.1007/978-94-009-2079-8_6.

[Pap68] J. Papay. "A Termeléstechnológiai Paraméterek Változása a Gáztelepek Muvelése Során". In: *OGIL Musz. Tud. Kozl.* (1968), pp. 267–73.

[Sal02] J. M. Saleh. *Fluid Flow Handbook*. McGraw-Hill Professional, 2002.

[Sch+17a] M. Schmidt et al. "GasLib – A Library of Gas Network Instances". In: *Data* 2.4 (2017), article 40. DOI: 10.3390/data2040040.

[Sch+17b] M. Schmidt et al. *GasLib-A Library of Gas Network Instances*. Tech. rep. Nov. 2017. URL: http://www.optimization-online.org/DB_HTML/2015/11/5216.html.

[Sch+17c] M. Schmidt et al. *GasLib-A Library of Gas Network Instances*. Tech. rep. Nov. 2017. URL: http://www.optimization-online.org/DB_HTML/2015/11/5216.html.

[Sch12] S. Scholtes. *Introduction to Piecewise Differentiable Equations*. SpringerBriefs in optimization. Springer New York, 2012. ISBN: 978-1-4614-4340-7. DOI: 10.1007/978-0-8176-8134-0.

[Sho98] N. Z. Shor. *Nondifferentiable Optimization and Polynomial Problems*. Kluwer, 1998.

[SL21] SIMONE Research Group and LIWACOM Informationstechnik GmbH. *SIMONE software – Equations and Methods*. en. 2021.

[Str21] T. Streubel. "Simulation of Piecewise Smooth Differential Algebraic Equations with Application to Gas Networks - Aspects of Modelling, Algorithmic Treatment and Numerical Analysis". eng. (unpublished manuscript). 2021.

[Str+14] T. Streubel, A. Griewank, M. Radons, and J.-U. Bernt. "Representation and Analysis of Piecewise Linear Functions in Abs-Normal Form". In: *System Modeling and Optimization*. Ed. by C. Pötzsche, C. Heuberger, B. Kaltenbacher, and F. Rendl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 327–336. DOI: 10.1007/978-3-662-45504-3_32.

[STG21] T. Streubel, C. Tischendorf, and A. Griewank. “Piecewise Polynomial Taylor Expansions–The Generalization of Faà di Bruno’s Formula”. en. In: *Modeling, Simulation and Optimization of Complex Processes HPSC 2018*. Ed. by H. G. Bock, W. Jäger, E. Kostina, and H. X. Phu. Cham: Springer International Publishing, 2021, pp. 63–82. ISBN: 9783030552404. DOI: 10.1007/978-3-030-55240-4_3.

[Vir+20] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[WG12] A. Walther and A. Griewank. “Getting Started with ADOL-C”. In: *Combinatorial Scientific Computing*. Chapman-Hall CRC Computational Science, 2012, pp. 181–202.