Adjoint-Based Control of Model and Discretization Errors for Gas Flow in Networks

Pia Domschke, Oliver Kolb and Jens Lang
Department of Mathematics, Technische Universität Darmstadt
Dolivostr. 15, D-64293 Darmstadt, Germany
Center of Smart Interfaces, Technische Universität Darmstadt
Petersenstr. 30, D-64287 Darmstadt, Germany
Graduate School Computational Engineering, Technische Universität Darmstadt
Dolivostr. 15, D-64293 Darmstadt, Germany

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Research Group
Numerical Analysis and
Scientific Computing

October 20, 2010

Abstract

We are interested in the simulation and optimization of gas transport in networks. The gas flow through pipes can be modelled on the basis of the (isothermal) Euler equations. Further network components are described by purely algebraic equations. Depending on the data and the resulting network dynamics, models of different fidelity can be used in different regions of the network. Using adjoint techniques, we derive model and discretization error estimators. Here, we apply a first-discretize approach. Based on the time-dependent structure of the considered problems, the adjoint systems feature a special structure and therefore allow for an efficient solution. A strategy that controls model and discretization errors to maintain the accuracy of the solution is presented. We provide (technical) details of our implementation and give numerical results.

1 Introduction

The flow of gas through pipelines is of great interest in the engineering community. The aim of operating a gas transmission network is to minimize the running costs of the compressor stations whereas the contractual demand of the customers has to be met. The enormous size of gas networks and their high complexity make the optimization a difficult computational task. In the field of simulation and optimization of gas transport in networks, a lot of research has been done in the last years [2, 5, 13, 14, 9, 4, 3, 10]. For the optimization, it is necessary to efficiently solve the underlying equations on networks within a certain tolerance. In this paper, we present an adaptive model switching and discretization algorithm that is suitable for these requirements.

The equations describing the transport of gas in pipelines are based on the Euler equations, a hyperbolic system of nonlinear partial differential equations, mainly consisting of the conservation of mass, momentum and energy. The transient flow of gas may be described appropriately by equations in one space dimension. Other components of the network, like compressor stations and valves, follow algebraic equations. For the whole network, adequate initial and boundary values as well as coupling conditions at the junctions are needed.

Although solving one-dimensional equations does not pose a challenge, the complexity increases with the size of the network. Thus, we use a hierarchy of three models that describe the flow of gas qualitatively different, but also with different computational effort. Obviously, simplified models are sufficient in network regions with low activity in the gas transport, while sophisticated models should be used to resolve high solution dynamics accurately.

Since the behaviour of the network changes both in space and time, an automatic steering of the model hierarchy as well as the discretization is essential. We introduce estimators for the discretization and model errors using adjoint techniques and present an algorithm to automatically control those errors with respect to a given tolerance.

Existent software packages like SIMONE [1] may also use different models for the simulation task. However, one model has to be chosen in advance, which is often too restrictive.

The paper is organized as follows. The modelling of the network as well as the different models are introduced in Sect. 2. In Sect. 3, error estimators for both, the model and the discretization error, are derived using adjoint techniques. We present a strategy to adaptively control model and discretization errors in Sect. 4. Section 5 gives more details about the implementation of the introduced error estimators. Numerical results are presented in Sect. 6.

2 Modelling of Gas Networks

In this section, we describe the modelling of the network. We introduce a hierarchy consisting of three different models describing the flow of gas through a pipe. Each model results from the previous one by making further simplifying assumptions [2]. The most complex model is the nonlinear model followed by the semilinear model. The simplest model used is the algebraic model (see Fig. 1). Also, further network components are given.

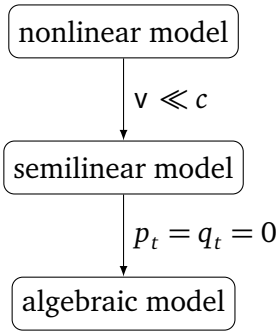

Figure 1: Model hierarchy

2.1 Network

The gas network is modelled as a directed graph $\mathcal{G} = (\mathcal{J}, \mathcal{V})$ with arcs $\mathcal{J}$ and vertices $\mathcal{V}$ (nodes, branching points). The set of arcs $\mathcal{J}$ consists of pipes $j \in \mathcal{J}_p$, compressor stations $c \in \mathcal{J}_c$ and valves $\mathsf{v} \in \mathcal{J}_\mathsf{v}$. Each pipe $j \in \mathcal{J}_p$ is defined as an interval $[x_j^a, x_j^b]$ with a direction from x_j^a to x_j^b. In each pipe, one of the models holds and adequate initial and coupling as well as boundary conditions have to be specified. Valves and compressor stations are described by algebraic equations.

2.2 Model Hierarchy Describing the Gas Flow

2.2.1 Nonlinear Model

The isothermal Euler equations, which describe the flow of gas, consist of the continuity and the momentum equation together with the equation of state for real gases. With some simplifying assumptions, as the pipe being horizontal and a constant speed of sound [3], the equations are

$$p_t + \frac{\rho_0 c^2}{A} q_x = 0\,, \tag{1a}$$

$$q_t + \frac{A}{\rho_0} p_x + \frac{\rho_0 c^2}{A} \left(\frac{q^2}{p}\right)_x = -\frac{\lambda \rho_0 c^2 |q| q}{2dAp}\,. \tag{1b}$$

Here, q denotes the flow rate under standard conditions (1 atm air pressure, temperature of 0 °C), p the pressure, c the speed of sound, λ the friction coefficient, d the diameter, A the cross-sectional area of the pipe and ρ_0 the density under standard conditions.

2.2.2 Semilinear Model

If the velocity of the gas is much smaller than the speed of sound, i.e., $|\mathsf{v}| \ll c$ with $\mathsf{v} = \frac{\rho_0 q}{\rho A}$, we can neglect the nonlinear term in the spatial derivative of the momentum equation in (1). This yields a semilinear model

$$p_t + \frac{\rho_0 c^2}{A} q_x = 0\,, \tag{2a}$$

$$q_t + \frac{A}{\rho_0} p_x = -\frac{\lambda \rho_0 c^2 |q| q}{2dAp}\,. \tag{2b}$$

2.2.3 Algebraic Model

A further simplification leads to the stationary model: Setting the time derivatives in (2) to zero results in an ordinary differential equation, which can be solved analytically:

$$q = \text{const.}\,, \tag{3a}$$

$$p(x) = \sqrt{p(x_0)^2 + \frac{\lambda \rho_0^2 c^2 |q| q}{dA^2}(x_0 - x)}\,. \tag{3b}$$

Here, $p(x_0)$ denotes the pressure at an arbitrary point $x_0 \in [x_j^a, x_j^b]$. Setting $x_0 = x_j^a$, that is the inbound of the pipe, and $x = x_j^b$, that is the end of the pipe, yields the algebraic model [15].

2.3 Further Network Components

Besides pipes, there are some other components a network can consist of. Those are, for example, compressor stations and valves. These components are, like the pipes, modelled as arcs. This way, the coupling conditions at the intersections are still valid. Flow rate and pressure are determined by algebraic equations that can be nonlinear.

Compressor Stations

A compressor station is a facility that increases the pressure of the gas. Running a compressor is relatively costly since the compressor station consumes some of the gas. The equation for the fuel consumption of a compressor is given by [11]

$$F(p_{in}, p_{out}, q_{in}) = c_F q_{in} \left(\left(\frac{p_{out}}{p_{in}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right), \tag{4}$$

the compressor power P is determined by

$$P(p_{in}, p_{out}, q_{in}) = c_P q_{in} \left(\left(\frac{p_{out}}{p_{in}} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right). \tag{5}$$

Here, γ is the isentropic coefficient of the gas, and c_F and c_P are compressor specific constants. The control of the compressor is given by the pressure difference.

Valves

Valves are used to regulate the flow of the gas by opening or closing. In case of an open valve, the equations $q_{in} = q_{out}$, $p_{in} = p_{out}$ hold. If the valve is closed, it is $q_{in} = q_{out} = 0$.

3 Error Estimators

With the different models for the pipes and the other network components, we can solve the whole network as a system using adequate initial, boundary and coupling conditions. A way to achieve a compromise between the accuracy of the solution and the computational costs is to use the more complex model only when necessary and to refine the discretization only where needed. Using the solution of adjoint equations as done in [6, 7, 3, 10], we deduce a model and a discretization error estimator to measure the influence of the model and the discretization on a user-defined target functional M.

The functional M can be of any form, for example measuring the pressure value troughout the network over the whole time interval,

$$M(p) = \int_0^T \int_\Omega p(x,t)\,\mathrm{d}x.$$

Another possibility is to measure the costs of the compressor stations in form of the fuel gas consumption,

$$M(p,q) = \sum_{c\in\mathcal{J}_c} \int_0^T F_c(t)\,\mathrm{d}t. \tag{6}$$

Let $\xi = (\xi_1,\xi_2)^T$ be the adjoint pressure and flow rate of one of the models with respect to the functional M. We now use the adjoint equations to assess the simplified models with respect to the quantity of interest. Let $u = (p,q)^T$ be the solution of the nonlinear model (1) and $u^h = (p^h,q^h)^T$ the solution of the discretized semilinear model (2). For the discretization of the PDE, we apply an implicit box scheme [12]. Then the difference between the target functional $M(u)$ and $M(u^h)$ can be approximated using Taylor expansion. Inserting the solution ξ of the adjoint system, we get a first order error estimator for the model and the discretization error respectively as in [3, 10] (see also [7]):

$$M(u) - M(u^h) \approx \eta_m + \eta_h \tag{7}$$

with the estimators η_m and η_h as follows:

$$\eta_m^{\text{LIN-NL}} = \int_0^T \int_\Omega -\xi^T \begin{pmatrix} 0 \\ \frac{\rho_0 c^2 (q^h)^2}{A p^h} \end{pmatrix}_x \mathrm{d}x\,\mathrm{d}t, \tag{8}$$

$$\eta_h^{\text{LIN}} = \int_0^T \int_\Omega -\xi^T \begin{pmatrix} p_t^h + \frac{\rho_0 c^2}{A} q_x^h \\ q_t^h + \frac{A}{\rho_0} p_x^h + \frac{\lambda \rho_0 c^2 |q^h| q^h}{2dAp^h} \end{pmatrix} \mathrm{d}x\,\mathrm{d}t. \tag{9}$$

If, the other way round, u denotes the solution of the semilinear model (2) and u^h the solution of the discretized nonlinear model, one gets the same estimator for the model error (except for the sign), and the discretization error reads as follows,

$$\eta_h^{\text{NL}} = \int_0^T \int_\Omega -\xi^T \begin{pmatrix} p_t^h + \frac{\rho_0 c^2}{A} q_x^h \\ q_t^h + \frac{A}{\rho_0} p_x^h + \frac{\rho_0 c^2}{A} \left(\frac{(q^h)^2}{p^h}\right)_x + \frac{\lambda \rho_0 c^2 |q^h| q^h}{2dAp^h} \end{pmatrix} \mathrm{d}x\,\mathrm{d}t. \tag{10}$$

Since the algebraic model can be solved exactly, the discretization error disappears and one only gets an estimator for the model error

$$\eta_m^{\text{ALG-LIN}} = \int_0^T \int_\Omega -\xi^T \begin{pmatrix} p \\ q \end{pmatrix}_t \mathrm{d}x\,\mathrm{d}t \tag{11}$$

with ξ being the solution of the adjoint equations either of the semilinear model or of the algebraic model. Here, $u = (p,q)^T$ denotes the solution of the stationary model (3).

The discretization error estimators η_h^{NL} and η_h^{LIN} may be split up into a temporal and a spatial discretization error estimator as follows. Let u be the exact and u^h be the solution of the discretized nonlinear model (1). We use a short notation of (1), i.e., $u_t + f(u)_x = g(u)$, which yields

$$\begin{aligned} \eta_h^{\text{NL}} &= \int_0^T \int_\Omega -\xi^T \left(u_t^h + f(u^h)_x - g(u^h)\right) \mathrm{d}x\,\mathrm{d}t \\ &= \int_0^T \int_\Omega -\xi^T \left((u_t^h - u_t) + (f(u^h)_x - f(u)_x) - (g(u^h) - g(u))\right) \mathrm{d}x\,\mathrm{d}t \end{aligned}$$

since u is the exact solution of (1). We may split the integral into two parts,

$$\eta_h^{\mathrm{NL}} = \underbrace{\int_0^T \int_\Omega -\xi^T (u_t^h - u_t)\,\mathrm{d}x\,\mathrm{d}t}_{=:\eta_t^{\mathrm{NL}}} + \underbrace{\int_0^T \int_\Omega -\xi^T \left((f(u^h)_x - f(u)_x) - (g(u^h) - g(u)) \right) \mathrm{d}x\,\mathrm{d}t}_{=:\eta_x^{\mathrm{NL}}}, \tag{12}$$

which gives the two estimators. The temporal and spatial discretization error estimators for the semi-linear model are derived analogously. For the computation, the exact solution is approximated by a higher order reconstruction using neighboring points. For the time derivative, we use a polynomial reconstruction of order 2 and denote it by $u_t \approx R_t(u^h)$. The spatial derivative of f and the value of g are reconstructed with order 4 (order 3 at the boundary), giving $f(u)_x \approx R_x(f(u^h))$ and $g(u) \approx R(g(u^h))$, respectively.

The estimators computed are then

$$\eta_t^{\mathrm{NL}} \approx \int_0^T \int_\Omega -\xi^T \left(u_t^h - R_t(u^h) \right) \mathrm{d}x\,\mathrm{d}t,$$
$$\eta_x^{\mathrm{NL}} \approx \int_0^T \int_\Omega -\xi^T \left(\left(f(u^h)_x - R_x(f(u^h)) \right) - \left(g(u^h) - R(g(u^h)) \right) \right) \mathrm{d}x\,\mathrm{d}t.$$

The detailed reconstruction formulas are given in Sect. 5.2.

4 Control of Model and Discretization Error

With the estimators defined above, we can introduce a strategy to control the model and discretization error inside the network. In practice, often smaller time horizons are optimized. Thus, we do not control the overall error any more, but the relative error piecewise. For this, we divide the time interval $[0, T]$ into blocks of equal size $[T_{k-1}, T_k]$, $k = 1, \ldots, N_B$. Regarding one subinterval $[T_{k-1}, T_k]$, we can compute the forward as well as the backward/adjoint solution and evaluate the error estimators, which yields

$$M_k(u) - M_k(u^h) \approx \eta_{m,k} + \eta_{t,k} + \eta_{x,k}.$$

Given a tolerance TOL for the relative error, we can approximate the exact error by the estimators giving

$$\frac{\left| M_k(u) - M_k(u^h) \right|}{\left| M_k(u) \right|} \approx \frac{\left| \eta_{m,k} + \eta_{t,k} + \eta_{x,k} \right|}{\left| M_k(u^h) \right|} \overset{!}{\leq} \mathtt{TOL}. \tag{13}$$

We first check the discretization error to ensure the discretization to be adequate. Then, the model error is taken into account. A scheme of the algorithm is given in Fig. 2.

Check discretization error

First, the discretization is checked. Given the tolerance TOL as above, we ensure the discretization error to be small enough by decreasing TOL by a user-defined factor $0 < \kappa < 1$ giving $\mathtt{TOL}_h := \kappa \cdot \mathtt{TOL}$. We demand the discretization error estimator to satisfy

$$\left| \eta_{t,k} + \eta_{x,k} \right| \leq \mathtt{TOL}_h \cdot \left| M_k(u^h) \right|.$$

If the error estimator exceeds the given upper bound, the temporal and spatial discretization errors are treated individually, that is,

$$\left| \eta_{t,k} \right| \leq \frac{1}{2} \mathtt{TOL}_h \cdot \left| M_k(u^h) \right| \text{ and } \left| \eta_{x,k} \right| \leq \frac{1}{2} \mathtt{TOL}_h \cdot \left| M_k(u^h) \right|.$$

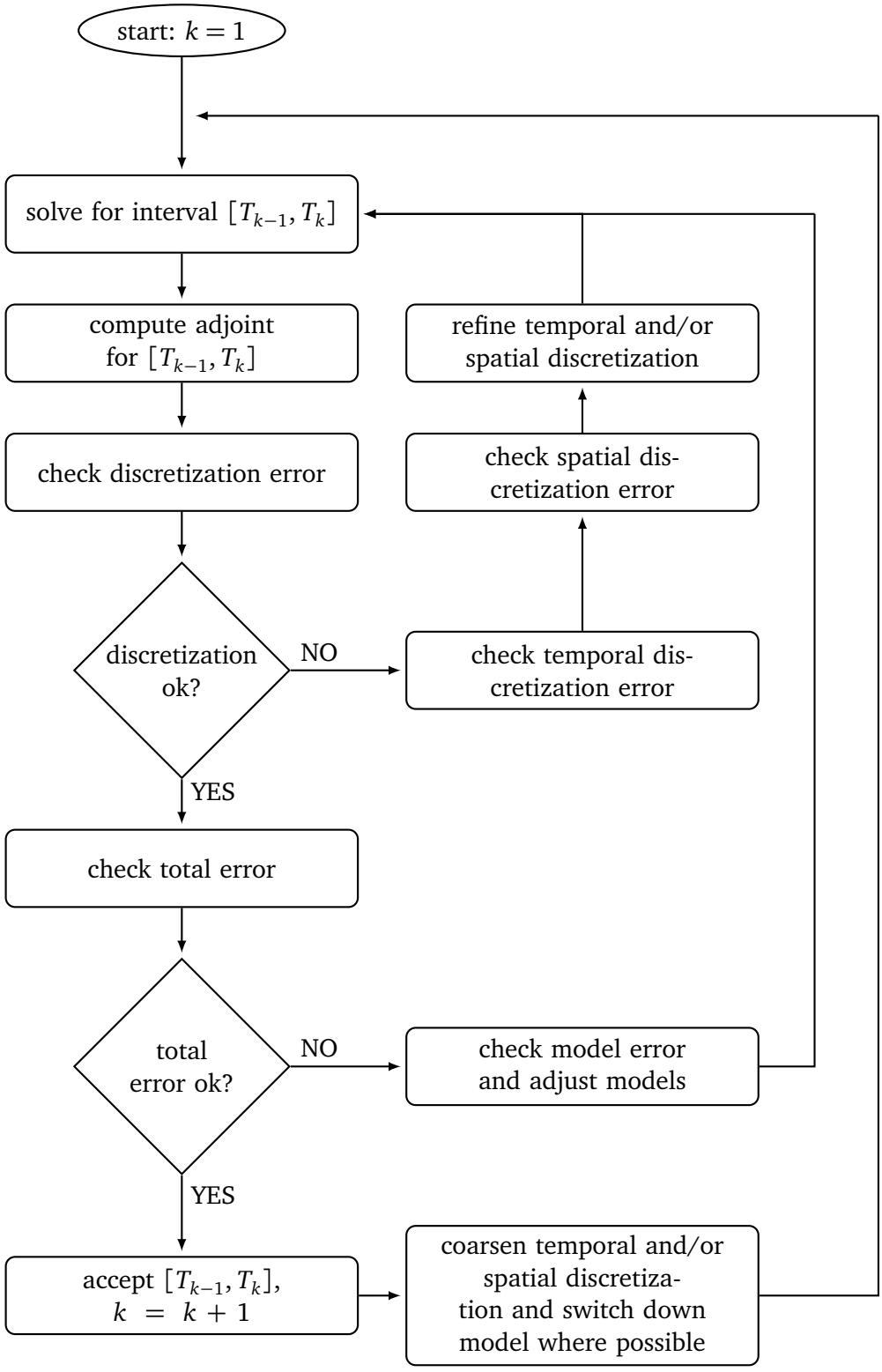

Figure 2: Scheme of the algorithm controlling discretization and model errors

Check Temporal Discretization Error

If the temporal error estimator exceeds the given tolerance, the time step size is marked for refinement. After checking the spatial discretization error, the time interval $[T_{k-1}, T_k]$ has to be computed again. If, in contrast, the error estimator $\left|\eta_{t,k}\right|$ is much smaller than the upper bound, the time step size is marked for coarsening. If the current time interval has to be recomputed due to spatial or model errors, the temporal coarsening is not applied.

Check Spatial Discretization Error

Now, the spatial discretization error is estimated locally for each pipe. We can split up the discretization error estimator from (12) for each pipe, giving for the nonlinear model

$$\eta^{\mathrm{NL}}_{x,k} = \int_{T_{k-1}}^{T_k} \sum_{j\in\mathcal{J}_p} \int_{x_j^a}^{x_j^b} -\xi^T \left((f(u^h)_x - f(u)_x) - (g(u^h) - g(u)) \right) \mathrm{d}x\,\mathrm{d}t = \sum_{j\in\mathcal{J}_p} \eta^{\mathrm{NL}}_{x,k,j}\,,$$

analogously for the other models. Thus, we can estimate the spatial discretization error for each pipe in the time interval $[T_{k-1}, T_k]$ with the corresponding model, which yields

$$\left| \sum_{j\in\mathcal{J}_p} \eta_{x,k,j} \right| \leq \frac{1}{2}\mathtt{TOL}_h \cdot \left| M_k(u^h) \right| .$$

For the inequality to hold, it suffices to claim

$$\sum_{j\in\mathcal{J}_p} \left| \eta_{x,k,j} \right| \leq \frac{1}{2}\mathtt{TOL}_h \cdot \left| M_k(u^h) \right| .$$

In order to get an upper bound for each pipe itself, we uniformly distribute the target functional, i.e., we divide it by the number of pipes, giving

$$\left| \eta_{x,k,j} \right| \leq \frac{1}{2}\mathtt{TOL}_h \cdot \frac{\left| M_k(u^h) \right|}{\left| \mathcal{J}_p \right|} \qquad \forall j \in \mathcal{J}_p\,.$$

If $\left|\eta_{x,k,j}\right|$ exceeds the given tolerance, the pipe is marked for refinement. If instead, the error estimator is much smaller than the right hand side, the pipe is marked for coarsening.

The time interval $[T_{k-1}, T_k]$ is computed again with a finer discretization where needed.

Check Total Error

If the discretization error is accepted, the total error estimator $\left|\eta_{m,k} + \eta_{t,k} + \eta_{x,k}\right|$ is evaluated. If

$$\left| \eta_{m,k} + \eta_{t,k} + \eta_{x,k} \right| > \mathtt{TOL} \cdot \left| M_k(u^h) \right| ,$$

that is, the total error is not ok while the discretization error was, the model error is checked.

Check Model Error

If the discretization error is small enough, but the total error is not, the model errors of all pipes are checked, i.e., the model error estimators (8) and/or (11) are evaluated for each pipe.

For the pipes using the semilinear or algebraic model, the estimators with respect to the higher models are evaluated. If the error estimator exceeds the given tolerance, that is,

$$\eta_{m,k,j} > \mathtt{TOL}_m \cdot \frac{M_k(u^h)}{\left| \mathcal{J}_p \right|}\,,$$

with $\mathtt{TOL}_m := (1-\kappa)\cdot\mathtt{TOL}$, the pipe is supposed to use the model above subject to the hierarchy. The time interval $[T_{k-1}, T_k]$ is computed again with the adjusted models.

Coarsen Temporal and/or Spatial Discretization and Switch Down Models

If the total error is ok, the time interval $[T_{k-1}, T_k]$ is accepted and k is increased.

If the time step size or any pipes were marked for coarsening, the coarsening is done. Then, the estimators with respect to the lower models are computed for the pipes using the nonlinear or the semilinear model. If the error estimator is much less than the given tolerance, that is,

$$\eta_{m,k,j} \leq s \cdot \mathrm{TOL}_m \cdot \frac{M_k(u^h)}{\left|\mathcal{J}_p\right|},$$

with a "shift down factor" $s \ll 1$ (e.g. 10^{-1} or 10^{-2}), the pipe can use the lower model for the next calculations and we go on to the next interval.

5 Implementation of the Error Estimators

The error estimators introduced in the previous sections are implemented using a *first-discretize* approach. This approach does not introduce additional discretization errors for the computation of the adjoint variables and can be efficiently implemented for the applied discretization scheme.

5.1 Model Error

According to the applied time steps t_j $(j = 0, \ldots, N)$, we split up the solution of the discretized model equations

$$(u^h)^T = \left((u_0^h)^T, \ldots, (u_N^h)^T\right).$$

In each time step, a system of the form

$$F_j(\underbrace{u_{j-1}^h}_{u_{old}}, \underbrace{u_j^h}_{u_{new}}) = 0 \tag{14}$$

is solved. Due to the blockwise decomposition of the simulation horizon in our algorithm, we define

$$(U_k^h)^T = \left((u_{j(k-1)+1}^h)^T, \ldots, (u_{j(k)}^h)^T\right)$$

for $k = 1, \ldots, N_B$, where $j(k)$ is given via

$$t_{j(k)} = T_k$$

for $k = 0, \ldots, N_B$. For later use, we also define

$$(E_k)^T = \left((F_{j(k-1)+1})^T, \ldots, (F_{j(k)})^T\right), \tag{15}$$

which summarizes the state-defining equations of the block $[T_{k-1}, T_k]$.

Our blockwise model error estimators with respect to the functional M are of the form

$$\eta_{m,k} = \frac{\partial}{\partial U_k} M(u^h) \Delta U_k^h \tag{16}$$

with

$$\Delta U_k^h = U_k - U_k^h.$$

Here, U_k formally denotes the reference solution in the kth block, which solves a different system of equations $\tilde{E}_k$ based on more complex or simpler models. The difference ΔU_k^h results from the differences in the models and is estimated by

$$\Delta U_k^h \approx -\big(\frac{\partial}{\partial U_k}\tilde{E}_k(u^h)\big)^{-1}\Delta\tilde{E}_k \tag{17}$$

with

$$\Delta\tilde{E}_k = \tilde{E}_k(U_k^h) - \underbrace{\tilde{E}_k(U_k)}_{=0} = \tilde{E}_k(U_k^h). \tag{18}$$

Inserting (17) and (18) in (16) gives

$$\eta_{m,k} = -\frac{\partial}{\partial U_k}M(u^h)\big(\frac{\partial}{\partial U_k}\tilde{E}_k(u^h)\big)^{-1}\tilde{E}_k(U_k^h) = -\xi_k^T\tilde{E}_k(U_k^h)$$

with ξ_k being the solution of the adjoint equation

$$\big(\frac{\partial}{\partial U_k}\tilde{E}_k(u^h)\big)^T\xi_k = \big(\frac{\partial}{\partial U_k}M(u^h)\big)^T. \tag{19}$$

Instead of $\frac{\partial}{\partial U_k}\tilde{E}_k(u^h)$ one may also apply $\frac{\partial}{\partial U_k}E_k(u^h)$ in (19), which corresponds to the choice of which model to use for the adjoint variables in Sect. 3. For performance reasons, we only compute one adjoint ξ_k for the error estimators with respect to more complex models and simpler models, and also for the discretization error estimators. Therefore, we use the current model(s) or shift up all models to the next higher model for assembling the adjoint equations, which is implemented by using a flag `useUpperAdjoint`.

Note that (19) can be solved very efficiently due to the special structure of $\tilde{E}_k$ and E_k. With (14) and (15) we get

$$\frac{\partial}{\partial U_k}E_k(u^h) = \begin{pmatrix} B_{j(k-1)+1} & & & & \\ A_{j(k-1)+2} & B_{j(k-1)+2} & & & \\ & A_{j(k-1)+3} & B_{j(k-1)+3} & & \\ & & \ddots & \ddots & \\ & & & A_{j(k)} & B_{j(k)} \end{pmatrix}$$

with

$$A_m = \frac{\partial}{\partial u_{old}}F_m(u_{m-1}^h, u_m^h),$$
$$B_m = \frac{\partial}{\partial u_{new}}F_m(u_{m-1}^h, u_m^h).$$

This yields for the adjoint system (19) (with E_k instead of $\tilde{E}_k$):

$$\begin{pmatrix} B^T_{j(k-1)+1} & A^T_{j(k-1)+2} & & & \\ & B^T_{j(k-1)+2} & A^T_{j(k-1)+3} & & \\ & & B^T_{j(k-1)+3} & \ddots & \\ & & & \ddots & A^T_{j(k)} \\ & & & & B^T_{j(k)} \end{pmatrix} \begin{pmatrix} \xi_{k,1} \\ \xi_{k,2} \\ \vdots \\ \vdots \\ \xi_{k,N_k} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial u_{j(k-1)+1}} M(u^h) \\ \frac{\partial}{\partial u_{j(k-1)+2}} M(u^h) \\ \vdots \\ \vdots \\ \frac{\partial}{\partial u_{j(k)}} M(u^h) \end{pmatrix}.$$

Obviously, the blockwise bidiagonal structure of the matrix $\frac{\partial}{\partial U_k} E_k(u^h)$ allows to solve this system blockwise backwards in time, reducing the size of the systems to be solved:

$$\begin{aligned} \xi_{k,N_k} &= B^{-T}_{j(k)} \frac{\partial}{\partial u_{j(k)}} M(u^h), \\ \xi_{k,N_k-1} &= B^{-T}_{j(k)-1} \frac{\partial}{\partial u_{j(k)-1}} M(u^h) - A^T_{j(k)} \xi_{k,N_k}, \\ &\dots \\ \xi_{k,1} &= B^{-T}_{j(k-1)+1} \frac{\partial}{\partial u_{j(k-1)+1}} M(u^h) - A^T_{j(k-1)+2} \xi_{k,2}. \end{aligned}$$

Moreover, the single systems are sparse and allow for very efficient solvers, e.g. [8].

5.2 Discretization Error

In this section, we will give the implemented reconstruction formulas of higher order that we use for the computation of the discretization error estimators. As described in Sect. 3, the difference between these reconstructions and the discrete approximations of the applied scheme has to be multiplied with the adjoint variable to get the corresponding error estimator. The same holds for the "first-discretize" approach.

Time Derivative

The implicit box scheme we use is of first order in time. Hence it suffices to reconstruct the temporal derivative with order 2. Let u_j be the solution at time $t_j \in [T_{k-1}, T_k]$, $j = j(k-1), \dots, j(k)$, with time step size τ. To yield an order 2 reconstruction, we need three values of u. For all except the last step, we simply use a central difference quotient

$$R_t(u^h_j) = \frac{1}{2\tau} \left(u^h_{j+1} - u^h_{j-1} \right) + \mathcal{O}(\tau^2)$$

for all x.

For the last time step in block $[T_{k-1}, T_k]$, we may use a backward differentiation formula of 2nd order, which needs two old values $u_{j(k)-1}$ and $u_{j(k)-2}$:

$$R_t(u^h_{j(k)}) = \frac{1}{2\tau} \left(u^h_{j(k)-2} - 4u^h_{j(k)-1} + 3u^h_{j(k)} \right) + \mathcal{O}(\tau^2).$$

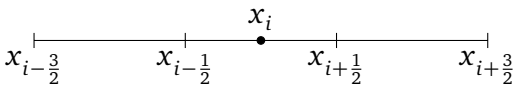

Figure 3: Reconstruct $u_x(x_i, t_j)$ using only values at $x_{i-\frac{3}{2}}$, $x_{i-\frac{1}{2}}$, $x_{i+\frac{1}{2}}$ and $x_{i+\frac{3}{2}}$

Spatial Derivative

For the spatial derivative, the reconstruction is a little more complicated. We use an implicit box scheme that evaluates the PDE in the center of a cell. Thus, we want to reconstruct the spatial derivative also in the center of the cell. Since the box scheme is of 2nd order in space, we need to have at least order 3. To achieve this, we use four grid points as indicated in Fig. 3.

We denote the spatial step size by h. For an arbitrary time step t_j, we have

$$R_x(u^h_{i,j}) = \frac{1}{24h}\left(u^h_{i-\frac{3}{2},j} - 27u^h_{i-\frac{1}{2},j} + 27u^h_{i+\frac{1}{2},j} - u^h_{i+\frac{3}{2},j}\right) + \mathcal{O}(h^4).$$

At the boundary, we cannot use the central reconstruction as above. Using the points $x_{i-\frac{1}{2}}$ to $x_{i+\frac{5}{2}}$ for the reconstruction of $u_x(x_i, t_j)$ at the left boundary as shown in Fig. 4, we still gain 3rd order with

$$R_x(u^h_{i,j}) = \frac{1}{24h}\left(-23u^h_{i-\frac{1}{2},j} + 21u^h_{i+\frac{1}{2},j} + 3u^h_{i+\frac{3}{2},j} - u^h_{i+\frac{5}{2},j}\right) + \mathcal{O}(h^3).$$

For the right boundary, the formula is simply reversed.

Figure 4: Reconstruct $u_x(x_i, t_j)$ using only values at $x_{i-\frac{1}{2}}$, $x_{i+\frac{1}{2}}$, $x_{i+\frac{3}{2}}$ and $x_{i+\frac{5}{2}}$

Reconstruction of the Function Value

Since we already use 4 points for the reconstruction of the spatial derivative, we use the same points for the reconstruction of the function value to gain at least 3rd order. The reconstructed values are then

$$R(u^h_{i,j}) = \frac{1}{16}\left(-u^h_{i-\frac{3}{2},j} + 9u^h_{i-\frac{1}{2},j} + 9u^h_{i+\frac{1}{2},j} - u^h_{i+\frac{3}{2},j}\right) + \mathcal{O}(h^4)$$

for the inner points and

$$R(u^h_{i,j}) = \frac{1}{16}\left(5u^h_{i-\frac{1}{2},j} + 15u^h_{i+\frac{1}{2},j} - 5u^h_{i+\frac{3}{2},j} + u^h_{i+\frac{5}{2},j}\right) + \mathcal{O}(h^4)$$

for the left boundary. Again, the formula for the right boundary is just reversed.

6 Numerical Results

In this section, we give numerical results for a medium sized real life network. The network consists of 12 pipes (P01 – P10, with lengths between 30km and 100km), 2 sources (S01 – S02), 4 consumers (C01 – C04), 3 compressor stations (Comp01 - Comp03) and one control valve (CV01). The graph of the network is shown in Fig. 5.

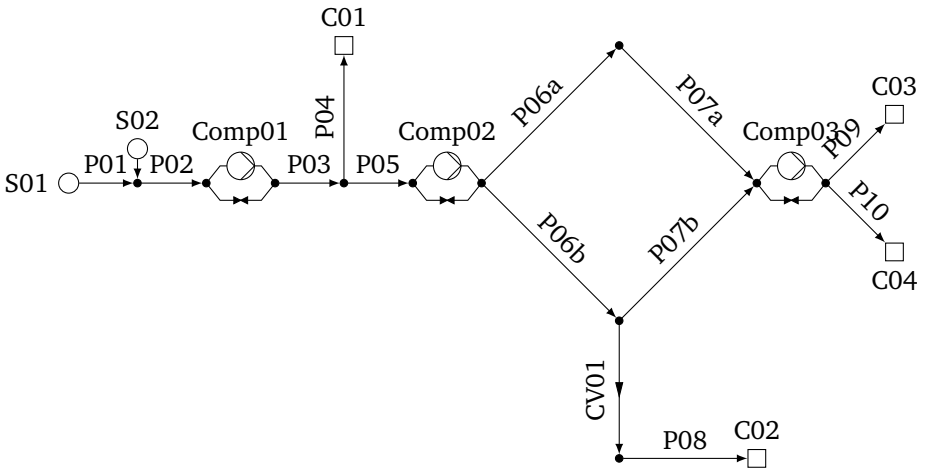

Figure 5: Network with compressor stations and control valve

The simulation starts with stationary initial data. The boundary conditions and the control for the compressor stations are time-dependent. The target functional is given by the total fuel gas consumption of the compressors, i.e.

$$M(u) = \sum_{c \in \mathcal{J}_c} \int_0^T F_c(t)\,\mathrm{d}t\,.$$

The simulation time is 86400 seconds (24 hours) with an initial time step size $\Delta t = 1800$ seconds. The subintervals are 7200 seconds (2 hours) each. The initial spatial step size is $\Delta x = 10000$m. The factor κ is set to 10^{-1} and the shift down factor $s = 10^{-1}$.

For the computations, we varied three different parameters. First, the tolerance `TOL` is set to values between 10^{-1} and 10^{-4}. Second, whether we use the adjoint of the current or the next higher model for the computation of the error estimators (`useUpperAdjoint = false/true`). Third, we apply two different implementations of the stationary model: One option is to use the algebraic equation as introduced in Sect. 2. Alternatively, one may apply the box scheme to the stationary equations (semilinear model without temporal derivatives). This way, we give up the advantage of knowing the analytical solution and we expect to increase the computing time. This option is based on the following idea: Since the discretization error equals zero for the algebraic model, a shift-up to the semilinear model will decrease the model error but increase the discretization error. Applying the box scheme to the stationary equations reduces this effect and the algorithm focuses on the model error. In the implementation, this option is set via the flag `balanceLawForStationaryModel = true/false`.

Tables 1(a) to 1(d) show the maximal relative error in the target functional

$$\text{rel.err.} = \max_k \frac{\left|M_k(u) - M_k(u^h)\right|}{\left|M_k(u)\right|}\,,$$

the total target functional, the maximal and the minimal time and spatial step size used and the total time for the computation subject to the tolerance `TOL`. The flag `useUpperAdjoint` is either set to `true` or `false`, so is `balanceLawForStationaryModel`. As an approximation of the exact solution we computed a solution with the nonlinear model and a finer discretization than used in the adaptive algorithm, which is shown in the last row. All computations were done on a 3100MHz AMD Athlon™ 64 X2 Dual Core 6000+.

As expected, the computing time using the option `balanceLawForStationaryModel = true` is larger if the algebraic model is used often (`TOL` $= 10^{-1}$ and 10^{-2}). The accuracy of the solution is comparable to that with `balanceLawForStationaryModel = false`.

If we use the adjoint of the higher model, the error estimators seem to be more precise: the actual relative errors are closer to the given tolerances. The computing times are similar. If we used the adjoint of the current model instead, we observed that the algorithm often did multiple down- and up-shifts of the models. This leads us to conjecture that this error estimator is not as reliable as using the adjoint of the higher model.

Let us mention that with all configurations, the error bounds were maintained.

Besides the discretization, it is also interesting how the model switching part works depending on `TOL`. Tables 2(a) to 2(d) show how often which model is used during the simulation. Generally, we observe that all options show the same trend. For smaller tolerances, the share of the more complex models is bigger. If `useUpperAdjoint` is set to `false`, the semilinear model is already used at tolerance $\texttt{TOL} = 10^{-1}$. Also, the fraction of the more complex models is higher compared to `useUpperAdjoint = true`.

Again, we conjecture that `useUpperAdjoint = true` leads to more accurate error estimators.

Table 1: Results using different parameters and values for `TOL`

(a) `useUpperAdjoint` = true, `balanceLawForStationaryModel` = false

TOL	rel.err.	$M(u^h)$	max/min Δt	max/min Δx	time [s]
1e-01	5.8781e-02	85.567278	3600/1800	20,000/10,000	1.4e-01
1e-02	1.0641e-03	86.219122	3600/900	20,000/5,000	4.5e-01
1e-03	9.1421e-04	86.213310	3600/56.25	20,000/2,500	6.3e+00
1e-04	8.0166e-05	86.218315	900/7.03125	20,000/625	1.3e+02
reference solution		86.217296	3.515625	312.5	9.0e+02

(b) `useUpperAdjoint` = true, `balanceLawForStationaryModel` = true

TOL	rel.err.	$M(u^h)$	max/min Δt	max/min Δx	time [s]
1e-01	5.8773e-02	85.568162	3600/1800	20,000/10,000	2.0e-01
1e-02	1.0588e-03	86.219352	3600/900	20,000/5,000	5.6e-01
1e-03	9.1286e-04	86.213442	3600/56.25	20,000/2,500	6.3e+00
1e-04	8.0176e-05	86.218269	900/7.03125	10,000/625	1.3e+02
reference solution		86.217296	3.515625	312.5	9.0e+02

(c) `useUpperAdjoint` = false, `balanceLawForStationaryModel` = false

TOL	rel.err.	$M(u^h)$	max/min Δt	max/min Δx	time [s]
1e-01	3.2508e-03	86.225223	3600/1800	20,000/10,000	1.2e-01
1e-02	9.1421e-04	86.216718	3600/450	20,000/5,000	4.0e-01
1e-03	1.8630e-04	86.220416	3600/56.25	20,000/2,500	5.5e+00
1e-04	2.3310e-05	86.217566	450/7.03125	20,000/625	1.3e+02
reference solution		86.217296	3.515625	312.5	9.0e+02

(d) `useUpperAdjoint` = false, `balanceLawForStationaryModel` = true

TOL	rel.err.	$M(u^h)$	max/min Δt	max/min Δx	time [s]
1e-01	3.2237e-03	86.225280	3600/1800	20,000/10,000	2.6e-01
1e-02	9.0886e-04	86.216088	3600/450	20,000/5,000	5.6e-01
1e-03	1.8629e-04	86.220465	3600/56.25	20,000/2,500	6.1e-01
1e-04	2.3348e-05	86.217569	450/7.03125	10,000/625	1.3e+02
reference solution		86.217296	3.515625	312.5	9.0e+02

Table 2: Models used during simulation for different parameters and values for TOL

(a) useUpperAdjoint = true, balanceLawForStationaryModel = false

TOL	ALG	LIN	NL
1e-01	100%	0%	0%
1e-02	81.9%	18.1%	0%
1e-03	68.7%	31.3%	0%
1e-04	38.2%	36.8%	25.0%

(b) useUpperAdjoint = true, balanceLawForStationaryModel = true

TOL	ALG	LIN	NL
1e-01	100%	0%	0%
1e-02	81.9%	18.1%	0%
1e-03	68.7%	31.3%	0%
1e-04	38.2%	36.8%	25.0%

(c) useUpperAdjoint = false, balanceLawForStationaryModel = false

TOL	ALG	LIN	NL
1e-01	90.3%	9.7%	0%
1e-02	79.2%	20.8%	0%
1e-03	56.2%	43.8%	0%
1e-04	31.2%	43.8%	25.0%

(d) useUpperAdjoint = false, balanceLawForStationaryModel = true

TOL	ALG	LIN	NL
1e-01	90.3%	9.7%	0%
1e-02	79.9%	20.1%	0%
1e-03	56.2%	43.8%	0%
1e-04	31.2%	43.8%	25.0%

Figure 6 shows a snapshot of the simulation at time $t = 32,400$s with TOL $= 10^{-4}$. At the sources and sinks, the upper numbers denote the pressure in bar, the lower ones denote the flow rate. At the compressor stations, the upper numbers are the increase in pressure and also the flow rate is shown below. At all inner nodes, only the pressure is printed. At each pipe, also the used model is indicated. The small white dots in the thick black lines represent the discretization. Note that the picture is not true to scale.

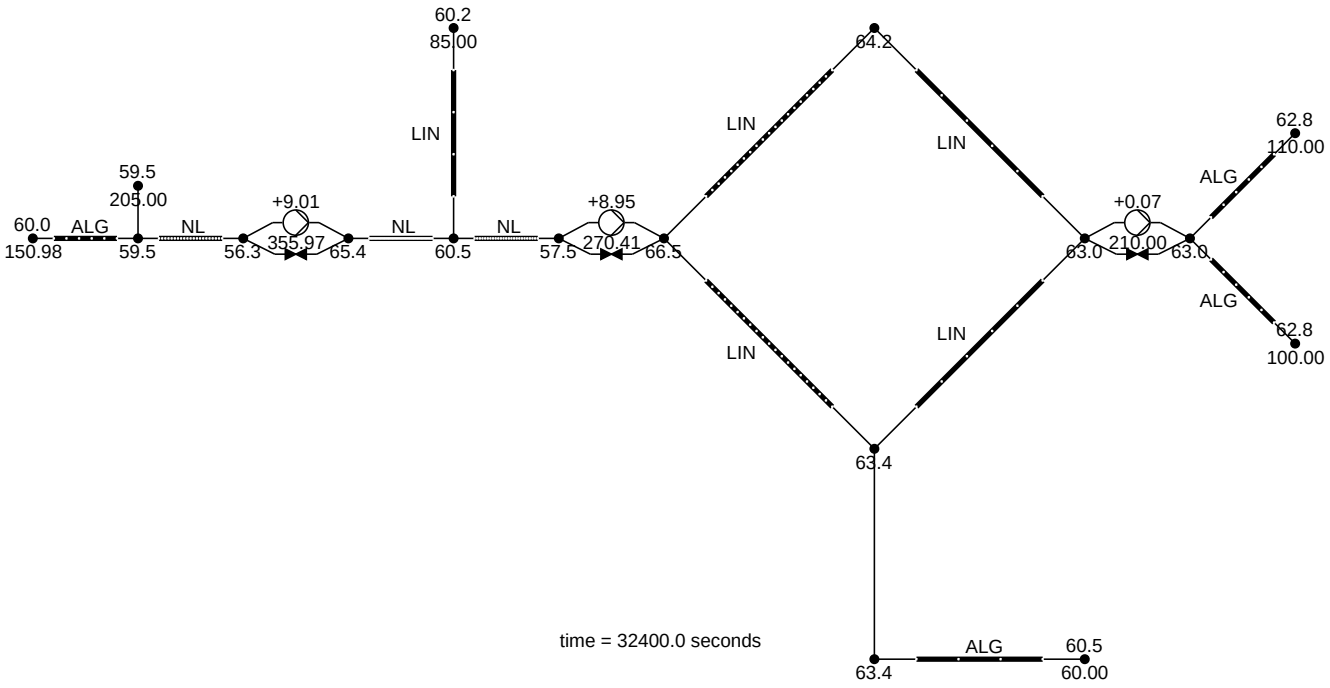

Figure 6: Snapshot of the network at time $t = 32,400$ s

7 Summary

We presented an algorithm that adaptively controls model and discretization errors with respect to a given target functional for gas flow in networks. For that, we used a hierarchy of models that describe the flow of gas through a pipe qualitatively different. Using adjoint techniques, we introduced error estimators for the model errors as well as for the discretization errors. With these estimators we developed an algorithm that controls the model and discretization errors subject to a given tolerance. Therefore, it automatically switches between the models in the hierarchy in both directions and refines or coarsens the discretizations. In the numerical results, it could be seen that for different tolerances, the discretization was adaptively adjusted and also the different models were used. Throughout the examples, the error estimators making use of the adjoint variables of the next higher model delivered the most accurate and therefore reliable results.

Regarding the computing time of the adaptive algorithm compared with the reference solution, the benefits of the adaptive approach become obvious. Clearly, the adaptive algorithm also delivers reliable information about the accuracy of the discrete solution. The prescribed error bounds were maintained in all examples.

Acknowledgements

This paper was supported by the German Research Foundation (DFG) under the grant LA1372/5-1.

References

[1] SIMONE software, `http://www.simone.eu`.

[2] Pia Bales. Hierarchische Modellierung der Eulerschen Flussgleichungen in der Gasdynamik. Diploma thesis, Technische Universität Darmstadt, Department of Mathematics, Darmstadt, 2005.

[3] Pia Bales, Oliver Kolb, and Jens Lang. Hierarchical modelling and model adaptivity for gas flow on networks. In *Computational Science - ICCS 2009*, volume 5544 of *Lecture Notes in Computer Science*, pages 337–346. Springer Berlin / Heidelberg, 2009.

[4] Mapundi Banda and Michael Herty. Multiscale modeling for gas flow in pipe networks. *Math. Meth. Appl. Sci.*, 31:915–936, August 2008.

[5] Mapundi Banda, Michael Herty, and Axel Klar. Coupling conditions for gas networks governed by the isothermal Euler equations. *Networks and Heterogeneous Media*, 1(2):295–314, June 2006.

[6] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta numerica*, 10:1–102, 2001.

[7] Malte Braack and Alexandre Ern. A posteriori control of modeling errors and discretization errors. *SIAM Journal of Multiscale Modeling and Simulation*, 1(2):221–238, 2003.

[8] Timothy A. Davis. *Direct methods for sparse linear systems (fundamentals of algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[9] Pia Domschke, Björn Geißler, Oliver Kolb, Jens Lang, Alexander Martin, and Antonio Morsi. Combination of nonlinear and linear optimization of transient gas networks. *To appear in INFORMS Journal on Computing*, 2010.

[10] Pia Domschke, Oliver Kolb, and Jens Lang. An adaptive model switching and discretization algorithm for gas flow on networks. *Procedia Computer Science*, 1(1):1325 – 1334, 2010. ICCS 2010.

[11] Michael Herty. Modeling, simulation and optimization of gas networks with compressors. *Networks and Heterogeneous Media*, 2(1):81–97, 2007.

[12] Oliver Kolb, Jens Lang, and Pia Bales. An implicit box scheme for subsonic compressible flow with dissipative source term. *Numerical Algorithms*, 53(2):293–307, 2010.

[13] Alexander Martin, Markus Möller, and Susanne Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical Programming*, 105:563–582, 2006.

[14] Susanne Moritz. *A mixed integer approach for the transient case of gas network optimization*. PhD thesis, TU Darmstadt, 2006.

[15] Erwin Sekirnjak. Transiente Technische Optimierung (TTO-Prototyp). Technical report, PSI AG, November 2000.