



TECHNISCHE UNIVERSITÄT BERLIN

**Adaptive Refinement Strategies for the
Simulation of Gas Flow in Networks
using a Model Hierarchy**

**Pia Domschke, Aseem Dua, Jeroen J. Stolwijk, Jens Lang,
Volker Mehrmann**

Preprint 2017/03

Preprint-Reihe des Instituts für Mathematik

Technische Universität Berlin

<http://www.math.tu-berlin.de/preprints>

Preprint 2017/03

Januar 2017

Adaptive Refinement Strategies for the Simulation of Gas Flow in Networks using a Model Hierarchy

Pia Domschke^a, Aseem Dua^c, Jeroen J. Stolwijk^{c,*}, Jens Lang^{a,b}, Volker Mehrmann^c

^a*Department of Mathematics, TU Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany*

^b*Graduate School of Energy Science and Engineering, TU Darmstadt, Jovanka-Bontschits-Str. 2, 64287 Darmstadt, Germany*

^c*Institut für Mathematik, MA 4-5, TU Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany*

Abstract

A model hierarchy that is based on the one-dimensional isothermal Euler equations of fluid dynamics is used for the simulation and optimisation of gas flow through a pipeline network. Adaptive refinement strategies have the aim of bringing the simulation error below a prescribed tolerance while keeping the computational costs low. While spatial and temporal stepsize adaptivity is well studied in the literature, model adaptivity is a new field of research. The problem of finding an optimal refinement strategy that combines these three types of adaptivity is a generalisation of the unbounded knapsack problem. A refinement strategy that is currently used in gas flow simulation software is compared to two novel greedy-like strategies. Both a theoretical experiment and a realistic gas flow simulation show that the novel strategies significantly outperform the current refinement strategy with respect to the computational cost incurred.

Keywords: gas supply networks, model hierarchy, error estimators, model adaptivity, refinement strategies

2010 MSC: 65K99, 65Z99, 65M22, 35Q31

*Corresponding author

Email addresses: domschke@mathematik.tu-darmstadt.de (Pia Domschke), dua.aseem@gmail.com (Aseem Dua), stolwijk@math.tu-berlin.de (Jeroen J. Stolwijk), lang@mathematik.tu-darmstadt.de (Jens Lang), mehrmann@math.tu-berlin.de (Volker Mehrmann)

1. Introduction

The simulation of gas flows in pipeline networks is a topic of research that has been studied at various scales: from the individual pipeline to the entire network. Studies in control and optimisation of gas supply in a dynamic supply-demand environment strongly depend on large scale simulations of pipeline networks. In the last decades, considerable research on the modelling, simulation and optimisation of gas flow through pipeline networks has been conducted, see e.g. [1–13]. Depending upon requirement, there exist multiple models to predict the system behaviour with varying levels of accuracy. Generally, more accurate models are computationally more expensive. Hence, in order to make real-time decisions, an appropriate trade-off between accuracy and computational complexity should be made. This can be achieved by using a hierarchy of models, where the models can be adaptively switched during the simulation process. Beside the models, the discretisation mesh may be varied in space and time, which places the demand for a strategy to automatically steer the simulation by changing the models and the discretisation meshes. This steering is based on simulation error estimates, which have been studied in detail in [14]. Since the simulations are the basis for decisions in the optimisation and control of the gas flow, the reliability of the simulation is of prime importance. The simulation is to be carried out such that the relative error in the state or in a functional of interest is below a specified tolerance. Starting with a coarse simulation, an adaptive strategy is used to bring the error below the tolerance by refining the discretisation in time and space or refining models, i.e., shifting to a model of higher accuracy. Hence, we have three different refinement possibilities for each pipe $j \in \mathcal{J}_p$ of the pipeline network, where \mathcal{J}_p denotes the set of pipes in the network. These refinement possibilities are indexed by $i = 1, \dots, 3N_p$, where $N_p := |\mathcal{J}_p|$ is the number of pipes. Refinements are to be chosen such that the computational costs are kept low. We define an optimal refinement strategy as a strategy which returns the solution of the constrained optimisation problem

$$\begin{aligned}
\min_{r_i} \quad & c + \sum_{i=1}^{3N_p} \sum_{k=1}^{r_i} v_{ik} \\
\text{s.t.} \quad & \eta - \sum_{i=1}^{3N_p} \sum_{k=1}^{r_i} w_{ik} \leq \text{tol}_\eta.
\end{aligned} \tag{1}$$

Here, for each refinement possibility i , r_i is the number of refinements, w_{ik} is the relative error reduction due to the k^{th} subsequent refinement and v_{ik} is the corresponding cost addition. The constants c and η denote the cost and relative error of the starting simulation, respectively. We note that if v_{ik} and w_{ik} are constant for all k , then this problem is equivalent to the unbounded knapsack problem which is NP-hard, see e.g. [15]. In this paper we aim to find a good approximation to the solution of this generalisation of the knapsack problem (1). For this, we propose three adaptive refinement strategies which return approximate solutions of (1). The ideas presented here are for the example of pipeline networks. By generalising pipes to functional sub-domains, the principles of adaptive refinement can be extended to simulations for other applications which use a model hierarchy, e.g., power grids and water supply networks.

This paper is organised as follows. Section 2 introduces a model hierarchy for the simulation of gas flow through a single pipe as well as the aim of the adaptive refinement strategies. Section 3 describes the three proposed refinement strategies and Section 4 introduces the design of a synthetic experiment. The results of this synthetic experiment are contained in Section 5 and an application of the refinement strategies to a realistic gas network simulation is given in Section 6. Finally, some conclusions are contained in Section 7.

2. Simulation of Gas Flow

In this section, we give an example of a model hierarchy for gas flow simulations. We then outline a framework for the adaptive simulation of pipeline networks using the given model hierarchy. Finally, we highlight the aims of the refinement strategies in the simulation framework.

2.1. The Model Hierarchy

As an example, we take a three-model hierarchy for the gas flow simulations, discussed in detail in [16]. At the highest level, we have the *isothermal Euler equations* relating the gas density ρ and gas flow ρv by

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho v) &= 0, \\ \frac{\partial}{\partial t} (\rho v) + \frac{\partial}{\partial x} (p + \rho v^2) &= -\frac{\lambda}{2D} \rho v |v| - g \rho h', \end{aligned} \tag{M1}$$

together with the equation of state for real gases $p = \rho z(p) R T$ with compressibility factor $z(p) = 1 - \alpha p$ for a constant $\alpha \in \mathbb{R}^+$. Here, p denotes the pressure, v the velocity of the gas, R the specific gas constant, and T the temperature. Further, $\lambda > 0$ is the Darcy friction coefficient, D the pipe diameter, $g = 9.81 \text{ m/s}^2$ the acceleration due to gravity, and h' the slope of the pipeline. If the term $\frac{\partial}{\partial x} (\rho v^2)$ is small, it can be dropped resulting in a *semilinear model*

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} (\rho v) &= 0, \\ \frac{\partial}{\partial t} (\rho v) + \frac{\partial p}{\partial x} &= -\frac{\lambda}{2D} \rho v |v| - g \rho h'. \end{aligned} \tag{M2}$$

A further simplification of assuming a stationary state and zero slope $h' = 0$ yields a system of two ODEs, which can be solved analytically and are referred to as the *algebraic model*

$$\begin{aligned} \rho v &= \text{constant}, \\ p(x) &= \sqrt{p_{\text{in}}^2 - \frac{\lambda c^2 x}{D} \rho v |\rho v|}. \end{aligned} \tag{M3}$$

Here, $c = \sqrt{p/\rho}$ denotes the speed of sound within the gas and $p_{\text{in}} = p(0)$ the input pressure. The three models are shown in hierarchical form in Fig. 1. The model hierarchy is set in the decreasing order of accuracy for our purpose of gas flows. Each pipe in the network is simulated using one of these three models and varying discretisation stepsizes in space and time.

2.2. Adaptive Gas Network Simulation

We consider a gas flow simulation over a pipeline network \mathcal{J}_p . The simulation time $[0, T]$ is divided into time intervals of equal size $[t_k, t_{k+1}]$, $k = 0, 1, \dots, N-1$

and $t_N = T$. Given a starting model distribution over the network $\mathbf{m}_0 = [m_{0,1}, m_{0,2}, \dots, m_{0,N_p}]^T$, with $m_{0,j} \in \{1, 2, 3\}$, and a corresponding discretisation in space $\mathbf{n}_{x,0}$ and in time $\mathbf{n}_{t,0}$, a simulation is run for $[t_0, t_1]$. We obtain error distributions along the network using *a posteriori* error estimations (see [14]): $\mathbf{e}_m = [e_{m,1}, e_{m,2}, \dots, e_{m,N_p}]^T$ for the model errors and $\mathbf{e}_x = [e_{x,1}, e_{x,2}, \dots, e_{x,N_p}]^T$ and $\mathbf{e}_t = [e_{t,1}, e_{t,2}, \dots, e_{t,N_p}]^T$ for the spatial and temporal discretisation errors, respectively. These error estimators are derived using a dual weighted residual method with a user-defined functional of interest $M(\mathbf{u})$, where $\mathbf{u} = [\rho, \rho v]^T$, see [14]. The simulation error for a single pipe is the sum of all three errors. For the simulation to be valid, the relative error must be below a given tolerance tol . Hence, we require that

$$\frac{\sum_{j \in \mathcal{J}_p} (e_{m,j} + e_{x,j} + e_{t,j})}{|M(\mathbf{u}^h)|} < \text{tol}. \quad (2)$$

If the tolerance is not achieved, models and discretisation meshes are refined. The task of deciding the required refinements is made by an adaptive strategy. A switch to a higher model in the hierarchy is called a *model refinement* and a refinement of the mesh is called a *discretisation refinement*. With the new models and discretisations we re-simulate for the time interval and continue the cycle. Once the solution meets the tolerance requirements, the models and

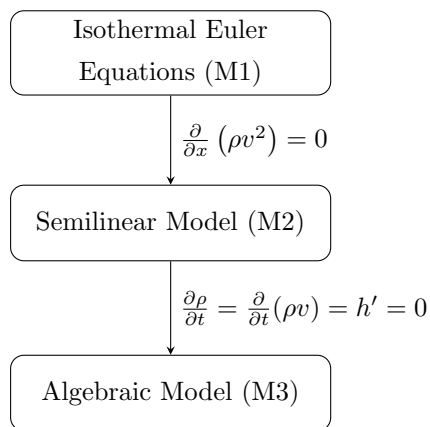


Figure 1: The model hierarchy that is considered in this paper.

55 discretisations are coarsened where possible, the simulation progresses to the next time interval and the cycle repeats. This simulation flow is shown in Figure 2 for an interval $[t_k, t_{k+1}]$.

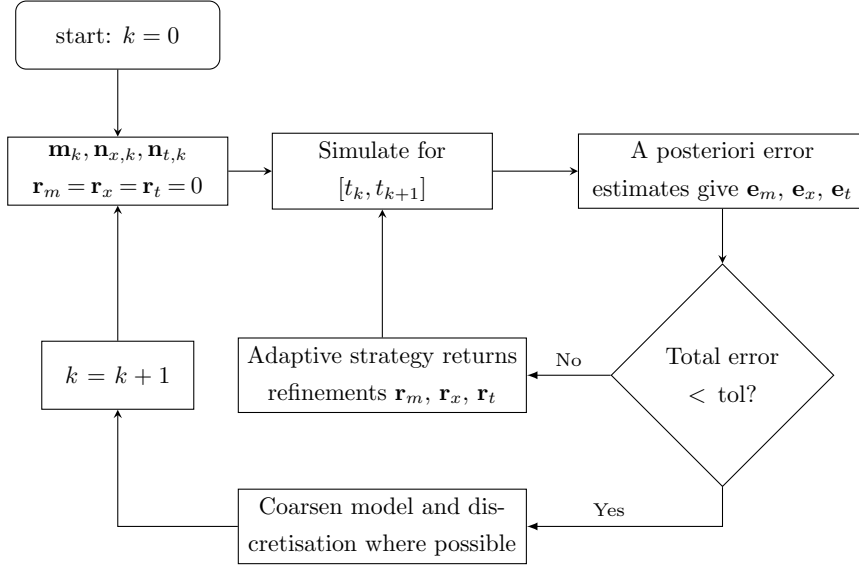


Figure 2: Gas flow adaptive simulation process using an adaptive refinement strategy which returns a set of refinements \mathbf{r}_m , \mathbf{r}_x and \mathbf{r}_t .

2.3. Structure and Aim of Adaptive Strategies

Our focus lies on finding adaptive strategies that control the errors and drive the
 60 simulation. For the time interval $[t_k, t_{k+1}]$, an adaptive strategy takes as input
 the error distributions $\mathbf{e}_m, \mathbf{e}_x, \mathbf{e}_t$, the model distribution \mathbf{m}_k and the number
 of nodes in the spatial and temporal discretisations $\mathbf{n}_{x,k}, \mathbf{n}_{t,k}$. The strategy
 returns a refinement scheme $\mathbf{r}_m = [r_{m,1}, \dots, r_{m,N_p}]^T$, $\mathbf{r}_x = [r_{x,1}, \dots, r_{x,N_p}]^T$
 and $\mathbf{r}_t = [r_{t,1}, \dots, r_{t,N_p}]^T$, where $r_{m,j} \in \{0, 1, 2\}$ and $r_{x,j}, r_{t,j} \in \mathbb{N}$ denote the
 65 number of refinements to be made in the models and in the discretisations for
 all pipes $j \in \mathcal{J}_p$ such that constraint (2) is satisfied. The aim of the adaptive
 strategies is to achieve this constraint while keeping the computational costs
 that are incurred in the simulation low.

3. Refinement Strategies

In this section, we discuss three strategies for the adaptive refinements in the network simulation. Errors $\mathbf{e}_m, \mathbf{e}_x, \mathbf{e}_t \in \mathbb{R}^{N_p}$ and refinements $\mathbf{r}_m, \mathbf{r}_x, \mathbf{r}_t \in \mathbb{N}^{N_p}$ are assigned to every pipe $j \in \mathcal{J}_p$. One refinement in the spatial mesh or temporal discretisation is defined to be taking the new stepsize as half of its previous value. Hence, approximate relations between the initial errors e and the errors after r refinements $e(r)$ are given by

$$e_{x,j}(r_{x,j}) \approx \frac{e_{x,j}}{2^{s_x \cdot r_{x,j}}}, \quad e_{t,j}(r_{t,j}) \approx \frac{e_{t,j}}{2^{s_t \cdot r_{t,j}}}, \quad \text{for all } j \in \mathcal{J}_p, \quad (3)$$

70 where s_x and s_t are the convergence orders of the spatial and temporal discretisation schemes. The reduction for the error is only an approximation. Therefore, to have a safe upper bound for the estimated error after refinement, we multiply these approximated errors by a *safety factor of refinement* $f_r > 1$ in each of the pipes that require refinements to be made. This ensures that it is very unlikely
75 that the error overshoots its estimated value after the refinement.

In the model hierarchy presented in Fig. 1, discretisation errors feature only in the models M1 and M2. The algebraic model M3 has no discretisation errors. Thus, when models are switched from M3 to M2, discretisation errors are introduced. For pipe j simulated with the most detailed model M1, we set the
80 model error to $e_{m,j} = 0$.

Strategy 1 - Individual Bounds (S1)

In order to meet the tolerance of the network error, we derive fixed individual error bounds for individual pipes for each of the three error types. The simulation is then carried out such that for each pipe the error is below the individual
85 bound for all three errors.

We set the tolerance for the model errors as

$$\text{tol}_m = \kappa \cdot \text{tol}, \quad \kappa \in (0, 1).$$

The remaining tolerance is equally divided between the spatial and temporal discretisations, i.e.,

$$\text{tol}_x = \text{tol}_t = (1 - \kappa) / 2 \cdot \text{tol}$$

are the bounds for both error types for the entire network. To get the bounds for individual pipes, we uniformly distribute these bounds over the entire network, i.e., we divide them by the number of pipes N_p .

For the refinements, first a discretisation refinement is computed to bring the
 90 errors below the respective tolerance for each pipe. Subsequently, if the network error still exceeds the tolerance, models are refined to the next model higher up in the hierarchy. The simulation errors are then re-evaluated and this cycle is repeated until (2) is satisfied. This strategy is discussed in detail in [14]. The pseudocode for *Individual Bounds* is given in Algorithm 1.

95 *Strategy 2 - Maximal Error Refinement (S2)*

Since Strategy 1 assigns individual tolerances, it loses the view of the network as a whole. However, the contribution of different errors to the overall network can balance each other without overshooting the total network tolerance. This is accounted for in the following strategy where we seek to make only those refinements which result in the maximal error reduction. This results in an iterative procedure. For every pipe j , we compute in every iteration the error reduction due to a single refinement in the model $\Delta e_{m,j}$ and in the space and time discretisations $\Delta e_{x,j}$ and $\Delta e_{t,j}$. The best option

$$b_j = \max_{j \in \mathcal{J}_p} \{ \Delta e_{m,j}, \Delta e_{x,j}, \Delta e_{t,j} \}$$

is passed to the network. On the network level, we mark those refinements for which the error reductions are larger than $\phi \cdot \max \{b\}$, with $\phi \leq 1$. This iteration is repeated until the network error is brought below the tolerance. The function *maximalErrorRefinement*, see Algorithm 2, represents the network controller
 100 and the function *errorReduction*, see Algorithm 3, represents the pipe level computations.

The spatial and temporal discretisation errors also depend on the simulation model. In lines 2 and 3 of Algorithm 3, we use $F_x(m_c)$ and $F_t(m_c)$, with m_c the current model, as error amplification factors for the discretisation errors which
105 account for this model dependency. Here, $m_c = 1, 2, 3$ refers to models M1, M2 and M3, respectively. Since the discretisation error is absent for $m_c = 3$, we set $F_x(3) = F_t(3) = 0$. Furthermore, we set $F_x(2) = F_t(2) = 1$. For model M1, the amplification factor for the discretisation errors is set with respect to the

Algorithm 1 individualBound

Input: $\mathbf{e}_m, \mathbf{e}_x, \mathbf{e}_t, \text{tol}, s_x, s_t$

Output: $\mathbf{r}_m, \mathbf{r}_x, \mathbf{r}_t$

```

1: while networkError > tol ·  $M(\mathbf{u}^h)$  do
2:   for  $j = 1, \dots, N_p$  do
3:     if  $e_{x,j} > \text{tol}_x \cdot M(\mathbf{u}^h)/N_p$  then
4:        $r_{x,j} \leftarrow \text{ceil} \left( \frac{\log(f_r e_{x,j} N_p / (\text{tol}_x \cdot M(\mathbf{u}^h)))}{\log(2^{s_x})} \right)$ 
5:     end if
6:     if  $e_{t,j} > \text{tol}_t \cdot M(\mathbf{u}^h)/N_p$  then
7:        $r_{t,j} \leftarrow \text{ceil} \left( \frac{\log(f_r e_{t,j} N_p / (\text{tol}_t \cdot M(\mathbf{u}^h)))}{\log(2^{s_t})} \right)$ 
8:     end if
9:   end for
10:  update networkError
11:  if networkError > tol ·  $M(\mathbf{u}^h)$  then
12:    for  $j = 1, \dots, N_p$  do
13:      if  $e_{m,j} > \text{tol}_m \cdot M(\mathbf{u}^h)/N_p$  then
14:         $r_{m,j} \leftarrow r_{m,j} + 1$ 
15:      end if
16:    end for
17:  end if
18:  update networkError
19: end while

```

benchmark model M2. The factor $F_m(a, b)$ denotes an error reduction for the
 110 model error when models are shifted from a to b . The notation $[b, z] = \max\{\cdot\}$ is
 similar to MATLAB notation where b denotes the maximal element and z is the
 corresponding index. Note that in determining Δe_m we also consider changes
 in the spatial and temporal discretisation errors. The central idea is to account
 for net error reduction.

115 *Strategy 3 - Maximal Error-to-Cost Refinement (S3)*

The adaptive refinements are made with an objective of reducing the com-
 putational cost without compromising on the simulation error. However, the
 previous two strategies do not address the computational costs explicitly. They
 address the error tolerance which is merely a constraint to the adaptive strate-
 gies, viewed in the optimization setting (1). In Strategy 3, however, we also take
 into account the computational costs that are incurred by the refinement. The

Algorithm 2 maximalErrorRefinement

Input: $\mathbf{m}, \mathbf{e}_m, \mathbf{e}_x, \mathbf{e}_t, \text{tol}, s_x, s_t, \phi$

Output: $\mathbf{r}_m, \mathbf{r}_x, \mathbf{r}_t$

```

1: for  $j = 1, \dots, N_p$  do
2:    $b_j, z_j \leftarrow \text{errorReduction}(m_j, r_{m,j}, r_{x,j}, r_{t,j}, e_{m,j}, e_{x,j}, e_{t,j}, s_x, s_t)$ 
3: end for
4: while networkError > tol ·  $M(\mathbf{u}^h)$  do
5:   bound  $\leftarrow \phi \cdot \max\{\mathbf{b}\}$ 
6:   for  $j = 1, \dots, N_p$  do
7:     if  $b_j > \text{bound}$  then
8:        $r_{z_j,j} \leftarrow r_{z_j,j} + 1$ 
9:        $b_j, z_j \leftarrow \text{errorReduction}(m_j, r_{m,j}, r_{x,j}, r_{t,j}, e_{m,j}, e_{x,j}, e_{t,j}, s_x, s_t)$ 
10:    end if
11:  end for
12:  update networkError
13: end while

```

idea of this strategy is similar to the greedy approximation algorithm for solving the unbounded knapsack problem, see [17]. Strategy 3, given in Algorithm 4, is similar to Algorithm 2 on the network level. On the pipe level, however, we compute the cost additions $\Delta c_{m,j}, \Delta c_{x,j}, \Delta c_{t,j}$, for all $j \in \mathcal{J}_p$, using the cost functional $F_c(m_c, r_x, r_t)$, for each of the error reductions $\Delta e_{m,j}, \Delta e_{x,j}, \Delta e_{t,j}$. The error controller on pipe level then passes the best option

$$b_j = \max \left\{ \frac{\Delta e_{m,j}}{\Delta c_{m,j}}, \frac{\Delta e_{x,j}}{\Delta c_{x,j}}, \frac{\Delta e_{t,j}}{\Delta c_{t,j}} \right\},$$

i.e., the maximal error-to-cost ratio, to the network. This extension in the error controller on pipe level is given in function *errorToCostReduction* in Algorithm 5.

Algorithm 3 errorReduction

Input: $m, r_m, r_x, r_t, e_m, e_x, e_t, s_x, s_t$

Output: b, z

- 1: **function** safety(x) **return** $1 + (f_r - 1) \text{sign}(x)$
 - 2: **function** spaceError(m,e,r) **return** $\frac{e}{2^{s_x \cdot r}} \cdot \text{safety}(r) \cdot F_x(m)$
 - 3: **function** timeError(m,e,r) **return** $\frac{e}{2^{s_t \cdot r}} \cdot \text{safety}(r) \cdot F_t(m)$
 - 4: $m_c \leftarrow m - r_m$
 - 5: **if** $m_c \neq 1$ **then**
 - 6: $\Delta e_m \leftarrow F_m(m_c, m_c - 1) \cdot e_m$
 $\quad + \text{spaceError}(m_c, e_x, r_x) - \text{spaceError}(m_c - 1, e_x, r_x)$
 $\quad + \text{timeError}(m_c, e_t, r_t) - \text{timeError}(m_c - 1, e_t, r_t)$
 - 7: **else**
 - 8: $\Delta e_m \leftarrow 0$
 - 9: **end if**
 - 10: $\Delta e_x \leftarrow \text{spaceError}(m_c, e_x, r_x) - \text{spaceError}(m_c, e_x, r_x + 1)$
 - 11: $\Delta e_t \leftarrow \text{timeError}(m_c, e_t, r_t) - \text{timeError}(m_c, e_t, r_t + 1)$
 - 12: $[b, z] = \max \{ \Delta e_m, \Delta e_x, \Delta e_t \}$
-

Algorithm 4 maximalErrorToCostRefinement

Input: $\mathbf{m}, \mathbf{e}_m, \mathbf{e}_x, \mathbf{e}_t, \text{tol}, s_x, s_t, \phi$ **Output:** $\mathbf{r}_m, \mathbf{r}_x, \mathbf{r}_t$

Same as Algorithm 2, replacing lines 2 and 9 with

$$b_j, z_j \leftarrow \text{errorToCostReduction}(m, r_{m,j}, r_{x,j}, r_{t,j}, e_{m,j}, e_{x,j}, e_{t,j}, s_x, s_t)$$

Algorithm 5 errorToCostReduction

Input: $m, r_m, r_x, r_t, e_m, e_x, e_t, s_x, s_t$ **Output:** b, z

Lines 1-11 same as Algorithm 3

12: **if** $m_c \neq 1$ **then**

13: $\Delta c_m \leftarrow F_c(m_c - 1, r_x, r_t) - F_c(m_c, r_x, r_t)$

14: **end if**

15: $\Delta c_x \leftarrow F_c(m_c, r_x + 1, r_t) - F_c(m_c, r_x, r_t)$

16: $\Delta c_t \leftarrow F_c(m_c, r_x, r_t + 1) - F_c(m_c, r_x, r_t)$

17: $[b, z] = \max \{ \Delta e_m / \Delta c_m, \Delta e_x / \Delta c_x, \Delta e_t / \Delta c_t \}$

4. Design of Experiment

We tested the performance of the three refinement strategies on 10^4 random samples of error distributions. The random samples represent errors in a simulation of gas flow in a given network of $N_p = 12$ pipes. A better strategy will lead to lower computational costs while reducing the simulation error to a level below the tolerance. We compute the computational cost per pipe using a cost functional of the form

$$F(m, n_x, n_t) = C_m \cdot n_x^{\alpha_m} \cdot n_t^{\beta_m}, \quad (4)$$

where $m \in \{1, 2, 3\}$ denotes the model and C_m, α_m and β_m are model-dependent constants. These constants, given in Table 1, are determined by the method of least squares. For this, gas flow simulations through a single pipe are performed using the software *ANACONDA* (cf. [18, 19]) with many different values of n_x and n_t , which return the corresponding computational cost values F . A plot

showing the computational costs of simulating a single pipe using different models depending on the number of nodes in space and time is depicted in Fig. 3. We can rewrite the functional (4) in terms of refinements assuming that the initial number of nodes $n_{x,0}, n_{t,0}$ are known. Then we get

$$F_c(m, r_x, r_t) = C_m (2^{r_x} \cdot n_{x,0})^{\alpha_m} \cdot (2^{r_t} \cdot n_{t,0})^{\beta_m} . \quad (5)$$

The initial number of space and time discretisation nodes are chosen from the interval $[100, 200]$ using a random number generator. All models are set to the most simple model M3 in the beginning. The error reduction upon model refinement also takes into account the introduction or increase of the spatial and temporal errors. This requires that the spatial and temporal errors are small when compared to the model error. Hence, for the experiment, the initial model errors are taken from $\mathcal{U}[0, 1]$, where $\mathcal{U}[a, b]$ denotes a uniform probability distribution on the interval $[a, b]$, and the spatial and temporal discretisation errors are taken from $\mathcal{U}[0, 0.2]$. For the model and discretisation errors after refinement, the approximate error reductions in lines 2, 3 and 6 of Algorithm 3 are used, where we choose $F_m(3, 2) = 3/4$ and $F_m(2, 1) = 1/4$. Models M1 and M2 are discretised with the implicit box scheme, as in [20], for which it holds that $s_x = 2$ and $s_t = 1$. The parameter $\kappa = 1/3$ for Strategy 1 is chosen such that all three errors have an equal fraction of the tolerance. Strategies 2 and 3 are tested for a fraction $\phi \in \{0.8, 0.9, 1\}$ of the maximal best option. The strategies work for a relative error tolerance of $\text{tol} = 10^{-1}$ with a target functional value $M(\mathbf{u}^h) = 2.5 \cdot N_p = 30$. Hence, we require for the total network

Table 1: Cost functional constants in (4) and (5).

m	C_m	α_m	β_m
1	$8.45 \cdot 10^{-5}$	0.952	0.937
2	$1.06 \cdot 10^{-4}$	0.908	0.925
3	$5.49 \cdot 10^{-5}$	0.694	0.857

simulation error that

$$\sum_{j \in \mathcal{J}_p} (e_{m,j} + e_{x,j} + e_{t,j}) < \text{tol} \cdot M(\mathbf{u}^h) = 3.$$

120 The results of this experiment are given in the next section.

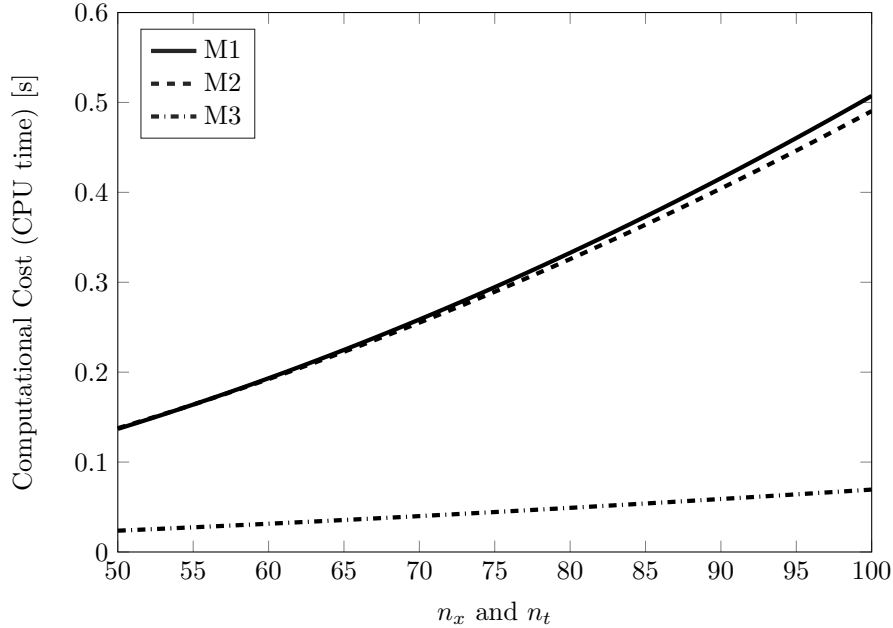


Figure 3: Computational costs for the simulation of a single pipe using different models from the hierarchy against the number of nodes in space n_x and time n_t .

5. Results and Discussions

Each strategy returns a refinement scheme which brings the simulation error below the tolerance. The goal is to have low computational costs. The mean of the total computational cost values in CPU seconds over 10^4 samples is shown in Table 2. We show the percentage savings in mean total computational cost of the strategies with respect to Strategy 1. We denote the strategies as $S1 - S3$. The subscripts 1, 2, 3 refer to $\phi = 0.8, 0.9$ and 1, respectively. We observe that strategies $S2$ and $S3$ have a percentage saving of over 77% with respect to $S1$

for all values of $\phi \in \{0.8, 0.9, 1\}$. Among the different values, $\phi = 1$ performs
 130 best for both $S2$ and $S3$.

Thus, by working with a greedy-like strategy for error control, an adaptive process can reduce the computational cost significantly. Furthermore, accounting for the computational cost explicitly in our estimates, we find even better refinement schemes that result in lower computational costs.

Table 2: Mean total cost values in CPU seconds for strategies $S1 - S3$ and savings with respect to $S1$. The subscripts 1, 2, 3 denote the different values of $\phi = 0.8, 0.9, 1$, respectively.

strategies	$S1$	$S2_1$	$S2_2$	$S2_3$	$S3_1$	$S3_2$	$S3_3$
Mean Total Cost	36.1	8.30	8.04	7.72	7.64	7.39	7.12
Savings (%)	-	77.0	77.7	78.6	78.8	79.5	80.3

135 6. Application to a Realistic Network Simulation

We now apply the three different strategies to a simulation of a gas supply network, which is shown in Fig. 4. The considered network consists of twelve pipes (P01 – P12, with lengths between 30km and 100km), two sources (S01 – S02), four consumers (C01 – C04), three compressor stations (Comp01 - Comp03) and
 140 one control valve (CV01). Starting with stationary initial data, the boundary conditions and the control for the compressor stations and the control valve are time-dependent. The simulation time is 14 400 s. The target functional $M(\mathbf{u})$ is given by the total fuel gas consumption of the three compressors and the error estimators are evaluated using a dual weighted residual method as developed,
 145 for example, in [21–24]. For details of the derivation of the error estimators we refer to [14, 25]. The simulation is performed using the software *ANACONDA* (cf. [18, 19]).

Remark 1. For the strategies proposed in Section 3, the temporal error was considered individually for each pipe. Since in the implementation of *ANACONDA*,
 150 *CONDA*, the time stepping is uniform for the entire network, the temporal

error η_t was computed globally and divided by the number of pipes in order to get a local temporal error. However, if a best option B_j was supposed to be the temporal error and had to be refined then all pipes were refined in time uniformly and the best options B_j were updated.

155 A reference solution was computed using model M1 and a very fine discretisation. The strategies were run with a relative tolerance of $\text{tol} = 10^{-4}$. Table 3 shows the relative error of the simulation compared to the reference solution, the CPU time taken and the percentage savings of strategy $S2$ and $S3$ in relation to strategy $S1$.

160 Compared to the synthetic experiment in Section 5, we see that the savings of strategies $S2$ and $S3$ applied to the simulation are in a similar range. The choice of the parameter $\phi \in \{0.8, 0.9, 1.0\}$, however, does not seem to have a significant influence on the saving. Moreover, the *Maximal Error-to-Cost* strategy $S3$ does not result in a larger saving of CPU time. What is noticeable is that the relative errors of the strategies $S2$ and $S3$ are closer to the proposed relative tolerance, which shows that they are not as restrictive as the *Individual Bounds* strategy $S1$.

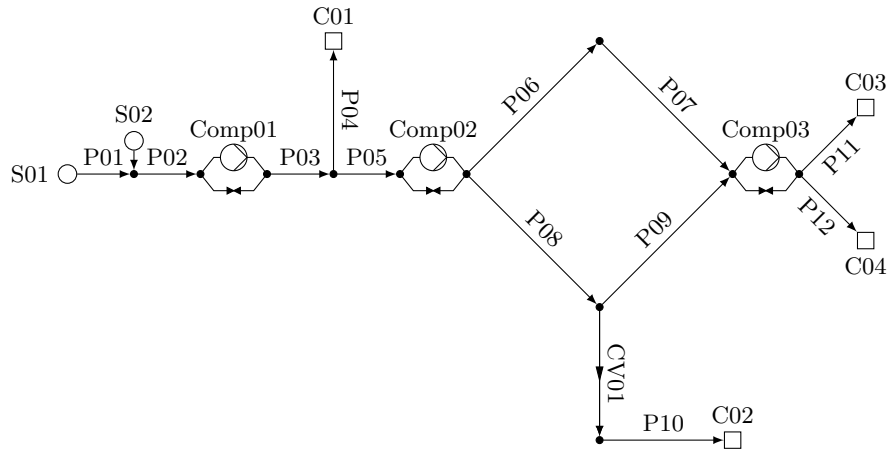


Figure 4: Gas supply network with compressor stations and a control valve.

Table 3: Relative error ($\text{tol} = 10^{-4}$) and computational costs (in s) of a network simulation using strategies $S1 - S3$ and savings with respect to $S1$. The subscripts 1, 2, 3 denote the different values of $\phi \in \{0.8, 0.9, 1\}$, respectively.

Strategy	Relative Error	CPU time [s]	Savings
$S1$	$1.384 \cdot 10^{-5}$	7.53	-
$S2_1$	$3.730 \cdot 10^{-5}$	2.42	67.9%
$S2_2$	$3.091 \cdot 10^{-5}$	2.27	69.9%
$S2_3$	$3.091 \cdot 10^{-5}$	2.29	69.9%
$S3_1$	$5.076 \cdot 10^{-5}$	3.12	58.6%
$S3_2$	$5.190 \cdot 10^{-5}$	2.83	62.4%
$S3_3$	$5.098 \cdot 10^{-5}$	2.83	62.4%
reference	-	1013	

7. Conclusions

In this paper we address the problem of automatic error control for large scale
170 gas flow simulations that use a model hierarchy. The simulation needs to be
reliable, i.e., keeping the total relative error below a specified tolerance, while
retaining low computational costs. The problem of finding an optimal refine-
ment strategy is a generalisation of the knapsack problem. We present three
strategies for adaptive simulation error control via spatial and temporal discreti-
175 sation mesh and model refinements. The strategy *Individual Bounds*, which is
currently implemented in *ANACONDA*, sets a uniform bound for each error type
and each pipe, *Maximal Error Refinement* iteratively chooses those refinements
that result in the largest error reduction and has a network overview, and *Max-*
imal Error-to-Cost Refinement also accounts for the increase in computational
180 cost inflicted by the refinement.

We constructed a synthetic experiment to test the three strategies. From this
experiment we see that the two greedy-like strategies significantly reduce the
computational cost as compared to the *Individual Bounds* strategy. This result
is largely reflected in an actual gas flow simulation using *ANACONDA* for a 12

185 pipe network including compressor stations and a control valve. Especially
when the simulation process is a key component in a gas flow optimisation
problem, the novel refinement strategies lead to considerable computational
savings without compromising on the simulation accuracy.

Acknowledgements

190 The authors would like to thank the Deutsche Forschungsgemeinschaft for their
support within the projects B01 and B03 of the Collaborative Research Centre/
Transregio 154 *Mathematical Modelling, Simulation and Optimization using the
Example of Gas Networks*. Jens Lang was also supported by the Excellence
Initiative of the German Federal and State Governments via the Darmstadt
195 Graduate School of Excellence Energy Science and Engineering.

References

- [1] M. A. Adewumi, J. Zhou, Simulation of transient flow in natural gas
pipelines, 27th Annual Meeting of PSIG (Pipeline Simulation Interest
Group), Albuquerque, NM (1995).
200 URL <https://www.onepetro.org/conference-paper/PSIG-9508>
- [2] M. K. Banda, M. Herty, A. Klar, Coupling conditions for gas networks
governed by the isothermal Euler equations, *Netw. Heterog. Media* 1 (2)
(2006) 295–314. doi:10.3934/nhm.2006.1.295.
- [3] M. K. Banda, M. Herty, A. Klar, Gas flow in pipeline networks, *Netw.
205 Heterog. Media* 1 (1) (2006) 41–56. doi:10.3934/nhm.2006.1.41.
- [4] K. S. Chapman, P. Krishniswami, V. Wallentine, M. Abbaspour, R. Ran-
ganathan, R. Addanki, J. Sengupta, L. Chen, Virtual pipeline system
testbed to optimize the U.S. natural gas transmission pipeline system,
Tech. Rep. DE-FC26-01NT41322, The National Gas Machinery Labora-
210 tory, Kansas State University (2005). arXiv:metadc794805.

- [5] R. M. Colombo, M. Garavello, A well-posed Riemann problem for the p -system at a junction, *Netw. and Heterog. Media* 1 (3) (2006) 495–511. doi:10.3934/nhm.2006.1.495.
- [6] P. Domschke, O. Kolb, J. Lang, Adjoint-based error control for the simulation and optimization of gas and water supply networks, *Appl. Math. Comput.* 259 (2015) 1003–1018. doi:10.1016/j.amc.2015.03.029.
- [7] K. Ehrhardt, M. C. Steinbach, Nonlinear optimization in gas networks, in: H. G. Bock, E. Kostina, H. X. Phu, R. Ranacher (Eds.), *Modeling, Simulation and Optimization of Complex Processes*, Springer, Berlin Heidelberg, 2005, pp. 139–148. doi:10.1007/3-540-27170-8_11.
- [8] K. Ehrhardt, M. C. Steinbach, KKT systems in operative planning for gas distribution networks, *Proc. Appl. Math. Mech.* 4 (1) (2004) 606–607. doi:10.1002/pamm.200410284.
- [9] M. Herty, J. Mohring, V. Sachers, A new model for gas flow in pipe networks, *Math. Methods Appl. Sci.* 33 (7) (2010) 845–855. doi:10.1002/mma.1197.
- [10] S. L. Ke, H. C. Ti, Transient analysis of isothermal gas flow in pipeline networks, *Chem. Eng. J.* 76 (2) (2000) 169–177. doi:10.1016/S1385-8947(99)00122-9.
- [11] A. Martin, M. Möller, S. Moritz, Mixed integer models for the stationary case of gas network optimization, *Math. Program.* 105 (2) (2006) 563–582. doi:10.1007/s10107-005-0665-5.
- [12] A. J. Osiadacz, M. Chaczykowski, Comparison of isothermal and non-isothermal pipeline gas flow models, *Chem. Eng. J.* 81 (13) (2001) 41–51. doi:10.1016/S1385-8947(00)00194-7.
- [13] M. C. Steinbach, On PDE solution in transient optimization of gas networks, *J. Comput. Appl. Math.* 203 (2) (2007) 345–361, special Issue: The

first Indo-German Conference on PDE, Scientific Computing and Optimization in Applications. doi:10.1016/j.cam.2006.04.018.

- 240 [14] P. Domschke, Adjoint-based control of model and discretization errors for gas transport in networked pipelines, Ph.D. thesis, TU Darmstadt (2011).
- [15] H. Kellerer, U. Pfersch, D. Pisinger, Knapsack problems, Springer-Verlag, Berlin, 2004. doi:10.1007/978-3-540-24777-7.
- [16] P. Domschke, O. Kolb, J. Lang, Adjoint-based control of model and discretisation errors for gas and water supply networks, in: X. Yang, S. Koziel (Eds.), Computational Optimization and Applications in Engineering and Industry, Springer, Berlin Heidelberg, 2011, pp. 1–17. doi:10.1007/978-3-642-20986-4.
- 245
- [17] G. B. Dantzig, Discrete-variable extremum problems, Operations Res. 5 (1957) 266–277.
- 250
- [18] O. Kolb, Simulation and optimization of gas and water supply networks, Ph.D. thesis, TU Darmstadt (2011).
- [19] O. Kolb, J. Lang, P. Bales, An implicit box scheme for subsonic compressible flow with dissipative source term, Numer. Algorithms 53 (2) (2010) 293–307. doi:10.1007/s11075-009-9287-y.
- 255
- [20] O. Kolb, J. Lang, P. Bales, An implicit box scheme for subsonic compressible flow with dissipative source term, Numer. Algorithms 53 (2-3) (2010) 293–307. doi:10.1007/s11075-009-9287-y.
- [21] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta Numer. 10 (2001) 1–102. doi:10.1017/S0962492901000010.
- 260
- [22] M. Braack, A. Ern, A posteriori control of modeling errors and discretization errors, Multiscale Model. Simul. 1 (2) (2003) 221–238. doi:10.1137/S1540345902410482.

- 265 [23] R. Rannacher, Adaptive finite element discretization of flow problems for goal-oriented model reduction, in: H. Choi, H. Choi, J. Yoo (Eds.), Computational Fluid Dynamics 2008, Springer Berlin Heidelberg, 2009, pp. 31–45. doi:10.1007/978-3-642-01273-0_3.
- [24] M. Besier, R. Rannacher, Goal-oriented space-time adaptivity in the finite element galerkin method for the computation of nonstationary incompressible flow, Int. J. Numer. Meth. Fl. 70 (9) (2012) 1139–1166. doi:10.1002/flid.2735.
- 270 [25] P. Domschke, O. Kolb, J. Lang, Adjoint-based error control for the simulation and optimization of gas and water supply networks, Applied Mathematics and Computation 259 (2015) 1003 – 1018. doi:10.1016/j.amc.2015.03.029.
- 275