

## ADAPTIVE REFINEMENT STRATEGIES FOR THE SIMULATION OF GAS FLOW IN NETWORKS USING A MODEL HIERARCHY\*

PIA DOMSCHKE<sup>†</sup>, ASEEM DUA<sup>‡</sup>, JEROEN J. STOLWIJK<sup>‡</sup>, JENS LANG<sup>†§</sup>, AND VOLKER MEHRMANN<sup>‡</sup>

**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 natural 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.

**Key words.** gas supply networks, model hierarchy, error estimators, model adaptivity, refinement strategies

**AMS subject classifications.** 65K99, 65Z99, 65M22, 35Q31

**1. Introduction.** The simulation of natural 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 natural 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, 2, 3, 14, 15, 22, 25, 26, 28, 29, 34, 37, 38]. Depending upon requirement, there exist multiple models, based on the Euler equations of fluid dynamics, 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 an adaptive strategy to automatically steer the simulation by changing the models and the discretisation meshes.

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

---

\*Received February 6, 2017. Accepted January 8, 2018. Published online on April 5, 2018. Recommended by Alexander Ostermann. The authors thank the DFG for their support within projects B01 and B03 in CRC TRR 154. Jens Lang was also supported by the Excellence Initiative of the German Federal and State Governments via the Darmstadt Graduate School of Excellence Energy Science and Engineering.

<sup>†</sup>Department of Mathematics, TU Darmstadt, Dolivostr. 15, 64293 Darmstadt, Germany (`{domschke, lang}@mathematik.tu-darmstadt.de`).

<sup>‡</sup>Institut für Mathematik, MA 4-5, TU Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany (`dua.aseem@gmail.com, {stolwijk, mehrmann}@math.tu-berlin.de`).

<sup>§</sup>Graduate School of Energy Science and Engineering, TU Darmstadt, Jovanka-Bontschits-Str. 2, 64287 Darmstadt, Germany.

define an optimal refinement strategy as a strategy which returns the solution of the constrained optimisation problem

$$\begin{aligned}
 (1.1) \quad & \min_{r_i} c + \sum_{i=1}^{3N_p} \sum_{k=1}^{r_i} \Delta c_{ik} \\
 & \text{s.t. } |\eta| - \sum_{i=1}^{3N_p} \sum_{k=1}^{r_i} \Delta \eta_{ik} \leq \text{tol}.
 \end{aligned}$$

Here, the constants  $c$  and  $\eta$  denote the cost and relative error of the starting simulation, respectively. For each refinement possibility  $i$ ,  $r_i$  is the number of refinements,  $\Delta \eta_{ik}$  the relative error reduction due to the  $k$ th subsequent refinement, and  $\Delta c_{ik}$  the corresponding cost addition. We note that if  $\Delta c_{ik}$ ,  $\Delta \eta_{ik}$  are constant for all  $k$ , then this problem is equivalent to the unbounded knapsack problem, which is NP-hard; see, e.g., [30]. We aim to find a good approximation to the solution of this generalisation of the knapsack problem. For this, we examine three adaptive refinement strategies which return approximate solutions of (1.1). We call these strategies *individual tolerances* (S1), *maximal error refinement* (S2), and *maximal error-to-cost refinement* (S3). The ideas of the first two strategies are frequently used in practice for the adaptation of the computational mesh; cf. the overview given in [4, Section 5.2]. An effective adaptive mesh refinement strategy for hyperbolic partial differential equations (PDEs) on a rectangular domain has been developed in [5, 6, 7, 8]. Based on the ideas of this strategy, a mesh and model adaptation strategy for hyperbolic PDEs on a pipe network has been presented in [17, 19, 20, 21, 22, 31]. The latter strategy is almost identical to Strategy S1. The idea of Strategy S2 is similar to the technique that is used in the adaptive finite element method; see, e.g., [11, 12, 13, 23, 24]. Strategy S2 also bears similarities to the adaptive strategy in a general multiscale setting that is discussed in [35]. Strategy S3 is based on the idea of a greedy approximation algorithm for solving the unbounded knapsack problem; see [16]. The aim of this paper is to compare the performance of the three mesh and model refinement Strategies S1–S3 on a gas pipeline network with respect to their computational cost incurred. The ideas presented here are for the example of pipe networks. By generalising pipes to functional sub-domains, the described 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 both a synthetic experiment on an abstract gas network and an application of the refinement strategies to a realistic network simulation. Finally, some conclusions are given in Section 5.

**2. Simulation of gas flow.** In this section, we present a framework for the simulation of gas flow in a pipe network. We give a description of a gas network in Section 2.1, including a model hierarchy for the pipes and the fuel gas consumption of a compressor. Model and discretisation error estimators w.r.t. a user-defined functional are derived in Section 2.2. We then outline a framework for the adaptive simulation of pipe networks using the given model hierarchy in Section 2.3. Finally, we highlight the aims of the refinement strategies within the simulation framework in Section 2.4.

**2.1. Gas network.** We follow the framework given in [17, 19] to describe the gas network. The network is modelled as a directed graph  $\mathcal{G} = (\mathcal{J}, \mathcal{V})$  with edges  $\mathcal{J}$  and vertices  $\mathcal{V}$ . The set of edges  $\mathcal{J}$  consists of pipes  $j \in \mathcal{J}_p$ , compressor stations  $c \in \mathcal{J}_c$ , and valves  $\mathcal{J}_v$ . 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 described in Section 2.1.1 holds and adequate initial and coupling as well as boundary conditions have to be specified. Valves and compressors are described by algebraic equations.

**2.1.1. Model hierarchy.** As an example, we take a three-model hierarchy for the gas flow simulations, discussed in detail in [20]. The *isothermal Euler equations* [33] consist of the continuity and the momentum equation together with the equation of state for real gases. We simplify these equations by assuming the pipe to be horizontal and the speed of sound to be constant, yielding a *nonlinear model*

$$(M1) \quad \begin{aligned} p_t + \frac{\rho_0 c^2}{A} q_x &= 0, \\ q_t + \frac{A}{\rho_0} p_x + \frac{\rho_0 c^2}{A} \left( \frac{q^2}{p} \right)_x &= -\frac{\lambda \rho_0 c^2}{2DA} \frac{|q|q}{p}. \end{aligned}$$

Here,  $q = A\rho v/\rho_0$  denotes the mass flow rate under standard conditions (1 atm air pressure, temperature of 0 °C),  $p$  denotes the pressure,  $c = \sqrt{p/\rho}$  the speed of sound,  $A$  the cross-sectional area of the pipe,  $\lambda > 0$  the Darcy friction coefficient,  $D$  the pipe diameter,  $\rho$  the gas density,  $v$  the gas velocity, and  $\rho_0$  the density under standard conditions. If the gas velocity is much smaller than the speed of sound, i.e., if  $|v| \ll c$ , then we can neglect the nonlinear term in the spatial derivative of the momentum equation in **M1**. This results in a *semilinear model* [17, 36]

$$(M2) \quad \begin{aligned} p_t + \frac{\rho_0 c^2}{A} q_x &= 0, \\ q_t + \frac{A}{\rho_0} p_x &= -\frac{\lambda \rho_0 c^2}{2DA} \frac{|q|q}{p}. \end{aligned}$$

A further simplification of assuming a stationary state, i.e., setting the time derivatives to zero, yields a system of two ordinary differential equations, which can be solved analytically and are referred to as the *algebraic model*

$$(M3) \quad \begin{aligned} q &= \text{const.}, \\ p(x) &= \sqrt{p_{\text{in}}^2 - \frac{\lambda \rho_0^2 c^2 |q|q}{DA^2} x}. \end{aligned}$$

Here,  $p_{\text{in}} = p(0)$  denotes the pressure at the inbound of the pipe. The three models are shown in hierarchical form in Figure 2.1. The model hierarchy is set in the decreasing order of accuracy for our purpose of gas flows. In general, models **M1** and **M2** can not be solved analytically. Therefore, a discretisation method has to be applied in order to obtain an approximate discrete solution. In practical gas flow applications, box schemes, which are introduced in [39], have been shown to be effective discretisation schemes for these two systems of hyperbolic PDEs. We apply the implicit box scheme, which is conservative and stable under mild conditions [18]. For the scalar balance law

$$u_t + f(u)_x = g(u),$$

with initial conditions  $u(x, 0) = u_0(x)$ , this box scheme is given by

$$(2.1) \quad \frac{u_{i-1}^\ell + u_i^\ell}{2} = \frac{u_{i-1}^{\ell-1} + u_i^{\ell-1}}{2} - \frac{\Delta t}{\Delta x} (f_i^\ell - f_{i-1}^\ell) + \Delta t \frac{g_{i-1}^\ell + g_i^\ell}{2},$$

where  $u_i^\ell = u(x_i, t_\ell)$  and  $f_i^\ell = f(u(x_i, t_\ell))$ . The implicit box scheme is convergent of order 2 in space and order 1 in time [32]. Each pipe in the network is simulated using one of the three models and varying discretisation stepsizes in space and time.

**2.1.2. Compressor station.** The pressure of the gas is increased using a compressor station, which consists of several compressors. A compressor consumes some of the gas during its operation. The equation for the fuel gas consumption of a compressor  $c \in \mathcal{J}_c$  is given by [27]

$$(2.2) \quad G_c(t) = c_F q_{\text{in}}(t) \left( \left( \frac{p_{\text{out}}(t)}{p_{\text{in}}(t)} \right)^{\frac{\gamma-1}{\gamma}} - 1 \right),$$

with in- and outgoing pressure  $p_{\text{in}}$ ,  $p_{\text{out}}$ , and ingoing flow rate  $q_{\text{in}}$ . The parameter  $c_F$  is a compressor specific constant and  $\gamma$  the isentropic coefficient of the gas.

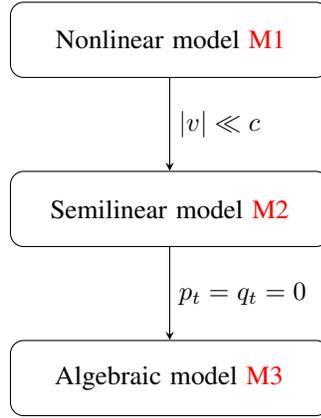


FIG. 2.1. The model hierarchy for the simulation of gas flow that is considered in this paper.

**2.2. Error estimators.** Using the solution of adjoint equations as done in [4, 9, 10, 17, 19], we derive a model and a discretisation error estimator which measure the influence of the model and the discretisation on a user-defined output functional  $M$ . The functional  $M$  can be of any form, for example measuring the fuel gas consumption of the compressor stations via

$$M(p, q) = \sum_{c \in \mathcal{J}_c} \int_0^T G_c(t) dt$$

with  $G_c(t)$  being the fuel gas consumption of the individual compressor  $c \in \mathcal{J}_c$  as given in (2.2). In the following, we consider the arbitrary cost functional

$$\begin{aligned}
 M(p, q) = & \int_0^T \int_{\Omega} N(p, q) dx dt + \int_{\Omega} N_T(p, q) dx \\
 & + \sum_{v \in \mathcal{V}} \int_0^T N_v(p, q) dt + \sum_{i \in \mathcal{J}_{\text{ALG}}} \int_0^T N_i(\mathbf{p}_i, \mathbf{q}_i) dt,
 \end{aligned}$$

where  $\Omega = \sum_{j \in \mathcal{J}_{\text{PDE}}} [x_j^a, x_j^b]$ ,  $\mathbf{p}_i = [p(x_i^a, t), p(x_i^b, t)]^T$ , and  $\mathbf{q}_i = [q(x_i^a, t), q(x_i^b, t)]^T$ . The sets  $\mathcal{J}_{\text{ALG}}$  and  $\mathcal{J}_{\text{PDE}}$  contain the arcs that are modelled by algebraic equations (i.e., pipes using model M3 or other network components like valves or compressor stations) and arcs that are modelled by PDEs (i.e., pipes using models M1 or M2), respectively. The components  $N(p, q)$ ,  $N_v(p, q)$ , and  $N_i(\mathbf{p}_i, \mathbf{q}_i)$  define tracking-type costs on the respective sets ( $\Omega$ , specific

nodes, or algebraic arcs) in the whole time interval  $(0, T)$  and the component  $N_T(p, q)$  defines costs on  $\Omega$  at the final time  $T$ . The adjoint equations of model **M2** with respect to  $M(p, q)$  are given by

$$(2.3) \quad \begin{aligned} \xi_{1t} + \frac{A}{\rho_0} \xi_{2x} &= -\frac{\lambda \rho_0 c^2}{2DA} \frac{|q|q}{p^2} \xi_2 - \frac{\partial N(p, q)}{\partial p}, \\ \xi_{2t} + \frac{\rho_0 c^2}{A} \xi_{1x} &= \frac{\lambda \rho_0 c^2}{DA} \frac{|q|}{p} \xi_2 - \frac{\partial N(p, q)}{\partial q}, \end{aligned}$$

together with appropriate “initial”, coupling, and node conditions. Note that in the adjoint equation (2.3) only the tracking-type component  $N(p, q)$  appears. The remaining components occur in the “initial”, coupling, and node conditions; cf. [17, Eq. (3.33)] or [22, Eq. (20)]. The solution  $\xi = [\xi_1, \xi_2]^T$  of the adjoint equations consists of the adjoint pressure and mass flow rate of the semilinear model **M2** with respect to the functional  $M(p, q)$ . Let  $\mathbf{u} = [p, q]^T$  be the solution of the nonlinear model **M1** and  $\mathbf{u}^h = [p^h, q^h]^T$  the discretised solution of the semilinear model **M2**. For the discretisation of **M2** we apply the implicit box scheme in (2.1). Then, the difference between the output functional  $M(\mathbf{u})$  and  $M(\mathbf{u}^h)$  can be approximated using Taylor expansion. Inserting the solution  $\xi$  of the adjoint system (2.3), we get a first order error estimator for the model and the discretisation error given by [17, 19, 22]

$$\begin{aligned} M(\mathbf{u}) - M(\mathbf{u}^h) &\approx \eta_m^{\text{LIN-NL}} + \eta_h^{\text{LIN}}, \\ \eta_m^{\text{LIN-NL}} &= \int_0^T \int_{\Omega} -\xi^T \begin{bmatrix} 0 \\ \frac{\rho_0 c^2 (q^h)^2}{Ap^h} \end{bmatrix}_x dx dt, \\ \eta_h^{\text{LIN}} &= \int_0^T \int_{\Omega} -\xi^T \begin{bmatrix} p_t^h + \frac{\rho_0 c^2}{A} q_x^h \\ q_t^h + \frac{A}{\rho_0} p_x^h + \frac{\lambda \rho_0 c^2 |q^h| q^h}{2DAp^h} \end{bmatrix} dx dt. \end{aligned}$$

The discretisation error estimator  $\eta_h^{\text{LIN}}$  may be split up into a temporal and spatial discretisation error estimator as follows. Let  $\mathbf{u}$  be the exact and  $\mathbf{u}^h$  be the discretised solution of the semilinear model **M2**. We use a short notation of **M2**, i.e.,  $\mathbf{u}_t + f(\mathbf{u})_x = g(\mathbf{u})$ , which yields

$$\begin{aligned} \eta_h^{\text{LIN}} &= \int_0^T \int_{\Omega} -\xi^T (\mathbf{u}_t^h + f(\mathbf{u}^h)_x - g(\mathbf{u}^h)) dx dt \\ &= \int_0^T \int_{\Omega} -\xi^T ((\mathbf{u}_t^h - \mathbf{u}_t) + (f(\mathbf{u}^h)_x - f(\mathbf{u})_x) - (g(\mathbf{u}^h) - g(\mathbf{u}))) dx dt \\ &= \underbrace{\int_0^T \int_{\Omega} -\xi^T (\mathbf{u}_t^h - \mathbf{u}_t) dx dt}_{=: \eta_t^{\text{LIN}}} \\ &\quad + \underbrace{\int_0^T \int_{\Omega} -\xi^T ((f(\mathbf{u}^h)_x - f(\mathbf{u})_x) - (g(\mathbf{u}^h) - g(\mathbf{u}))) dx dt}_{=: \eta_x^{\text{LIN}}}, \end{aligned}$$

since  $\mathbf{u}$  is the exact solution of **M2**. The temporal discretisation error estimator can be split up into error estimators for the individual pipes via

$$(2.4) \quad \eta_t^{\text{LIN}} = \int_0^T \sum_{j \in \mathcal{J}_p} \int_{x_j^a}^{x_j^b} -\xi^T (\mathbf{u}_t^h - \mathbf{u}_t) dx dt = \sum_{j \in \mathcal{J}_p} \eta_{t,j}^{\text{LIN}},$$

analogously for the spatial discretisation and model error estimator. For the computation of the discretisation error estimators, the exact solution is approximated by a higher-order reconstruction using neighbouring points. We use a polynomial reconstruction of order 2 for the time derivative and denote it by  $\mathbf{u}_t \approx R_t(\mathbf{u}^h)$ . The spatial derivative of  $f$  and the value of  $g$  are reconstructed with order 4, giving  $f(\mathbf{u})_x \approx R_x(f(\mathbf{u}^h))$  and  $g(\mathbf{u}) \approx R(g(\mathbf{u}^h))$ . Thus, the computed estimators are given by

$$(2.5a) \quad \eta_{t,j}^{\text{LIN}} \approx \int_0^T \int_{x_j^a}^{x_j^b} -\boldsymbol{\xi}^T (\mathbf{u}_t^h - R_t(\mathbf{u}^h)) \, dx \, dt,$$

$$(2.5b) \quad \eta_{x,j}^{\text{LIN}} \approx \int_0^T \int_{x_j^a}^{x_j^b} -\boldsymbol{\xi}^T \left( (f(\mathbf{u}^h))_x - R_x(f(\mathbf{u}^h)) - (g(\mathbf{u}^h) - R(g(\mathbf{u}^h))) \right) \, dx \, dt,$$

$$(2.5c) \quad \eta_{m,j}^{\text{LIN-NL}} = \int_0^T \int_{x_j^a}^{x_j^b} -\boldsymbol{\xi}^T \left[ \begin{array}{c} 0 \\ \frac{\rho_0 c^2 (q^h)^2}{Ap^h} \end{array} \right]_x \, dx \, dt,$$

for all  $j \in \mathcal{J}_p$ . The model error estimator  $\eta_{m,j}^{\text{ALG-LIN}}$  between the algebraic and the semilinear model and the discretisation error estimators  $\eta_{t,j}^{\text{NL}}$  and  $\eta_{x,j}^{\text{NL}}$  for the nonlinear model are derived analogously for every pipe  $j \in \mathcal{J}_p$ ; see [21].

**2.3. 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 the *a posteriori* error estimators in Section 2.2:  $\boldsymbol{\eta}_m = [\eta_{m,1}, \eta_{m,2}, \dots, \eta_{m,N_p}]^T$  for the model errors and  $\boldsymbol{\eta}_x = [\eta_{x,1}, \eta_{x,2}, \dots, \eta_{x,N_p}]^T$  and  $\boldsymbol{\eta}_t = [\eta_{t,1}, \eta_{t,2}, \dots, \eta_{t,N_p}]^T$  for the spatial and temporal discretisation errors, respectively. The simulation error for a single pipe is the sum of all three errors. For the simulation to be valid, the relative error in  $M$  must be below a given tolerance  $tol$ . Hence, we require that

$$(2.6) \quad \frac{|M(\mathbf{u}) - M(\mathbf{u}^h)|}{|M(\mathbf{u}^h)|} \approx \frac{|\sum_{j \in \mathcal{J}_p} (\eta_{m,j} + \eta_{x,j} + \eta_{t,j})|}{|M(\mathbf{u}^h)|} < tol.$$

If the tolerance is not achieved, then 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 discretisations are coarsened if appropriate, the simulation progresses to the next time interval, and the cycle repeats. This simulation flow is shown in Figure 2.2 for an interval  $[t_k, t_{k+1}]$ .

**2.4. Structure and aim of adaptive strategies.** Our focus lies on comparing adaptive strategies that control the errors and drive the simulation. For the time interval  $[t_k, t_{k+1}]$  an adaptive strategy takes as input the error distributions  $\boldsymbol{\eta}_{m,k}$ ,  $\boldsymbol{\eta}_{x,k}$ ,  $\boldsymbol{\eta}_{t,k}$ , the model distribution  $\mathbf{m}_k$ , and the number of (equidistant) nodes in the spatial and temporal discretisations  $\mathbf{n}_{x,k}$ ,  $\mathbf{n}_{t,k}$ . For notational convenience we drop the dependence on  $k$  in the following. The strategy returns a refinement scheme  $\mathbf{r}_m = [r_{m,1}, \dots, r_{m,N_p}]^T$ ,  $\mathbf{r}_x = [r_{x,1}, \dots, r_{x,N_p}]^T$ ,  $\mathbf{r}_t = [r_{t,1}, \dots, r_{t,N_p}]^T$ , where  $r_{m,j} \in \{0, 1, 2\}$ ,  $r_{x,j}, r_{t,j} \in \mathbb{N}$  denote the number of refinements to be made in the models and in the discretisations for all pipes  $j \in \mathcal{J}_p$  such that the

constraint (2.6) 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. The first strategy assigns individual error tolerances to each pipe and executes multiple discretisation refinements simultaneously. The second and third strategy use the overall error tolerance and iteratively refine only the best option(s) in the network.

We first provide some general remarks which hold for every strategy. Relations between the initial numbers of discretisation intervals  $n-1$  and the numbers of intervals after  $r$  refinements  $n(r)-1$  are given by

$$(3.1) \quad \begin{aligned} n_{x,j}(r_{x,j}) - 1 &= 2^{r_{x,j}}(n_{x,j} - 1), \\ n_{t,j}(r_{t,j}) - 1 &= 2^{r_{t,j}}(n_{t,j} - 1), \quad \forall j \in \mathcal{J}_p, \end{aligned}$$

where  $n$  is the number of nodes. Thus, the number of intervals is doubled for a single refinement. We choose this factor 2 such that, given a conservative initialisation for the implicit box scheme (2.1) on the coarser grid, it is possible to obtain a conservative initialisation on the refined grid. Namely, the pointwise values of corresponding grid points are copied from the coarser grid and the intermediate grid points are assigned the arithmetic mean of the two neighbouring grid points [22, 31]. From (3.1) it follows that approximate relations between the initial discretisation error estimators  $\eta$  and the estimators after  $r$  refinements  $\eta(r)$  are given

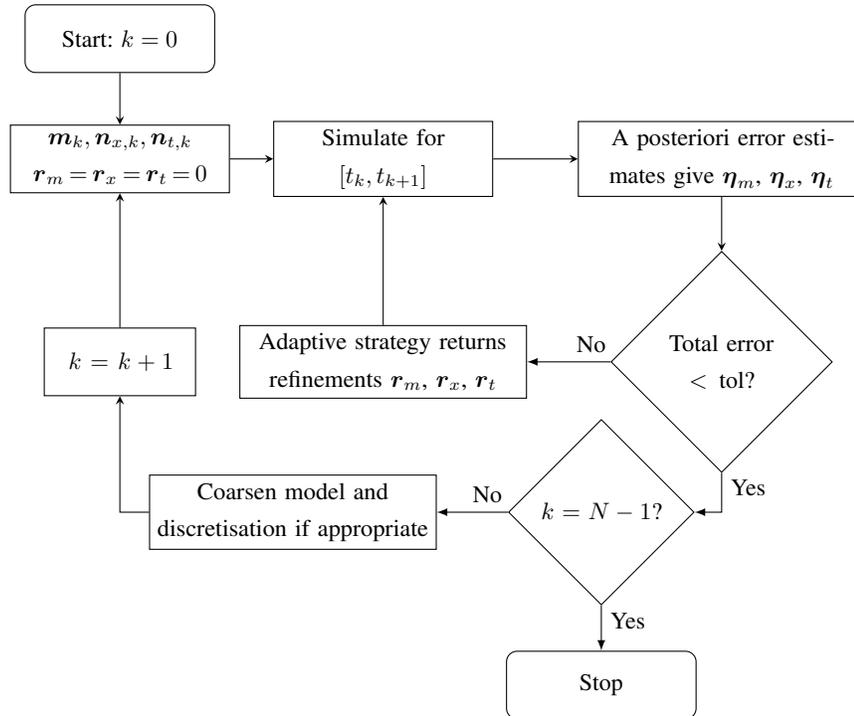


FIG. 2.2. Gas flow adaptive simulation process using an adaptive refinement strategy which returns a set of refinements  $\mathbf{r}_m, \mathbf{r}_x, \mathbf{r}_t$ .

by

$$\eta_{x,j}(r_{x,j}) \approx \frac{\eta_{x,j}}{2^{s_x r_{x,j}}}, \quad \eta_{t,j}(r_{t,j}) \approx \frac{\eta_{t,j}}{2^{s_t r_{t,j}}}, \quad \forall j \in \mathcal{J}_p,$$

where  $s_x$  and  $s_t$  are the convergence orders of the spatial and temporal discretisation schemes, respectively. 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 that the actual error overshoots its estimated value after the refinement. In the model hierarchy presented in Figure 2.1, 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 model error to  $\eta_{m,j} = 0$ .

**Strategy 1: Individual tolerances (S1).** In order to meet the tolerance of the network error, we derive fixed individual error tolerances for the 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 tolerance for all three errors. The pseudocode for *individual tolerances* is given in Algorithm 1. We set the tolerance for the model errors as

$$tol_m = \kappa \cdot tol, \quad \kappa \in (0, 1).$$

---

**Algorithm 1** : Individual tolerances
 

---

**Input:**  $\eta_m, \eta_x, \eta_t, tol, s_x, s_t, f_r$

```

1:  $r_m \leftarrow 0$ 
2: while  $|\sum_{j \in \mathcal{J}_p} (\eta_{m,j} + \eta_{x,j} + \eta_{t,j})| > tol |M(\mathbf{u}^h)|$  do
3:   for  $j = 1, \dots, N_p$  do
4:     if  $|\eta_{x,j}| > tol_x |M(\mathbf{u}^h)| / N_p$  then
5:        $r_{x,j} \leftarrow \text{ceil} \left( \log \left( \frac{f_r |\eta_{x,j}| N_p}{tol_x |M(\mathbf{u}^h)|} \right) \frac{1}{\log(2^{s_x})} \right)$ 
6:     end if
7:     if  $|\eta_{t,j}| > tol_t |M(\mathbf{u}^h)| / N_p$  then
8:        $r_{t,j} \leftarrow \text{ceil} \left( \log \left( \frac{f_r |\eta_{t,j}| N_p}{tol_t |M(\mathbf{u}^h)|} \right) \frac{1}{\log(2^{s_t})} \right)$ 
9:     end if
10:   end for
11:   update  $\eta_m, \eta_x, \eta_t$ 
12:   if  $|\sum_{j \in \mathcal{J}_p} (\eta_{m,j} + \eta_{x,j} + \eta_{t,j})| > tol |M(\mathbf{u}^h)|$  then
13:     for  $j = 1, \dots, N_p$  do
14:       if  $|\eta_{m,j}| > tol_m |M(\mathbf{u}^h)| / N_p$  then
15:          $r_{m,j} \leftarrow r_{m,j} + 1$ 
16:       end if
17:     end for
18:   end if
19:   update  $\eta_m, \eta_x, \eta_t$ 
20: end while

```

**Output:**  $r_m, r_x, r_t$

---

The remaining tolerance  $tol - tol_m = (1 - \kappa)tol$  is equally divided between the spatial and temporal discretisations, i.e.,

$$tol_x = tol_t = (1 - \kappa) / 2 \cdot tol$$

are the tolerances for both error types for the entire network. To get the tolerances for individual pipes, we uniformly distribute these tolerances over the entire network, i.e., we divide them by the number of pipes  $N_p$ . For the refinements, first the number of discretisation refinements are computed that bring the discretisation errors below the respective tolerances for every pipe. That is, for every pipe  $j$  for which

$$(3.2) \quad |\eta_{x,j}| > \frac{tol_x}{N_p} |M(\mathbf{u}^h)|$$

holds, we require for the spatial discretisation error after  $r_{x,j}$  refinements that

$$f_r |\eta_{x,j}(r_{x,j})| = f_r \frac{|\eta_{x,j}|}{2^{s_x r_{x,j}}} \leq \frac{tol_x}{N_p} |M(\mathbf{u}^h)|.$$

Solving this inequality for the number of refinements  $r_{x,j}$  yields

$$r_{x,j} = \text{ceil} \left( \log \left( \frac{f_r |\eta_{x,j}| N_p}{tol_x |M(\mathbf{u}^h)|} \right) \frac{1}{\log(2^{s_x})} \right),$$

where  $\text{ceil}$  is the ceiling function, i.e.,  $\text{ceil}(\alpha)$  is the smallest integer greater than or equal to  $\alpha$ . The mesh size is then refined  $r_{x,j}$  times for every pipe  $j$  for which (3.2) holds; see Algorithm 1, lines 4, 5. The procedure for the computation of the number of temporal discretisation refinements is analogous; see Algorithm 1, lines 7, 8. The gas network is simulated again with the refined spatial and temporal stepsizes and the new error distributions  $\eta_m, \eta_x, \eta_t$  are computed (Algorithm 1, line 11). Subsequently, if the relative network error still exceeds the tolerance, i.e., if (2.6) is not satisfied, then the model of every pipe  $j$  for which

$$|\eta_{m,j}| > \frac{tol_m}{N_p} |M(\mathbf{u}^h)|$$

holds is refined to the next model higher up in the hierarchy (Algorithm 1, lines 12, 14, 15). The simulation errors are then re-evaluated again and this cycle is repeated until (2.6) is satisfied (Algorithm 1, lines 19, 2). This strategy is very similar to the one discussed in [17, Section 3.3]. The only differences are that the temporal error and tolerance are there considered for the entire network instead of the individual pipes.

**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, which we split in a network controller (see Algorithm 2a) and pipe level computations; see Algorithm 2b. For every pipe  $j$ , the estimated error reductions due to a single refinement in

the model  $\Delta\eta_{m,j}$  and in the space and time discretisations  $\Delta\eta_{x,j}$ ,  $\Delta\eta_{t,j}$  are given by

$$\begin{aligned}
 (3.3a) \quad \Delta\eta_{m,j} &= F_m(m_{c,j}, m_{c,j} - 1) |\eta_{m,j}| + |\eta_{x,j}(m_{c,j}, r_{x,j})| - |\eta_{x,j}(m_{c,j} - 1, r_{x,j})| \\
 &\quad + |\eta_{t,j}(m_{c,j}, r_{t,j})| - |\eta_{t,j}(m_{c,j} - 1, r_{t,j})| \\
 &= F_m(m_{c,j}, m_{c,j} - 1) |\eta_{m,j}| \\
 &\quad + (F_x(m_{c,j}) - F_x(m_{c,j} - 1)) (1 + (f_r - 1) \text{sign}(r_{x,j})) \frac{|\eta_{x,j}|}{2^{s_x r_{x,j}}} \\
 &\quad + (F_t(m_{c,j}) - F_t(m_{c,j} - 1)) (1 + (f_r - 1) \text{sign}(r_{t,j})) \frac{|\eta_{t,j}|}{2^{s_t r_{t,j}}},
 \end{aligned}$$

$$\begin{aligned}
 (3.3b) \quad \Delta\eta_{x,j} &= |\eta_{x,j}(m_{c,j}, r_{x,j})| - |\eta_{x,j}(m_{c,j}, r_{x,j} + 1)| \\
 &= F_x(m_{c,j}) (1 + (f_r - 1) \text{sign}(r_{x,j}) - 2^{-s_x} f_r) \frac{|\eta_{x,j}|}{2^{s_x r_{x,j}}},
 \end{aligned}$$

$$\begin{aligned}
 (3.3c) \quad \Delta\eta_{t,j} &= |\eta_{t,j}(m_{c,j}, r_{t,j})| - |\eta_{t,j}(m_{c,j}, r_{t,j} + 1)| \\
 &= F_t(m_{c,j}) (1 + (f_r - 1) \text{sign}(r_{t,j}) - 2^{-s_t} f_r) \frac{|\eta_{t,j}|}{2^{s_t r_{t,j}}},
 \end{aligned}$$

where  $\eta_{x,j}$ ,  $\eta_{t,j}$ ,  $\eta_{m,j}$  denote the error estimators for the initial discretisation stepsizes and model distribution and the functions

$$\begin{aligned}
 \eta_{x,j}(m_{c,j}, r_{x,j}) &= F_x(m_{c,j}) (1 + (f_r - 1) \text{sign}(r_{x,j})) \frac{\eta_{x,j}}{2^{s_x r_{x,j}}}, \\
 \eta_{t,j}(m_{c,j}, r_{t,j}) &= F_t(m_{c,j}) (1 + (f_r - 1) \text{sign}(r_{t,j})) \frac{\eta_{t,j}}{2^{s_t r_{t,j}}},
 \end{aligned}$$

estimate the new discretisation errors after model and/or grid refinements. Here, the factor  $F_m(a, b)$  denotes an error reduction factor for the model error when models are shifted from  $a$  to  $b$ . The spatial and temporal discretisation errors also depend on the simulation model. The factors  $F_x(m_{c,j})$ ,  $F_t(m_{c,j})$  denote error amplification factors for the discretisation errors which account for this model dependency, where  $m_{c,j} = 1, 2, 3$  denotes the current model of pipe  $j$  and refers to models M1, M2, 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 benchmark

---

**Algorithm 2a** : Maximal error refinement

---

**Input:**  $m, \eta_m, \eta_x, \eta_t, \text{tol}, s_x, s_t, \phi$

```

1: for  $j = 1, \dots, N_p$  do
2:    $b_j, z_j \leftarrow \text{MAXERRORRED}(m_j, r_{m,j}, r_{x,j}, r_{t,j}, \eta_{m,j}, \eta_{x,j}, \eta_{t,j}, s_x, s_t)$ 
3: end for
4: while  $\text{networkError} > \text{tol} |M(\mathbf{u}^h)|$  do
5:    $\text{bound} \leftarrow \phi \cdot \max_j b_j$ 
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:        $b_j, z_j \leftarrow \text{MAXERRORRED}(m_j, r_{m,j}, r_{x,j}, r_{t,j}, \eta_{m,j}, \eta_{x,j}, \eta_{t,j}, s_x, s_t)$ 
10:    end if
11:  end for
12:  update  $\text{networkError}$ 
13: end while

```

**Output:**  $r_m, r_x, r_t$

---

model **M2**. Note that in determining  $\Delta\eta_{m,j}$  in (3.3a) we also consider changes in the spatial and temporal discretisation errors since the central idea is to account for a net error reduction. The approximate error reductions  $\Delta\eta_{m,j}, \Delta\eta_{x,j}, \Delta\eta_{t,j}$  are computed in Algorithm 2b, lines 4, 8, 9. Then, the best option

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

is passed to the network; see Algorithm 2b, line 10. There, the notation  $[b, z] = \max\{\cdot\}$  is similar to MATLAB notation where  $b$  denotes the maximal element and  $z$  is the corresponding index. On the network level, we mark those pipes for refinement for which the error reductions are larger than  $\phi \cdot \max_j b_j$ , with  $\phi \leq 1$  (Algorithm 2a, lines 5, 7). The numbers of refinements of the selected pipes are increased by one and the best options of these pipes are updated (Algorithm 2a, lines 8, 9). Then, the total network error is updated (Algorithm 2a, line 12). This can be done either by simulating the gas network again with the refined models and discretisations and computing the new error estimates in (2.5) or by using the error reduction estimates  $\Delta\eta$  in (3.3) to update the error estimators via

$$(3.4) \quad \eta(r+1) = \eta(r) - \text{sign}(\eta(r))\Delta\eta,$$

with  $r$  the number of refinements. This iteration is repeated until the relative absolute network error is brought below the tolerance (Algorithm 2a, line 4).

**Strategy 3: Maximal error-to-cost refinement (S3).** The adaptive refinements are made with an objective of reducing the computational 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 strategies viewed in the optimization setting (1.1). In this strategy we also take into account the computational costs that are incurred by the refinements. Strategy 3, given in Algorithm 3a for the network level and Algorithm 3b for the pipe level, is identical to Strategy 2 on the network level. On the pipe level, however, we also compute the cost additions  $\Delta c_{m,j}, \Delta c_{x,j}, \Delta c_{t,j}$  using a cost functional  $F_c(m_c, r_x, r_t)$  for each of the corresponding error reductions  $\Delta\eta_{m,j}, \Delta\eta_{x,j}, \Delta\eta_{t,j}$ ; see Algorithm 3b, lines 11, 13, 14. The error controller on the pipe level passes the best option

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

---

**Algorithm 2b** : Maximal pipe error reduction

---

```

1: function MAXERRORRED( $m, r_m, r_x, r_t, \eta_m, \eta_x, \eta_t, s_x, s_t$ )
2:    $m_c \leftarrow m - r_m$ 
3:   if  $m_c \neq 1$  then
4:     Compute  $\Delta\eta_m$  as in (3.3a)
5:   else
6:      $\Delta\eta_m \leftarrow 0$ 
7:   end if
8:   Compute  $\Delta\eta_x$  as in (3.3b)
9:   Compute  $\Delta\eta_t$  as in (3.3c)
10:   $[b, z] = \max\{\Delta\eta_m, \Delta\eta_x, \Delta\eta_t\}$ 
11:  return  $b, z$ 
12: end function

```

---

i.e., the maximal error-to-cost ratio, to the network (Algorithm 3b, line 15). The idea of this strategy is similar to the greedy approximation algorithm for solving the unbounded knapsack problem, see [16].

For the experiments and simulations performed in this paper we compute the computational cost per pipe in CPU seconds using a cost functional of the form

$$(3.5) \quad F(m, n_x, n_t) = C_m n_x^{\alpha_m} n_t^{\beta_m},$$

where  $n_x, n_t$  denote the number of nodes in space and time,  $m \in \{1, 2, 3\}$  denotes the model, and  $C_m, \alpha_m, \beta_m$  are model-dependent constants. These constants are determined by the method of least squares. For this, gas flow simulations through a single pipe are performed using the software *ANACONDA* (cf. [31, 32]) with many different values of  $n_x$  and  $n_t$ , which return the corresponding computational cost values  $F$ . The constants are given in Table 3.1. We note that although no discretisation is required for the algebraic model M3, still a mesh is used in this model to determine the evaluation points. These evaluation points are used for the computation of the model error estimator  $\eta_{m,j}^{\text{ALG-LIN}}$ ; cf. Section 2.2. We can rewrite the functional (3.5) in terms of refinements, which is needed in Algorithm 3b, assuming that the initial number of nodes  $n_{x,0}, n_{t,0}$  are known. Then we get the cost functional

$$(3.6) \quad F_c(m, r_x, r_t) = C_m (2^{r_x} n_{x,0})^{\alpha_m} (2^{r_t} n_{t,0})^{\beta_m}.$$

Having defined two new greedy-like refinement Strategies S2 and S3, in the next section we compare their performance with the existing Strategy S1.

TABLE 3.1  
Cost functional constants in (3.5), (3.6).

$m$	$C_m$	$\alpha_m$	$\beta_m$
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

---

**Algorithm 3a** : Maximal error-to-cost refinement

---

**Input:**  $m, \eta_m, \eta_x, \eta_t, tol, s_x, s_t, \phi$

Same as Algorithm 2a, replacing lines 2, 9 with

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

**Output:**  $r_m, r_x, r_t$

---



---

**Algorithm 3b** : Maximal pipe error-to-cost ratio

---

1: **function** MAXERRORTOCOSTRATIO( $m, r_m, r_x, r_t, \eta_m, \eta_x, \eta_t, s_x, s_t$ )

Lines 2-9 same as Algorithm 2b

10: **if**  $m_c \neq 1$  **then**  
 11:      $\Delta c_m \leftarrow F_c(m_c - 1, r_x, r_t) - F_c(m_c, r_x, r_t)$   
 12: **end if**  
 13:  $\Delta c_x \leftarrow F_c(m_c, r_x + 1, r_t) - F_c(m_c, r_x, r_t)$   
 14:  $\Delta c_t \leftarrow F_c(m_c, r_x, r_t + 1) - F_c(m_c, r_x, r_t)$   
 15:  $[b, z] = \max\{\Delta \eta_m / \Delta c_m, \Delta \eta_x / \Delta c_x, \Delta \eta_t / \Delta c_t\}$   
 16: **return**  $b, z$   
 17: **end function**

---

**4. Numerical results.** In this section we numerically test the performance of the three refinement strategies given in Section 3. A synthetic experiment on an abstract gas network is conducted in Section 4.1 and an application of the algorithms to the numerical simulation of a realistic gas network is presented in Section 4.2.

**4.1. Synthetic experiment.** Let us consider an abstract gas network that is given by its number of pipes  $N_p$ . On this network we perform a synthetic experiment in order to test and compare the three refinement strategies given in Section 3. The experiment is conducted by first drawing random samples of initial discretisation and model errors  $\eta_m, \eta_x, \eta_t \in \mathbb{R}^{N_p}$  and initial numbers of discretisation nodes  $n_x, n_t \in \mathbb{N}^{N_p}$  from probability distributions. Then, for every sample, the three algorithms are executed using the approximate error reductions in (3.3) to update the errors. The computational simulation cost after applying the resulting refinement scheme is computed using the functional in (3.6) for every sample and strategy. Finally, the mean of the computational cost values over all samples is computed for every strategy. Each error sample represents the evaluation of the error estimators in (2.5) on a specific but unknown network configuration of  $N_p$  pipes. Thus, the abstract network setting enables us to consider many possible pipe network configurations by drawing a large number of error samples. It also enables us to execute the three algorithms a large number of times with a low computational cost since the need for actual gas flow simulations within the network is circumvented.

We test the performance of the three refinement strategies on  $10^4$  random samples of error distributions  $\eta_m, \eta_x, \eta_t \in \mathbb{R}^{12}$  in a network of  $N_p = 12$  pipes. The initial number of space and time discretisation nodes  $n_{x,j}, n_{t,j}$  are randomly chosen from the interval  $[100, 200]$  for every pipe  $j$ . 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; see (3.3a). 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 drawn from the distribution  $\mathcal{U}[0, 1]$ , where  $\mathcal{U}[a, b]$  denotes a uniform probability distribution on the interval  $[a, b]$ . The initial spatial and temporal discretisation errors are drawn from the distribution  $\mathcal{U}[0, 0.2]$ . The model and discretisation error estimators  $\eta$  after increasing the number of refinements  $r$  by one are given in (3.4), where  $\Delta\eta$  is given in (3.3) with  $F_m(3, 2) = 3/4$ ,  $F_m(2, 1) = 1/4$ . We choose  $s_x = 2$ ,  $s_t = 1$ , since these are the convergence orders of the implicit box scheme in (2.1). The parameter  $\kappa = 1/3$  for Strategy 1 is chosen such that all three error types have an equal fraction of the network 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 with a relative error tolerance of  $tol = 10^{-1}$  with a target functional value  $M(\mathbf{u}^h) = 2.5 \cdot N_p = 30$ . Hence, we require for the total network simulation error that

$$\left| \sum_{j \in \mathcal{J}_p} (\eta_{m,j} + \eta_{x,j} + \eta_{t,j}) \right| < tol |M(\mathbf{u}^h)| = 3.$$

Each strategy returns a refinement scheme which brings the simulation error below the network tolerance. The goal is to have a low computational cost. The mean of the total computational cost values in CPU seconds over  $10^4$  samples is displayed in Table 4.1 for every refinement strategy. We also provide the percentage savings in the 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  $\phi = 0.8, 0.9, 1$ , 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 best for both S2 and S3.

In this experiment we find that 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 4.1

Mean total computational 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.

Strategy	Mean CPU time [s]	Savings
S1	36.1	-
S2 <sub>1</sub>	8.30	77.0 %
S2 <sub>2</sub>	8.04	77.7 %
S2 <sub>3</sub>	7.72	78.6 %
S3 <sub>1</sub>	7.64	78.8 %
S3 <sub>2</sub>	7.39	79.5 %
S3 <sub>3</sub>	7.12	80.3 %

**4.2. 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 Figure 4.1. The considered network consists of twelve pipes (P01–P12, with lengths between 30 km and 100 km), two sources (S01–S02), four consumers (C01–C04), three compressor stations (Comp01–Comp03) and 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 4 hours. 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; see Section 2.2. The simulation is performed using the software *ANACONDA*; cf. [31, 32].

REMARK 4.1. For the strategies proposed in Section 3, the temporal error  $\eta_t = \sum_{j \in \mathcal{J}_p} \eta_{t,j}$ , cf. (2.4), is considered individually for each pipe. In the implementation of *ANACONDA*, however, it is only computed globally. In order to get a local temporal error and hence to fit into the setting, we distribute the temporal error uniformly among the pipes. If the time step size has to be refined in a single pipe following one of the strategies above, then it will be refined globally and the temporal error estimators of all pipes are updated.

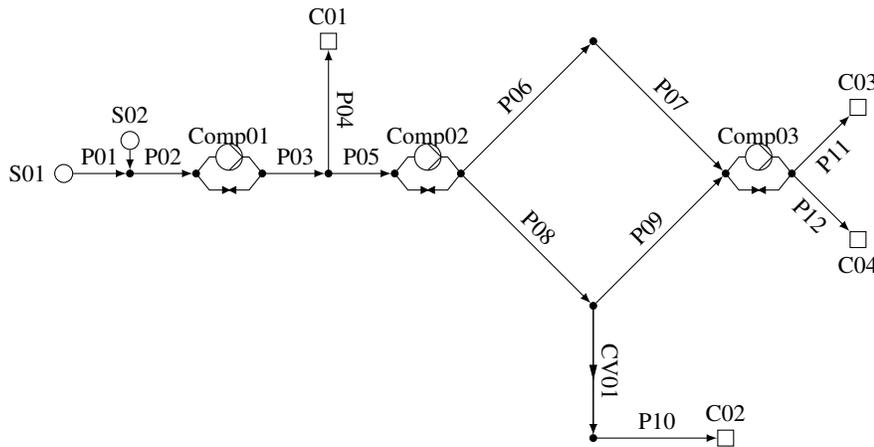


FIG. 4.1. Gas supply network with compressor stations and a control valve.

TABLE 4.2

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

Strategy	Relative error	CPU time [s]	DOF-ST	M1/M2/M3 [%]	Savings
S1	$1.384 \cdot 10^{-5}$	7.53	49 664	100 / 0 / 0	73.3 %
S2 <sub>1</sub>	$3.730 \cdot 10^{-5}$	2.42	11 264	92 / 0 / 8	91.4 %
S2 <sub>2</sub>	$3.091 \cdot 10^{-5}$	2.27	10 496	92 / 0 / 8	92.0 %
S2 <sub>3</sub>	$3.091 \cdot 10^{-5}$	2.29	10 496	92 / 0 / 8	91.9 %
S3 <sub>1</sub>	$5.076 \cdot 10^{-5}$	3.12	11 684	83 / 17 / 0	88.9 %
S3 <sub>2</sub>	$5.190 \cdot 10^{-5}$	2.83	12 928	100 / 0 / 0	90.0 %
S3 <sub>3</sub>	$5.098 \cdot 10^{-5}$	2.83	11 684	92 / 8 / 0	90.0 %
conventional	$4.573 \cdot 10^{-5}$	28.2	358 848	100 / 0 / 0	-
reference	-	1013	23 923 200		

A reference solution was computed using model **M1** and a very fine spatial-temporal discretisation with approximately 24 million degrees of freedom in space and time (DOF-ST). The simulation was performed using the strategies from Section 3 with a relative tolerance of  $tol = 10^{-4}$  and with a conventional modelling using the most accurate model and a uniform space and time discretisation with 360 000 DOF-ST. Note that with the conventional modelling, we did not estimate any model or discretisation errors and hence did not gain any information about the accuracy of the solution. Table 4.2 shows the relative errors of the simulations compared to the reference solution, the total CPU time, the degrees of freedom in space and time (DOF-ST), the models used (in percent), and the percentage savings of the Strategies S1 to S3 in relation to the conventional modelling.

We see that the relative error of the Strategies S1 to S3 are in a similar range as of the simulation using the conventional modelling. However, due to the adaptive error control, the DOF-ST are reduced significantly. We see that the results of Strategies S2 and S3 are in a similar range and that these strategies gain an additional saving of 15 % to 20 % as compared to Strategy S1. Regarding the synthetic experiment in Section 4.1, this means that the savings with respect to S1 are in the range of 60 % to 70 %. 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 tolerances* Strategy S1.

**5. Conclusions.** In this paper we address the problem of automatic error control for large scale 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 refinement strategy is a generalisation of the knapsack problem. We present three strategies for adaptive simulation error control via spatial and temporal discretisation and model refinements. The strategy *individual tolerances*, which is currently implemented in *ANACONDA*, sets a uniform tolerance 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 *maximal error-to-cost refinement* also accounts for the increase in computational cost inflicted by the refinement.

We construct a synthetic experiment to test the three strategies on many different network configurations. From this experiment we find that the two greedy-like strategies significantly reduce the computational cost as compared to the *individual tolerances* strategy. This result is

largely reflected in an actual gas flow simulation using *ANACONDA* for a 12 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 greedy-like refinement strategies lead to considerable computational savings without compromising on the simulation accuracy.

## REFERENCES

- [1] M. A. ADEWUMI AND J. ZHOU, *Simulation of transient flow in natural gas pipelines*, in 27th Annual Meeting of PSIG Albuquerque, 9508 PSIG Conference Paper, OnePetro, Richardson, 1995.
- [2] M. K. BANDA, M. HERTY, AND A. KLAR, *Coupling conditions for gas networks governed by the isothermal Euler equations*, *Netw. Heterog. Media*, 1 (2006), pp. 295–314.
- [3] ———, *Gas flow in pipeline networks*, *Netw. Heterog. Media*, 1 (2006), pp. 41–56.
- [4] R. BECKER AND R. RANNACHER, *An optimal control approach to a posteriori error estimation in finite element methods*, *Acta Numer.*, 10 (2001), pp. 1–102.
- [5] J. BELL, M. BERGER, J. SALTZMAN, AND M. WELCOME, *Three-dimensional adaptive mesh refinement for hyperbolic conservation laws*, *SIAM J. Sci. Comput.*, 15 (1994), pp. 127–138.
- [6] M. BERGER AND P. COLELLA, *Local adaptive mesh refinement for shock hydrodynamics*, *J. Comput. Phys.*, 82 (1989), pp. 64–84.
- [7] M. BERGER AND A. JAMESON, *Automatic adaptive grid refinement for the Euler equations*, *AIAA J.*, 23 (1985), pp. 561–568.
- [8] M. J. BERGER AND J. OLIGER, *Adaptive mesh refinement for hyperbolic partial differential equations*, *J. Comput. Phys.*, 53 (1984), pp. 484–512.
- [9] M. BESIER AND R. RANNACHER, *Goal-oriented space-time adaptivity in the finite element Galerkin method for the computation of nonstationary incompressible flow*, *Internat. J. Numer. Methods Fluids*, 70 (2012), pp. 1139–1166.
- [10] M. BRAACK AND A. ERN, *A posteriori control of modeling errors and discretization errors*, *Multiscale Model. Simul.*, 1 (2003), pp. 221–238.
- [11] S. C. BRENNER AND L. R. SCOTT, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Springer, New York, 2008.
- [12] C. CARSTENSEN AND R. H. W. HOPPE, *Convergence analysis of an adaptive nonconforming finite element method*, *Numer. Math.*, 103 (2006), pp. 251–266.
- [13] ———, *Error reduction and convergence for an adaptive mixed finite element method*, *Math. Comp.*, 75 (2006), pp. 1033–1042.
- [14] K. S. CHAPMAN, P. KRISHNISWAMI, V. WALLENTINE, M. ABBASPOUR, R. RANGANATHAN, R. ADDANKI, J. SENGUPTA, AND L. CHEN, *Virtual pipeline system testbed to optimize the U.S. natural gas transmission pipeline system*, Tech. Report DE-FC26-01NT41322, The National Gas Machinery Laboratory, Kansas State University, Kansas 2005.
- [15] R. M. COLOMBO AND M. GARAVELLO, *A well posed Riemann problem for the  $p$ -system at a junction*, *Netw. Heterog. Media*, 1 (2006), pp. 495–511.
- [16] G. B. DANTZIG, *Discrete-variable extremum problems*, *Operations Res.*, 5 (1957), pp. 266–277.
- [17] P. DOMSCHKE, *Adjoint-Based Control of Model and Discretization Errors for Gas Transport in Networked Pipelines*, PhD. Thesis, Dept. of Math., TU Darmstadt, Darmstadt, 2011.
- [18] P. DOMSCHKE, B. GEISLER, O. KOLB, J. LANG, A. MARTIN, AND A. MORSI, *Combination of nonlinear and linear optimization of transient gas networks*, *INFORMS J. Comput.*, 23 (2011), pp. 605–617.
- [19] P. DOMSCHKE, O. KOLB, AND J. LANG, *An adaptive model switching and discretization algorithm for gas flow on networks*, *Procedia Comput. Sci.*, 1 (2010), pp. 1325 – 1334.
- [20] ———, *Adjoint-based control of model and discretisation errors for gas and water supply networks*, in *Computational Optimization and Applications in Engineering and Industry*, X.-S. Yang and S. Koziel, eds., *Studies in Computational Intelligence*, 359, Springer, Berlin, 2011, pp. 1–17.
- [21] ———, *Adjoint-based control of model and discretisation errors for gas flow in networks*, *Int. J. Math. Model. Numer. Optim.*, 2 (2011), pp. 175–193.
- [22] ———, *Adjoint-based error control for the simulation and optimization of gas and water supply networks*, *Appl. Math. Comput.*, 259 (2015), pp. 1003–1018.
- [23] W. DÖRFLER, *A convergent adaptive algorithm for Poisson’s equation*, *SIAM J. Numer. Anal.*, 33 (1996), pp. 1106–1124.
- [24] W. DÖRFLER AND M. RUMPF, *An adaptive strategy for elliptic problems including a posteriori controlled boundary approximation*, *Math. Comp.*, 67 (1998), pp. 1361–1382.
- [25] K. EHRHARDT AND M. C. STEINBACH, *KKT systems in operative planning for gas distribution networks*, *PAMM Proc. Appl. Math. Mech.*, 4 (2004), pp. 606–607.

- [26] ———, *Nonlinear optimization in gas networks*, in Modeling, Simulation and Optimization of Complex Processes, H. G. Bock, E. Kostina, H. X. Phu, and R. Ranacher, eds., Springer, Berlin, 2005, pp. 139–148.
- [27] M. HERTY, *Modeling, simulation and optimization of gas networks with compressors*, Netw. Heterog. Media, 2 (2007), pp. 81–97.
- [28] M. HERTY, J. MOHRING, AND V. SACHERS, *A new model for gas flow in pipe networks*, Math. Methods Appl. Sci., 33 (2010), pp. 845–855.
- [29] S. L. KE AND H. C. TI, *Transient analysis of isothermal gas flow in pipeline networks*, Chem. Engrg. J., 76 (2000), pp. 169–177.
- [30] H. KELLERER, U. PFERSCHY, AND D. PISINGER, *Knapsack Problems*, Springer, Berlin, 2004.
- [31] O. KOLB, *Simulation and Optimization of Gas and Water Supply Networks*, PhD. Thesis, Dept. of Math., TU Darmstadt, Darmstadt, 2011.
- [32] O. KOLB, J. LANG, AND P. BALES, *An implicit box scheme for subsonic compressible flow with dissipative source term*, Numer. Algorithms, 53 (2010), pp. 293–307.
- [33] R. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.
- [34] A. MARTIN, M. MÖLLER, AND S. MORITZ, *Mixed integer models for the stationary case of gas network optimization*, Math. Program., 105 (2006), pp. 563–582.
- [35] J. T. ODEN, S. PRUDHOMME, A. ROMKES, AND P. T. BAUMAN, *Multiscale modeling of physical phenomena: adaptive control of models*, SIAM J. Sci. Comput., 28 (2006), pp. 2359–2389.
- [36] A. OSIADACZ, *Different transient flow models – limitations, advantages, and disadvantages*, in 28th Annual Meeting of PSIG San Francisco, 9606 PSIG Conference Paper, OnePetro, Richardson, 1996.
- [37] A. J. OSIADACZ AND M. CHACZYKOWSKI, *Comparison of isothermal and non-isothermal pipeline gas flow models*, Chem. Engrg. J., 81 (2001), pp. 41–51.
- [38] M. C. STEINBACH, *On PDE solution in transient optimization of gas networks*, J. Comput. Appl. Math., 203 (2007), pp. 345–361.
- [39] B. WENDROFF, *On centered difference equations for hyperbolic systems*, J. Soc. Indust. Appl. Math., 8 (1960), pp. 549–555.