

EFFICIENT NUMERICAL METHODS FOR GAS NETWORK MODELING AND SIMULATION

YUE QIU*

School of Information Science and Technology
ShanghaiTech University
393 Middle Huaxia Road, 201210, Shanghai, China

Computational Methods for Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstraße 1, 39106, Magdeburg, Germany

SARA GRUNDEL

Computational Methods for Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstraße 1, 39106, Magdeburg, Germany

MARTIN STOLL

Faculty of Mathematics
Technische Universität Chemnitz
Reichenhainer Straße 41, 09126, Chemnitz, Germany

PETER BENNER

Computational Methods for Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstraße 1, 39106, Magdeburg, Germany

(Communicated by Pierre-Emmanuel Jabin)

ABSTRACT. We study the modeling and simulation of gas pipeline networks, with a focus on fast numerical methods for the simulation of transient dynamics. The obtained mathematical model of the underlying network is represented by a system of nonlinear differential algebraic equations (DAEs). With our modeling approach, we reduce the number of algebraic constraints, which correspond to the $(2, 2)$ block in our semi-explicit DAE model, to the order of junction nodes in the network, where a junction node couples at least three pipelines. We can furthermore ensure that the $(1, 1)$ block of all system matrices including the Jacobian is block lower triangular by using a specific ordering of the pipes of the network. We then exploit this structure to propose an efficient preconditioner for the fast simulation of the network. We test our numerical methods on benchmark problems of (well-)known gas networks and the numerical results show the efficiency of our methods.

2020 *Mathematics Subject Classification.* Primary: 65F08, 37M05, 37N30, 94C15.

Key words and phrases. Gas networks modeling, isothermal Euler equation, directed acyclic graph (DAG), differential algebraic equation (DAE), preconditioning.

This work is partially funded by the European Regional Development Fund (ERDF/EFRE: ZS/2016/04/78156) within the Center Dynamic Systems (CDS).

* Corresponding author: Yue Qiu.

1. Introduction. Natural gas is one of the most widely used energy sources in the world, as it is easily transportable, storable and usable to generate heat and electricity. Even though research on transient gas networks dates back to the 1980s [26, 27], often only stationary solutions of the gas network are computed. This is also reasonable as the variation in a classically operated gas transportation networks enforces no need for a truly transient simulation. However, as we move from classical to renewable energy sources in which we may use the gas pipelines to deal with flexibility from volatile energy creation, the need for fast transient simulation will increase. In recent years, research on natural gas networks focuses on a variety of topics: transient simulations [26, 27, 39, 14, 17], optimization and control [37, 45, 18], time splitting schemes for solving the parabolic flow equations [44], discretization methods [8, 28], and model sensitivity [6] to mention a few. It is obvious that efficient simulation techniques are needed both for design and for control.

The objective of this paper is to speed up the computations at the heart of each simulation. For that we make use of a discretization that respects the hyperbolic nature of the problem by using a finite volume method (FVM), as well as exploiting the network structure to create good properties for the computations. The use of the finite volume method is not entirely new in the literature [5], but typically comes with a Riemann solver, which by definition is an explicit time integration and therefore often slow. It is also not clear how to extend such a numerical scheme to a network with circles, which is straightforward in our approach. We start as it is standard for modeling gas transport in a pipeline by the one-dimensional isothermal Euler equation, which is a coupled partial differential equation (PDE). We introduce the necessary discrete variables for the pressure and the flux on each pipe and make sure to use as little algebraic equations as possible when considering a network of pipes. Other discretization schemes were studied in [3, 23], but our setup is favorable for the speedup of the DAE simulation with an implicit time integration scheme. By further exploiting the structure of the system, we propose a preconditioner that enables fast solution of such a nonlinear equation using a preconditioned Krylov solver at each Newton iteration.

The structure of this paper is as follows. We introduce the incompressible isothermal Euler equation for gas dynamics modeling of each pipeline of the network in Section 2, where we also apply the finite volume method (FVM) to discretize the incompressible isothermal Euler equation. In Section 3, we introduce the details of gas network modeling starting from assembling all pipelines. This results in a set of nonlinear DAEs for the network model. We propose numerical algorithms to solve the resulting nonlinear DAE in Section 4 to simulate the gas network. We use benchmark problems from gas pipeline networks to show the efficiency and the advantage of our numerical algorithms in Section 5, and we draw conclusions in the last section.

2. Gas dynamics in pipelines. In a typical gas transport network, the main components are pipelines (or pipes, for short). In this section, we will discuss the dynamics of gas transported along pipes. In Section 2.1, we discuss the isothermal Euler equation, which is the partial differential equation used for modeling transport in the pipe. Some standard simplifications lead to equation (4) and after some more assumptions, equation (5) can be derived, which is the basis for our simulations. In Section 2.2, equation (5) is discretized in space by a standard finite volume scheme leading to (7).

2.1. 1D isothermal Euler equation. The dynamics of gas transported along pipes is described by the Euler equation, which represents the laws of mass conservation, momentum conservation, and energy conservation. In this paper, we assume that the temperature is constant throughout the gas network, leading to the isothermal Euler equations. Therefore, the energy equation can be neglected. This may seem unrealistic, but for onshore gas networks, in which the pipes are buried underground, the temperature along pipes does not vary much. This assumption greatly reduces the complexity of modeling and is widely used in the simulation of gas networks [19, 22, 13, 17, 11].

Consider the 1D isothermal Euler equation over the spatial domain $[0, L]$ given by

$$\frac{\partial}{\partial t} \rho = -\frac{\partial}{\partial x} \varphi, \quad (1a)$$

$$\frac{\partial}{\partial t} \varphi = -\frac{\partial}{\partial x} p - \frac{\partial}{\partial x} (\rho v^2) - g\rho \frac{\partial}{\partial x} h - \frac{\lambda(\varphi)}{2d} \rho v |v|, \quad (1b)$$

$$p = \gamma(T) z(p, T) \rho. \quad (1c)$$

Here, ρ is the density of the gas (kg/m^3), φ represents the flow rate $\varphi = \rho v$ with v the velocity of the gas (m/s), d is the diameter of the pipe (m), λ is the friction factor of the gas, and g is the gravity constant. Meanwhile, p denotes the pressure of the gas (N/m^2), T is the temperature of the gas (K), and z denotes the compressibility factor. The conservation of mass is given by (1a), and the conservation of momentum is represented by (1b), while the state equation (1c) couples the pressure with the density.

By using the mass flow $q = a\varphi$ to substitute into (1a)–(1b), where a is the cross-section area of the considered pipes, we get

$$\frac{\partial}{\partial t} \rho = -\frac{1}{a} \frac{\partial}{\partial x} q, \quad (2a)$$

$$\frac{1}{a} \frac{\partial}{\partial t} q = -\frac{\partial}{\partial x} p - \frac{1}{a^2} \frac{\partial}{\partial x} \frac{q^2}{\rho} - g\rho \frac{\partial}{\partial x} h - \frac{\lambda(q)}{2da^2} \frac{q|q|}{\rho}, \quad (2b)$$

$$p = \gamma(T) z(p, T) \rho. \quad (2c)$$

For the isothermal case, the temperature T equals T_0 throughout the network, hence $\gamma(T) = \gamma(T_0) = \gamma_0$, and $z(p, T) = z(p, T_0) = z_0(p)$. Therefore, the compressibility factor $z(p, T)$ is only related to the pressure p and we can rewrite (2a)–(2c) as

$$\frac{1}{\gamma_0} \frac{\partial}{\partial t} \frac{p}{z_0(p)} = -\frac{1}{a} \frac{\partial}{\partial x} q, \quad (3a)$$

$$\frac{1}{a} \frac{\partial}{\partial t} q = -\frac{\partial}{\partial x} p - \underbrace{\frac{\gamma_0}{a^2} \frac{\partial}{\partial x} \frac{q^2 z_0(p)}{p}}_{\text{inertia term}} - \underbrace{\frac{g}{\gamma_0} \frac{p}{z_0(p)} \frac{\partial}{\partial x} h}_{\text{gravity term}} - \underbrace{\frac{\lambda(q) \gamma_0}{2da^2} z_0(p) \frac{q|q|}{p}}_{\text{friction term}}. \quad (3b)$$

For the inertia term, it is concluded in [19] that for realistic gas pipes

$$\frac{\gamma_0}{a^2} \frac{\partial}{\partial x} \frac{q^2 z_0(p)}{p} \approx 10^{-3} \frac{\partial}{\partial x} p.$$

Therefore, the inertia term can be neglected and this neglecting greatly simplifies the model, which is standard in the study of gas networks [18, 14, 45]. In this paper, we also use this simplification. Meanwhile, we will often assume that the elevation of pipes is homogeneous. The gravity term in (3b) then vanishes. However, this

term is easily treatable within our framework, which we will illustrate later in this section.

Now, we get the model that describes the dynamics of isothermal gas transported along homogeneous elevation pipes given by

$$\frac{\partial}{\partial t} \frac{p}{z_0(p)} = -\frac{\gamma_0}{a} \frac{\partial}{\partial x} q, \quad (4a)$$

$$\frac{\partial}{\partial t} q = -a \frac{\partial}{\partial x} p - \frac{\lambda(q)\gamma_0}{2da} z_0(p) \frac{q|q|}{p}. \quad (4b)$$

The details of modeling the compressibility factor $z_0(p)$ and the friction factor $\lambda(q)$ are described in [3]. In this paper, we only consider incompressible gas, i.e., $z_0(p) = 1$. For compressible gas dynamics, the compressibility factor $z_0(p)$ can often be represented by a polynomial of the pressure p [3], which in turn makes (4a)–(4b) a highly nonlinear transport problem and requires further investigation.

2.2. Finite volume discretization. In this paper, the dynamics of the gas transported along pipes are described by the 1D isothermal incompressible Euler equation ($z_0(p) \equiv 1$) over the spatial domain $[0, L]$ with homogeneous elevation. According to (4a)–(4b) we have

$$\frac{\partial}{\partial t} p + \frac{c}{a} \frac{\partial}{\partial x} q = 0, \quad (5a)$$

$$\frac{\partial}{\partial t} q + a \frac{\partial}{\partial x} p + \frac{c\lambda}{2da} \frac{q|q|}{p} = 0. \quad (5b)$$

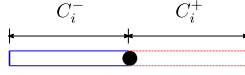
For simplification of notation we introduce $c = \gamma_0$ and we also assume $\lambda(q) \equiv \lambda$. The system (5a)–(5b) is nonlinear due to the friction term. For gas transportation pipes, the boundary condition at the inflow point $x = 0$ is given by the prescribed pressure p_s , while the boundary condition at the outflow point $x = L$ is represented by the given mass flow (gas demand) q_d . Therefore, the boundary conditions for (5a)–(5b) are given as

$$\begin{cases} p = p_s, & \text{at } x = 0, \\ q = q_d, & \text{at } x = L. \end{cases} \quad (6)$$

For the well-posedness and the regularity of the solution of the system (5)–(6), we refer to [9].

More advanced numerical schemes for hyperbolic PDEs, such as the total variation diminishing (TVD) method [40], or the discontinuous Galerkin method (DG) [24], could be implemented at the next step of our research to investigate more complicated dynamics of the gas networks. The scope of our paper is to develop a systematic numerical methodology for the fast simulation of the network dynamics while taking numerical accuracy into account. Thus, here we stick to this simple discretization scheme for the purpose of exposition.

To apply the FVM to discretize the PDE and the boundary condition, we integrate (5a)–(5b) over each control volume. The i -th control pressure control volume C_i is depicted in Figure 1. The discretization point in C_i is either a virtual node along a pipe or a real node that connects two different pipes. Therefore, the coefficient a of the PDE (5a)–(5b), which represents the cross-section area of a pipe, may have a sudden change at the node in control volume C_i . Here, we use C_i^- and C_i^+ to partition the control volume C_i with $C_i = C_i^- \cup C_i^+$, and the lengths of C_i^- and C_i^+ are $\frac{h_{i-1}}{2}$ and $\frac{h_i}{2}$, respectively.

FIGURE 1. Separation of control volume C_i

After the FVM discretization, we get

$$\underbrace{\begin{bmatrix} M_p & \\ & M_q \end{bmatrix}}_{\mathcal{M}} \underbrace{\begin{bmatrix} \partial_t p \\ \partial_t q \end{bmatrix}}_{\mathcal{K}} = \underbrace{\begin{bmatrix} 0 & K_{pq} \\ K_{qp} & 0 \end{bmatrix}}_{\mathcal{K}} \underbrace{\begin{bmatrix} p \\ q \end{bmatrix}}_{\mathcal{B}_q} + \underbrace{\begin{bmatrix} B_q \\ 0 \end{bmatrix}}_{\mathcal{B}_q} q_d + \underbrace{\begin{bmatrix} 0 \\ B_p \end{bmatrix}}_{\mathcal{B}_p} p_s + \begin{bmatrix} 0 \\ g(p_s, p, q) \end{bmatrix}, \quad (7)$$

where the mass matrices M_p and M_q are given by

$$M_p = \begin{bmatrix} \frac{h_1+h_2}{2} & & & & \\ & \frac{h_2+h_3}{2} & & & \\ & & \ddots & & \\ & & & \frac{h_{n-2}+h_{n-1}}{2} & \\ & & & \frac{h_{n-1}}{8} & \frac{3h_{n-1}}{8} \end{bmatrix}, \quad M_q = \begin{bmatrix} \frac{3h_1}{8} & \frac{h_1}{8} & & & \\ & \frac{h_1+h_2}{2} & & & \\ & & \frac{h_2+h_3}{2} & & \\ & & & \ddots & \\ & & & & \frac{h_{n-2}+h_{n-1}}{2} \end{bmatrix},$$

and

$$K_{pq} = -\frac{c}{2} \begin{bmatrix} -\frac{1}{a_1} & \frac{1}{a_1} - \frac{1}{a_2} & & & \\ & -\frac{1}{a_2} & \frac{1}{a_2} - \frac{1}{a_3} & & \\ & & \ddots & \ddots & \\ & & & -\frac{1}{a_{n-3}} & \frac{1}{a_{n-3}} - \frac{1}{a_{n-2}} \\ & & & & \frac{1}{a_{n-2}} - \frac{1}{a_{n-1}} \end{bmatrix},$$

is an upper-triangular matrix with 3 diagonals and

$$K_{qp} = -\frac{1}{2} \begin{bmatrix} a_1 & & & & \\ a_1 - a_2 & a_2 & & & \\ -a_2 & a_2 - a_3 & a_3 & & \\ & -a_3 & a_3 - a_4 & a_4 & \\ & & \ddots & \ddots & \ddots \\ & & & -a_{n-2} & a_{n-2} - a_{n-1} & a_{n-1} \end{bmatrix},$$

is a lower-triangular matrix with 3 diagonals. Meanwhile,

$$B_q = -\frac{c}{2} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{1}{a_{n-1}} \\ \frac{1}{a_{n-1}} \end{bmatrix}, \quad B_p = \frac{1}{2} \begin{bmatrix} a_1 \\ a_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$g(p_s, p, q) = -\frac{c}{4} \begin{bmatrix} \frac{h_1 \lambda_1}{a_1 d_1} \frac{q_1 |q_1|}{p_s} \\ \left(\frac{h_1 \lambda_1}{a_1 d_1} + \frac{h_2 \lambda_2}{a_2 d_2} \right) \frac{q_2 |q_2|}{p_2} \\ \left(\frac{h_2 \lambda_2}{a_2 d_2} + \frac{h_3 \lambda_3}{a_3 d_3} \right) \frac{q_3 |q_3|}{p_3} \\ \vdots \\ \left(\frac{h_{n-2} \lambda_{n-2}}{a_{n-2} d_{n-2}} + \frac{h_{n-1} \lambda_{n-1}}{a_{n-1} d_{n-1}} \right) \frac{q_{n-1} |q_{n-1}|}{p_{n-1}} \end{bmatrix}. \quad (8)$$

The vectors

$$p = [p_2 \ p_3 \ \cdots \ p_n]^T, \quad q = [q_1 \ q_2 \ \cdots \ q_{n-1}]^T,$$

represent the discretized analog of p and q to be computed, while $p_1 = p_s$, and $q_n = q_d$.

Remark 1. The obtained model in (7) results from the discretization of the incompressible isothermal Euler equation of homogeneous elevation (5a)–(5b). However, we note that for the heterogeneous elevation case, the gravity term in (3b) is linear in the pressure p . After discretization, this term will introduce an additional term in the position of K_{qp} in (7). This new term does not change the structure of the model that we obtained in (7). The structure we refer to here is the block structure of the matrices involved in describing the equation as well as the sparsity pattern of all Jacobians of the nonlinear functions.

Remark 2. In this discretization, the cross-sectional area of the pipe does not have to be constant. This will be needed later on in the modeling of the network.

Note that we impose the boundary condition (6) by using the prescribed pressure at the inflow point and the given mass flow at the outflow point. However one can also impose a negative mass flow at the outflow point making it an inflow point, and for more complex topology of the network with more than one supply node, it can happen that the gas flows out at a so called inflow point. Computational results in the numerical experiments show this.

3. Network modeling. We have introduced the single pipeline model for the gas dynamics in Section 2. In this section, we introduce the gas pipeline network using a graph in which the pipelines are the edges (Section 3.1). We simplify this graph by smoothing. This procedure is explained in detail for readers that are not familiar with this concept in Section 3.2. The dynamics of gas transport through this network are described by combining (7) on each edge with the nodal conditions on each node (Section 3.3), which are then put together into one large differential-algebraic system resulting in equation (13) in Section 3.4. The smoothing of the graph gets rid of internal nodes, which are not junction nodes and in turn cuts down the number of algebraic constraints for the global network model assembly so that the complexity of the global network model is reduced.

Within this paper, we focus on passive networks to demonstrate how advanced numerical linear algebra can benefit the fast simulation of such a network. When we say passive network we mean a network that does not contain active elements, such as compressors, valves, etc. [12]. This simplification allows us to maintain a differential algebraic model without combinatorial aspects. For the modeling of the network with compressors and valves we refer to [3, 20].

3.1. Pipeline network as graph. The abstract gas network is described by a directed graph

$$\mathcal{G} = (\mathcal{E}, \mathcal{N}), \tag{9}$$

where \mathcal{E} denotes the set of edges, which contains the pipes in the gas network. \mathcal{N} represents the set of nodes, which consist of the set of supply nodes \mathcal{N}_s , demand nodes \mathcal{N}_d , and interior nodes \mathcal{N}_0 of the network. Here, the supply nodes represent the set of nodes in the network where gas is injected into the network or more precisely where the pressure is given, and the demand nodes form a set of nodes where the gas is extracted, i.e., an outgoing flux is described, and interior nodes

account for the rest. We assume from now on that demand nodes and supply nodes are the only boundary nodes. That means they are only connected to one pipe. If supply or demand nodes exist that are connected to more than one edge, we add a short pipe to that node and declare the new node as the demand or supply node and the old one becomes an interior node. Sometimes interior nodes are called junction nodes [16], but for us junction nodes are more specific.

Definition 3.1. The nodes inside a graph \mathcal{G} , which connect at least three edges, are called *junction nodes*.

Starting from the network, we use the so called smoothing of a graph to end up with a smoothed graph in which all interior nodes are junction nodes.

3.2. Smoothed graph. Denote by $N_j \subset \mathcal{N}$ the set of junction nodes of a given graph \mathcal{G} . Since in our graph $\mathcal{N}_s \cup \mathcal{N}_d$ is equal to the set of boundary nodes, we have $\mathcal{N}_j \subset \mathcal{N}_0$. From now on we will assume that besides our original graph \mathcal{G} , we also have the graph $\tilde{\mathcal{G}}$, which is the graph created from \mathcal{G} by smoothing out the vertices $\mathcal{N}_0 \setminus \mathcal{N}_j$. Smoothing the vertex w , which is connected to the edges e_1 and e_2 , is the operation which removes w as well as both edges and adds a new edge to the starting and end points of the pair. This edge can be given the direction of any of the two removed edges. Here, it is emphasized that only vertices that connect exactly two edges can be smoothed. This is however the case for the vertices in $\mathcal{N}_0 \setminus \mathcal{N}_j$. This means our new graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{E}}, \tilde{\mathcal{N}})$ is a directed graph, which only has demand nodes, supply nodes and junction nodes in the sense of Definition 3.1.

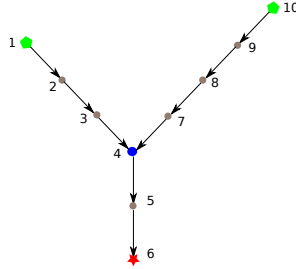


FIGURE 2. A typical gas network

An example network is shown in Figure 2. Here, nodes 1 and 10 denote the supply nodes, node 6 represents the demand node. According to Definition 3.1, only node 4 in Figure 2 is a junction node. This means we can replace this graph by the smoothed graph given in Figure 3

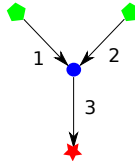


FIGURE 3. Smoothed network of Figure 2 with an ordering of the pipes

By making use of the smoothed graph, the number of algebraic constraints is kept small, while the edges which represent several pipes become long. More details on the number of algebraic constraints will be presented in Proposition 1 at the end of

this section. For each such long pipe $i \in \tilde{\mathcal{E}}$, we have the set of variables $p_1^{(i)}, \dots, p_{n^{(i)}}^{(i)}$ and $q_1^{(i)}, \dots, q_{n^{(i)}}^{(i)}$ that represent the discrete analog of p and q , respectively. Here $n^{(i)}$ is the number of discretization points in a certain edge and includes the number of boundary points. To discretize a certain edge, we ensure that the nodes of the original graph that were smoothed become a subset of the discretization points. We will denote $p_1^{(i)} = p_s^{(i)}$ and $q_{n^{(i)}}^{(i)} = q_d^{(i)}$, for now, as they are boundary conditions in a one pipe system.

Remark 3. Our gas network is modeled by a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{N})$, where the set of boundary nodes is equal to the union of supply nodes and demand nodes. All edges \mathcal{E} are pipes, and an edge attached to a supply node is called a supply pipe, while an edge attached to a demand node is called a demand pipe. A supply pipe is directed away from the supply node and a demand pipe is directed towards the demand node. By smoothing this graph as explained above, we also get a directed graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{E}}, \tilde{\mathcal{N}})$, whose interior nodes are all junction nodes in the sense of Definition 3.1. In $\tilde{\mathcal{G}}$, supply edges and demand edges still exist, and they should be directed as above, but are possibly longer.

Remark 4. An edge in the smoothed graph can represent more than one pipe. This means that the cross-sectional area is not necessarily constant on one edge.

3.3. Nodal conditions. There are two types of constraints concerning the connection of edges, namely the pressure constraint, and the mass flow constraint. These two types of constraints represent the equality of the dynamic pressure and conservation of mass at junction nodes [21].

The so-called pressure nodal condition describes the pressure continuity among pipes connected at a joint junction node, which is given by

$$p_{n^{(i)}}^{(i)} = p_s^{(j)}, \text{ if a node connects the incoming pipe } i \text{ and the outgoing pipe } j. \quad (10)$$

The pressure nodal condition states that the pressure at the end of the outflow pipes should equal the pressure at the beginning of the inflow pipes that connect to the same junction node and ensures that there is only one pressure value at each node.

The second type of nodal condition i.e., the mass flow nodal condition, states the conservation of mass flow at the junction nodes, and it is given by

$$\sum_{i \in \mathcal{I}_k} q_d^{(i)} = \sum_{i \in \mathcal{O}_k} q_1^{(i)} \text{ for every node } k, \quad (11)$$

where \mathcal{I}_k is the set of edges incoming the node k and \mathcal{O}_k the set of edges outgoing of node k . Equation (11) states that the inflow at the junction node k should equal to the outflow at the same junction node k .

3.4. Network assembly. In the discretized model (7) describing the dynamics of gas transported along one single pipe, the variables $p_1 = p_s$ and $q_n = p_d$ are given by the boundary condition, i.e., the prescribed pressure at the input node, and the prescribed mass flow at the demand node. For the network, all variables including $p_s^{(i)}$ and $q_d^{(i)}$ are treated as variables and we add the algebraic constraints (10)–(11) to the system. We take the network in Figure 2, whose smoothed graph, with edge ordering is given by Figure 3 as an example to build the full DAE system. For each pipe we have the pipe dynamics of the discretized system given by (7). The supply pressure for pipe 1 and 2 is given as well as the demand flux for pipe 3. This means we have the extra variables $p_s^{(3)}, q_d^{(1)}, q_d^{(2)}$. We will first use the fact that $p_s^{(3)}$ is equal

to $p_{n(1)}^{(1)}$ and replace it in the equation. We will then still have to make sure that $p_{n(1)}^{(1)} = p_{n(2)}^{(2)}$ and also that the incoming flux at the junction is equal to the outgoing flux. To summarize $q_d^{(1)}, q_d^{(2)}$ are the added variables as $p_s^{(3)}$ was replaced directly and

$$\begin{aligned} q_d^{(1)} + q_d^{(2)} &= q_1^{(3)}, \\ p_{n1}^{(1)} &= p_{n2}^{(2)}, \end{aligned} \quad (12)$$

are the added algebraic constraints.

By using the single pipe model (7), we obtain the mathematical model for the network in Figures 2–3,

$$\begin{aligned} & \underbrace{\begin{bmatrix} \mathcal{M}^{(1)} & & & \\ & \mathcal{M}^{(2)} & & \\ & & \mathcal{M}^{(3)} & \\ & & & 0 \\ & & & & 0 \end{bmatrix}}_{\mathbf{M}} \frac{\partial}{\partial t} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ u^{(3)} \\ q_d^{(1)} \\ q_d^{(2)} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{B}_p^{(1)} & & \\ & \mathcal{B}_p^{(2)} & \\ & & \end{bmatrix}}_{\mathbf{B}_p} \begin{bmatrix} p_s^{(1)} \\ p_s^{(2)} \end{bmatrix} \\ & + \underbrace{\begin{bmatrix} 0 \\ 0 \\ B_q^{(3)} \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}_q} q_d^{(3)} + \underbrace{\begin{bmatrix} \mathcal{K}^{(1)} & & & \mathcal{B}_q^{(1)} \\ & \mathcal{K}^{(2)} & & \mathcal{B}_q^{(2)} \\ \bar{\mathcal{B}}_p^{(3)} & & \mathcal{K}^{(3)} & \\ & e_3 & 1 & 1 \\ e_1 & e_2 & & 0 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} u^{(1)} \\ u^{(2)} \\ u^{(3)} \\ q_d^{(1)} \\ q_d^{(2)} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathcal{G}_1(u^{(1)}, p_s^{(1)}) \\ \mathcal{G}_2(u^{(2)}, p_s^{(2)}) \\ \mathcal{G}_3(u^{(3)}, \bar{e}_3 u^{(1)}) \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{g}^{(*)}}. \end{aligned} \quad (13)$$

Here $u^{(i)} = [p^{(i)}, q^{(i)}]$ and,

$$\begin{aligned} \bar{\mathcal{B}}_p^3 &= [0, \quad 0, \dots, \quad 1, \quad 0, \quad \dots, \quad] \otimes \mathcal{B}_p^3, \\ \mathcal{G}_i(u^{(i)}, p_s^{(i)}) &= \begin{bmatrix} 0 \\ g(p_s^{(i)}, p^{(i)}, q^{(i)}) \end{bmatrix}. \end{aligned} \quad (14)$$

The mass flow nodal condition is represented by the 4th block row in (13), and the pressure nodal condition is given by the 5th block row and also the (3, 1) block of \mathbf{K} in (13). The row vectors e_1 , e_2 , and e_3 are just elementary vectors with 1 or -1 on a certain position and zeros elsewhere, which select the corresponding variables for the nodal conditions (10)–(11).

Note that the matrix \mathbf{K} in (13) is not uniquely defined. This is because we use $p_s^{(3)} = p_{n(1)}^{(1)}$. We can also employ $p_s^{(3)} = p_{n(2)}^{(2)}$, and this in turn moves $\bar{\mathcal{B}}_p^{(3)}$ from the (3, 1) block to the (3, 2) block of \mathbf{K} .

Although we need extra variables for both, the pressure and mass flow, to assemble a global network model, we only introduce extra mass flow variables explicitly while the extra pressure variables can be obtained via applying some pressure nodal conditions directly. This reduces the redundancy in the network modeling.

There is a degenerate case that a network has only one long pipe, i.e., this network has one supply node and one demand node, but no junction node. For such a degenerate network, which is equivalent to one pipe, we do not need to introduce

extra variables since we already have the left and right boundary conditions. For a non-degenerate network, we have the following proposition.

Proposition 1. *Suppose that the network $\tilde{\mathcal{G}} = (\tilde{\mathcal{E}}, \tilde{\mathcal{N}})$ is a connected graph as in Remark 3, and has n_s supply pipes, n_a demand pipes, and n_j junction pipes. Here junction pipes are edges of the smoothed graph that are not supply pipes or demand pipes. Then the following relation between the number of extra variables n_e and the number of extra algebraic constraints n_a of the mathematical model holds:*

$$n_e = n_a = n_s + n_j.$$

Proof. As stated before, we only introduce extra variables for the mass flow of each supply and junction pipe, since the extra pressure variables are directly included at the process of network assembling. Then we have

$$n_e = n_s + n_j,$$

which is due to the fact that the mass flows at the demand pipes are already prescribed.

The algebraic constraints are obtained via applying nodal conditions at the junction nodes. Suppose that the junction node i has $n_{\text{in}}^{(i)}$ injection pipes, and $n_{\text{out}}^{(i)}$ outflow pipes, then we need $(n_{\text{in}}^{(i)} - 1)$ equality constraints to apply the pressure nodal conditions for injection pipes since the pressure nodal conditions for outflow pipes are directly applied at the stage of the network assembly. We have one algebraic constraint to prescribe the mass flow nodal condition for junction node i . Therefore, we need $n_{\text{in}}^{(i)}$ algebraic constraints for junction node i . The sum over all the junction nodes of the network gives the overall number of algebraic constraints:

$$n_a = \sum_i n_{\text{in}}^{(i)}.$$

On the other hand,

$$\sum_i n_{\text{in}}^{(i)} = n_s + n_j,$$

as incoming pipes are never demand pipes and the sum over all incoming pipes is the number of all supply and all junction pipes. \square

Proposition 1 states that the total number of extra variables is equal to the total number of algebraic constraints. This is very important for us to simulate the network model in the form of (13), which will be shown in the next section.

4. Fast numerical methods for simulation. Based on the discretization of the one-dimensional isothermal Euler equation, using the smoothed graph, we are able to reduce the algebraic constraints to a very small number and obtain the nonlinear DAE system (13).

In this section, we introduce the direction following ordering method to order the edges of the smoothed graph to exploit the matrix structure of the DAE for fast computations. We integrate the DAE by a simple implicit Euler method, such that we have to solve a nonlinear system in each time step, which is done by Newton's method (Section 4.1). Looking closely at the matrix structure of the system that has to be solved in each Newton iteration (Section 4.2), we develop a special ordering of the nodes and edges in the graph such that the Jacobian is block lower triangular (Section 4.3), which is then illustrated in an example (Section 4.4). Last but not least, this linear system is solved with a preconditioned iterative method to achieve

maximal speedup of the numerical computation (Section 4.5). The entire workflow is shown in Figure 7.

4.1. Numerical algorithms for differential algebraic equations. Here, we reuse the notation from (13) with simplifications. We obtain the general mathematical model,

$$M\partial_t x = Kx + Bu(t) + f(x, u(t)), \quad (15)$$

where the mass matrix M is singular when there is at least one junction node, and the right hand side function f is nonlinear. In general, the mathematical model (15) is a large system of nonlinear DAEs, where the size of the DAE (15) is proportional to the combined length of the pipes in the network. To solve/simulate such a DAE model is challenging and possibly time-consuming. Related work either focuses on exploiting the DAE structure such that the differential part and the algebraic part are decoupled, and one can solve these two parts separately [1], or reducing the so-called tractability index [16]. Here, we propose a fast numerical method by directly tackling the expensive numerical linear algebra. To simulate the DAE model (15), we discretize in time using the implicit Euler method, and at time step k , we have

$$M \frac{x^k - x^{k-1}}{\tau} = Kx^k + Bu^k + f(x^k, u^k),$$

i.e., we need to solve the following nonlinear system of equations,

$$F(x) = (M - \tau K)x - \tau f(x, u^k) - Mx^{k-1} - \tau Bu^k = 0, \quad (16)$$

at each time step k to compute the solution x^k . Here, we apply Newton's method described by Algorithm 1 to solve the nonlinear system (16) to study the nonlinear dynamics of the network.

Algorithm 1 Newton's method to solve (16)

- 1: **Input:** maximal number of Newton steps n_{\max} , stop tolerance ε_0 , initial guess x_0
 - 2: $m = 0$
 - 3: **while** $m \leq n_{\max}$ & $\|F(x)\| \geq \varepsilon_0$ **do**
 - 4: Compute the Jacobian matrix $D_F(x_m) = \frac{\partial}{\partial x} F|_{x=x_m}$
 - 5: Solve $F(x_m) + D_F(x_m)(x - x_m) = 0$
 - 6: $m \leftarrow m + 1$, $x_m \leftarrow x$
 - 7: **end while**
 - 8: **Output:** solution $x \approx x_m$
-

The biggest challenge for Algorithm 1 is to solve the linear system in line 5 at each Newton iteration, since the Jacobian matrix $D_F(x_m)$ is large. Krylov subspace methods such as the generalized minimal residual (GMRES) method [35] or induced dimension reduction (IDR(s)) method [36] are then appropriate to solve such a system. To accelerate the convergence of such a Krylov subspace method, we need to apply a preconditioning technique by exploiting the structure of the Jacobian matrix $D_F(x)$.

4.2. Matrix structure. The Jacobian matrix is given by

$$D_F(x) = (M - \tau K) + \tau \frac{\partial}{\partial x} f(x, u), \quad (17)$$

where the matrices M and K are two-by-two block matrices, and

$$M = \begin{bmatrix} \bar{M} & \\ & 0 \end{bmatrix}, \quad K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}. \quad (18)$$

Here \bar{M} is block-diagonal, and the second block row of A comes from the algebraic constraints of the network by applying the nodal conditions introduced in Section 3. The size of A_{11} is much bigger than the size of A_{22} since A_{11} comes from the discretization of the isothermal Euler equations over all the edges of the smoothed network, while the size of A_{22} is equal to $n_s + n_j$ according to Proposition 1. Moreover, the partial derivative of the nonlinear term $\frac{\partial}{\partial x} f(x, u)$ has the structure

$$\frac{\partial}{\partial x} f = \begin{bmatrix} D_f & 0 \\ 0 & 0 \end{bmatrix}, \quad (19)$$

since the nonlinear term only acts on the differential part of the DAE (13).

4.3. Edge and node ordering. In the following, we will present a concept that allows us to order the nodes and edges of the graph $\tilde{\mathcal{G}}$ such that the first block of the Jacobian matrix is block lower triangular as presented in Theorem 4.5. In order to get there, we need a few concepts first.

Definition 4.1. A directed acyclic graph (DAG) is a finite, directed graph, which has no directed cycles, meaning there is no vertex from which we can start a path along directed edges that ends up at that vertex again.

Lemma 4.2. *Given a graph with directed and undirected edges, such that the directed edges do not create cycles, we can always direct the undirected edges such that the resulting graph is a directed acyclic graph (DAG).*

Proof. Take a graph satisfying the assumptions and remove all undirected edges. This results in a graph that may not be connected. However, it is a DAG so that we can order the nodes in a topological ordering [2]. Applying such a topological ordering to the original graph, direct the undirected edges according to that topological ordering, which induces another DAG. \square

This means, even by fixing the direction of the supply and demand pipes, we can redirect all the other edges in our graph $\tilde{\mathcal{G}}$ such that the resulting graph is a DAG. To build such a DAG.

Lemma 4.3. *Given a DAG, we can order the edges in such a way that at every node all incoming edges have a lower order than all the outgoing edges, or it does not have an incoming edge. We call this ordering **direction following (DF) ordering**.*

Proof. Since we have a DAG, there exists a topological ordering of the nodes. This means: if an edge goes from node a to node b , then b has a higher order than a . We now order the edges, by their starting node. Hence, an edge has a higher order if its starting node has a higher order. If two edges have the same starting node, then it does not matter in which order we list them, we just pick one. Once we have this ordering of the edges based on the order of the nodes we ensure that all outgoing edges at a node have a higher order than all the incoming edges of a node. \square

The network in Figure 4(a) is a smoothed graph that represents a gas network. Only the directions of the supply and demand pipes are fixed. We start ordering the nodes of the graph from the supply nodes, and end up with the demand nodes, which gives a topological ordering of the nodes. Now, we can plot the graph along a line as in Figure 4(b). The directions of the undirected edges of the graph are picked up by pointing away from lower order nodes to higher order nodes, which induces a DAG. This in turn leads to an ordering of the edges, which is the index above each edge in Figure 4(b). Note that the DF ordering is not unique.

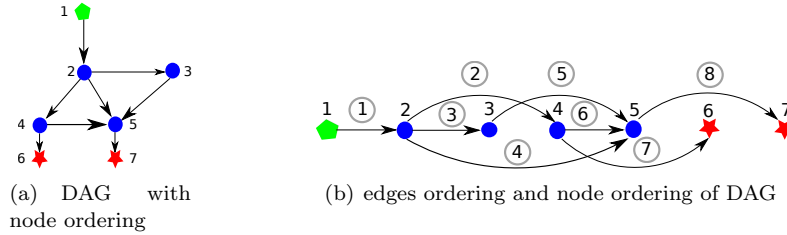


FIGURE 4. An illustrative network example of a DAG

Definition 4.4. A **smoothed direction following (SDF)** gas network graph is a directed acyclic graph, whose boundary nodes are supply or demand nodes, directions are away from supply nodes and towards demand nodes. All edges are sorted with the DF ordering and there are no nodes in the graph that connect exactly two edges.

From now on, we assume that our modeling is such that $\tilde{\mathcal{G}}$ is a SDF gas network graph. Then we have the following proposition to illustrate the structure of the partial derivative of the nonlinear term (19).

Proposition 2. *Given a SDF gas network graph, we can construct the DAE system in such a way that D_f in (19) has a block lower-triangular structure.*

Proof. The block of f corresponding to the i -th pipe has the structure,

$$f_i = \mathcal{G}_i = \begin{bmatrix} 0 \\ g(p_{\text{in}}, p^{(i)}, q^{(i)}) \end{bmatrix},$$

where the structure of $g(p_{\text{in}}, p^{(i)}, q^{(i)})$ is given by (8). Here

$$p_{\text{in}} = \begin{cases} p_s, & \text{when the } i\text{-th pipe is a supply pipe,} \\ p_{\text{out}}^{(j)}, & \text{when pipe } j \text{ is an incoming pipe of the node where pipe } i \text{ is outgoing.} \end{cases}$$

By the selected ordering, we always have $j < i$. Then the upper triangular blocks of D_f are 0. The diagonal blocks are given by

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{g(p_{\text{in}}, p^{(i)}, q^{(i)})}{\partial p^{(i)}} & \frac{g(p_{\text{in}}, p^{(i)}, q^{(i)})}{\partial q^{(i)}} \end{bmatrix},$$

which is again block lower triangular and in particular, with an easy structure of the diagonal blocks, since $\frac{g(p_{\text{in}}, p^{(i)}, q^{(i)})}{\partial q^{(i)}}$ in our discretization is tridiagonal. \square

Similar to Proposition 2, we can also show that the $(1, 1)$ block of K in (18) has a lower-triangular block structure.

Proposition 3. *Given an SDF gas network graph, K_{11} in (18) is also block lower-triangular.*

Proof. For the i -th block row of K_{11} , the off-diagonal blocks are zero if the i -th pipe is a supply pipe. If the i -th pipe is not a supply pipe and connected with other pipes, then the off-diagonal block (i, j) is nonzero if the j -th pipe corresponds to one of the flow injection pipes of the i -th pipe. This is because the pressure nodal condition (10) is applied to the i -th pipe. According to Definition 4.4, we can pick the pressure condition for the i -th pipe by any one of the injecting pipes j , which are all of lower order and therefore ensure $i > j$, and this completes the proof. \square

If we partition the Jacobian matrix (17) by a 2-by-2 block structure as in (18), then we have the following theorem to illustrate the structure of the $(1, 1)$ block of the Jacobian matrix.

Theorem 4.5. *Given an SDF gas network graph, we are able to model the system in such a way, that the $(1, 1)$ block of the Jacobian matrix (17) has a block lower-triangular structure.*

Proof. According to (17)–(19), the $(1, 1)$ block of the Jacobian matrix $D_F(x)$ is,

$$\bar{M} - \tau K_{11} + \tau D_f.$$

According to Proposition 2 and Proposition 3, K_{11} and D_f are block lower-triangular matrices. Since \bar{M} is a block diagonal matrix, this completes the proof. \square

4.4. Example for a large network. Next, we use a benchmark network from [34], given by Figure 5, to show the structure of the Jacobian matrix $D_F(x)$ of the first Newton iteration for the first time step, i.e., $D_F(x_1^1)$ before and after applying the DF ordering. The network parameters are also given in [34]. We set the mesh size for the FVM discretization to be 20 meters, i.e., $h = 20$. The sparsity pattern of $D_F(x_1^1)$ before and after the DF ordering are given by Figure 6.

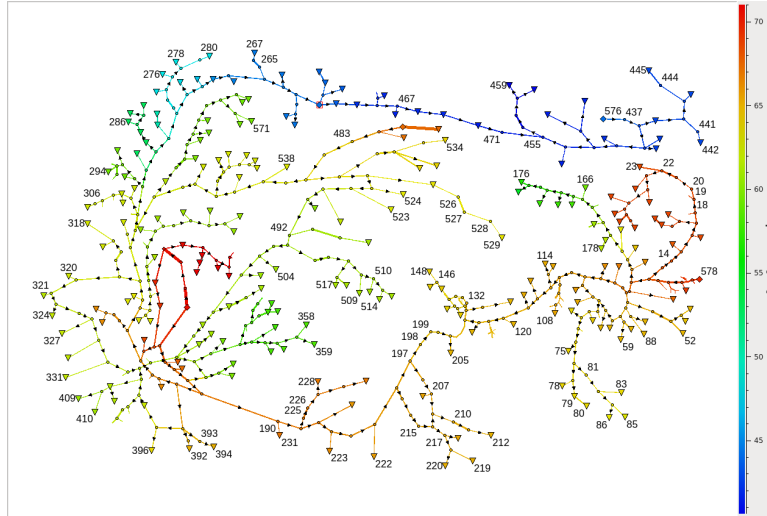


FIGURE 5. Big benchmark network in [34]

Figure 6 shows that the Jacobian matrix $D_F(x)$ is a sparse matrix. After applying the DF ordering, the $(1, 1)$ block has a block lower-triangular structure, and the size

of the $(1, 1)$ block is much bigger than the $(2, 2)$ block. For the case when the mesh size is 20 meters, the $(1, 1)$ block is a $200,348 \times 200,348$ block lower-triangular matrix while the size of the $(2, 2)$ block is 417×417 . In general, the size of the $(2, 2)$ block of the Jacobian matrix is fixed since it equals the number of algebraic constraints. According to Proposition 1, it equals $n_s + n_j$, where n_s is the number of supply pipes and n_j is the number of junction pipes. The size of the $(1, 1)$ block depends on the mesh size and equals twice the total pipeline length divided by the mesh size. Therefore, it is much bigger than $n_s + n_j$, if the total length of the pipelines is much larger compared to the mesh size.

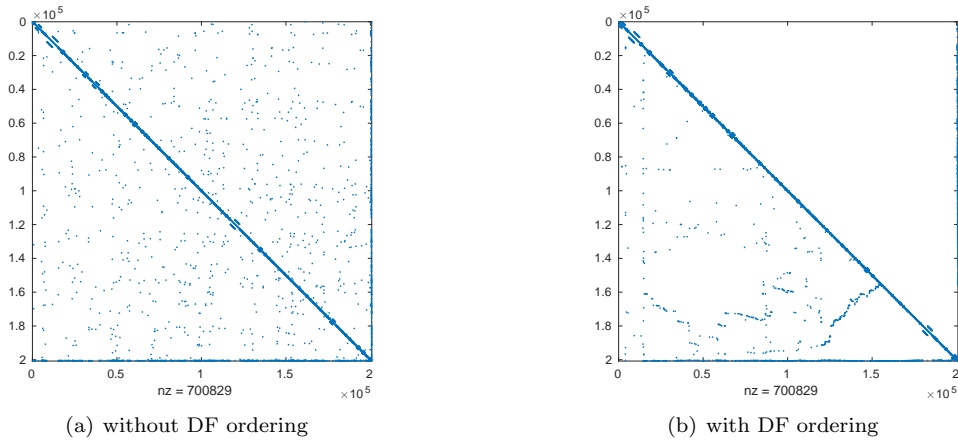


FIGURE 6. Sparsity pattern of J without and with DF ordering

4.5. Preconditioning technique. The specific structure of the Jacobian matrix can be exploited to solve the Jacobian system fast for the simulation of the gas network. Recall that in order to simulate the discretized gas network model, we need to apply Algorithm 1, where we need to solve a Jacobian system at each Newton iteration for each time step k ($k = 1, 2, \dots, n_t$). To solve the Jacobian system, we exploit the 2-by-2 structure of the Jacobian matrix. Here we write the Jacobian matrix $D_F(x)$ as

$$D_F(x) = \begin{bmatrix} D_{F_{11}} & D_{F_{12}} \\ D_{F_{21}} & D_{F_{22}} \end{bmatrix}.$$

Note that the Jacobian matrix $D_F(x)$ has a special structure, which is called *generalized saddle-point structure* [4]. This enables us to make use of the preconditioning techniques designed for generalized saddle-point systems to solve the Jacobian system. Generalized saddle-point systems come from many applications, such as computational fluid dynamics [10], PDE-constrained optimization [33], or optimal flow control [32]. Many efforts have been dedicated to the efficient numerical solution of such systems using preconditioning techniques [29, 31, 43, 30, 38], we recommend [4, 42] for a general survey of preconditioning generalized saddle-point systems.

We can compute a block LU factorization by

$$D_F(x) = \begin{bmatrix} D_{F_{11}} & \\ & S \end{bmatrix} \begin{bmatrix} I & D_{F_{11}}^{-1} D_{F_{12}} \\ & I \end{bmatrix}. \quad (20)$$

Here $S = D_{F_{22}} - D_{F_{21}} D_{F_{11}}^{-1} D_{F_{12}}$ is the Schur complement of $D_F(x)$. According to Theorem 4.5, $D_{F_{11}}$ has a block lower-triangular structure, and the size of $D_{F_{22}}$ is much smaller than that of $D_{F_{11}}$. Therefore, we can compute the Schur complement S by block forward substitution, and apply the preconditioner

$$P = \begin{bmatrix} D_{F_{11}} & \\ D_{F_{21}} & S \end{bmatrix}, \quad (21)$$

to solve the Jacobian system using a preconditioned Krylov solver. Associated with the block LU factorization (20), we can immediately see that the preconditioned spectrum satisfies $\lambda(P^{-1}D_F(x)) = \{1\}$. Moreover, the minimal polynomial of the preconditioned matrix $P^{-1}D_F(x)$ has degree 2, so that a method like generalized minimum residual (GMRES) [35] would converge in at most two steps [4].

At each iteration of the Krylov solver, we need to solve the system

$$\begin{bmatrix} D_{F_{11}} & \\ D_{F_{21}} & S \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix},$$

which can be solved easily since $D_{F_{11}}$ is a block lower-triangular system, and $S = D_{F_{22}} - D_{F_{21}} D_{F_{11}}^{-1} D_{F_{12}}$ can be computed directly since the size of S is much smaller than $D_{F_{11}}$. Note that at each time step k , we need to solve a nonlinear system using Newton's method, and we need to apply a preconditioned Krylov subspace method to solve a Jacobian system at each Newton iteration. For such a preconditioned Krylov solver, we need to compute the Schur complement S at each Newton iteration. This can still be computationally expensive for gas network simulation within a certain time horizon. We can further simplify the preconditioner by applying a fixed preconditioner P_1 for all Newton iterations and all time steps, i.e., we choose

$$P_1 = \begin{bmatrix} D_{F_{11}}^1 & \\ D_{F_{21}}^1 & S^1 \end{bmatrix}, \quad (22)$$

where P_1 comes from the block LU factorization of the Jacobian matrix $D_F^1(x_1)$ of the first Newton iteration for the first time step, and $S^1 = D_{F_{22}}^1 - D_{F_{21}}^1 (D_{F_{11}}^1)^{-1} D_{F_{12}}^1$. Note that for the preconditioner P_1 , we just need to compute the Schur complement once and use it for all the Newton iterations of all time steps.

Next, we show the performance of the DF ordering for the Schur complement S^1 computation. Again, we use the network given in Figure 5 as an example and perform a FVM discretization of the network using different mesh sizes. We report the computational results in Table 1, where all timings are given in seconds. Here, h is the mesh size, and $\#D_F$ represents the size of the Jacobian matrix D_F given by (20). The applications of the operator $(D_{F_{11}}^1)^{-1}$ are performed using the MATLAB `backslash` operator for both cases.

TABLE 1. Computational time (seconds) for Schur complement S^1

h	$\#D_F$	with DF	without DF
20	2.01e+05	8.12	8.75
10	3.97e+05	17.84	19.14
5	7.91e+05	38.44	41.75
2.5	1.58e+06	81.42	87.77

We have noticed that for medium problem sizes, the advantage of computing S^1 using the block lower-triangular structure obtained from the DF ordering over

that without using the DF ordering is not very prominent. This is due to the fact that while computing S^1 using the block lower-triangular structure of D_{F11}^1 obtained from the DF ordering, MATLAB has an overhead calling the block forward substitution. This overhead is comparable with the hardcore computation time for medium problem sizes. When the problem size gets bigger, this overhead is less comparable with the hardcore computation time, which is demonstrated by the results in Table 1. We believe that the advantage of computations with the DF ordering over computations without the DF ordering will become more apparent once we use a tailored high performance computation implementation.

Since P_1 is a good preconditioner for $D_F^1(x_1)$, it is only a good preconditioner for the Jacobian matrix $D_F(x)$ at the other Newton iterations and other time steps, if it remains close to $D_F^1(x_1)$. This is often true for gas networks since the Jacobian matrix (17) has two parts, i.e., the linear part and the linearized part. The linear part is dominant since it models the transportation phenomenon of the gas while the nonlinear term acts as the friction term for such a transportation. This makes P_1 a good preconditioner for solving the Jacobian systems for all Newton steps of all time steps, as it will be demonstrated by numerical experiments in the next section. Note that if we keep updating the preconditioner (21) more often than simply using a single preconditioner P_1 in (22), we will obtain better performance for the preconditioned Krylov solver, which in turn needs more time for preconditioner computation. A compromise has to be made to achieve the optimal performance for the gas network simulation with respect to total computational time.

By applying the preconditioner P_1 in (22), we show the computational diagram to illustrate the process of gas network simulation in Figure 7.

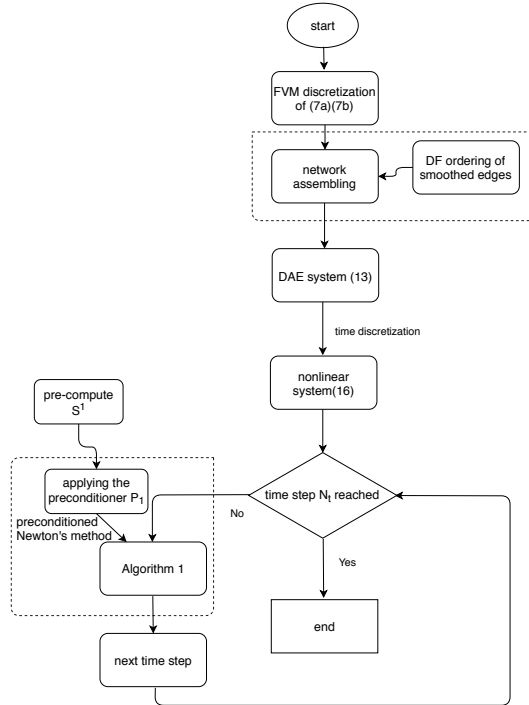


FIGURE 7. Computational diagram for gas network simulation

5. Numerical results. In this section, we report the performance of our numerical algorithms for the simulation of gas networks. We apply our numerical algorithms to the benchmark problems of several gas networks given in [13, 16, 15, 14] to show the performance of our methods. All numerical experiments are performed in MATLAB 2017a on a desktop with Intel(R) Core(TM)2 Quad CPU Q8400 of 2.66GHz, 8 GB memory and the Linux 4.9.0-6-amd64 kernel.

5.1. Comparison of discretization methods. In this section, we compare the performance of the finite volume method (FVM) with that of the finite difference method (FDM) for the discretization of the gas networks. We apply both the FVM and FDM to a pipeline network illustrated in Figure 8. Parameter settings for this pipeline network are given in [14].



FIGURE 8. Pipeline network in [16]

We discretize the pipeline network using FVM and FDM with different mesh sizes, and the discretized pipeline networks result in ordinary differential equations (ODEs) since there is no algebraic constraint. We simulate the ODE systems using the routine `ode15s` in MATLAB over the time horizon $[0, 10^5]$ with the same setting of the initial condition for the ODEs. The computational results are given in Figure 9, where the x -axis represents the mesh sizes in meters.

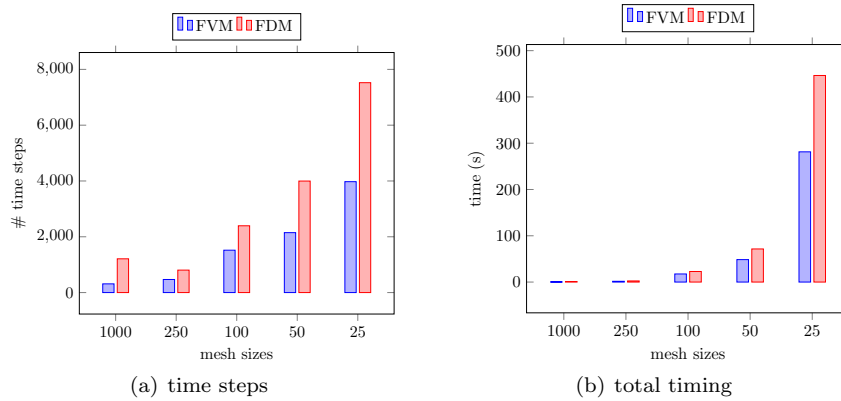


FIGURE 9. Comparison of FVM and FDM for a single pipe network

Figure 9(a) shows the number of time steps that `ode15s` uses to simulate the ODEs given by FVM and FDM discretizations over the time horizon $[0, 10^5]$. One can see that for a given mesh size, we need less time steps for the ODE given by the FVM discretization than for the ODE given by the FDM discretization which results in less total computation time for the simulation of the ODE given by the FVM discretization, which is also shown in Figure 9(b). Obviously, the error control in `ode15s` leads to more time steps for the FDM discretization than for the FVM discretization to reach the same accuracy, which is confirmed by the following experiment.

Next, we use another network to show that with the same mesh, the model given by the FVM discretization gives more accurate results than the FDM discretization. The test network is given in Figure 10, where the network parameters are given

in [16]. We use the FVM and FDM methods to discretize the network in Figure 10 and apply the computational method depicted in Figure 7. We choose different mesh sizes for the discretization, and fix the step size for the time discretization to be one second, i.e., $\tau = 1$. We plot the mass flow at the supply node 57 in Figure 11.

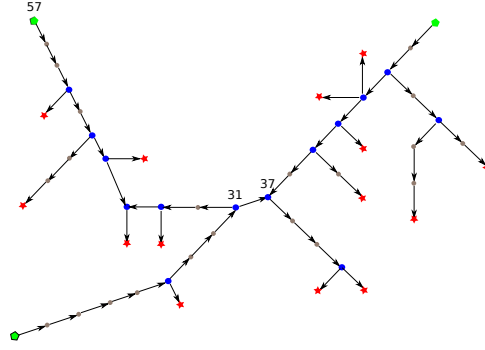


FIGURE 10. Medium size network

The mass flow at node 57 computed by using different discretized DAE models (16) is plotted in Figure 11(a), showing similar dynamical behavior of the different models. However, when we look at the dynamics of the mass flow during the first 5 seconds, we can see quite a big difference in Figure 11(b). With the mesh refinement, the solutions of the model given by both the FVM and the FDM discretizations converge. Moreover, we can infer that we can use a bigger mesh size for the FVM discretization than for the FDM discretization to get the same accuracy.

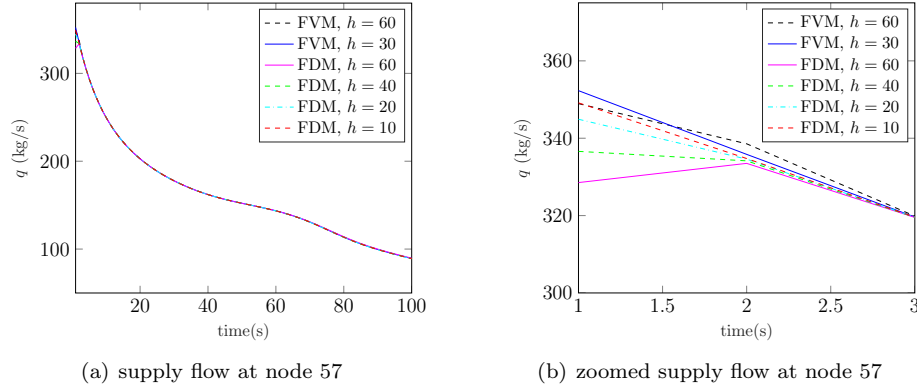


FIGURE 11. Comparison of FVM and FDM for a medium network

The computational results given by Figure 9 and Figure 11 show that when we use the same mesh size to discretize the network, the model given by the FVM discretization is more accurate than the model by the FDM discretization. To get the same model accuracy, we can use a bigger mesh size to discretize the network by FVM than that by FDM. This in turn yields a smaller model given by the FVM discretization than the model given by the FDM discretization. This in turn means that the FVM discretized model is easier to solve than the FDM discretized model.

We also plot the condition number of the Jacobian matrix ($\kappa(D_F)$) of all the Newton iterations for the first time step with a mesh size $h = 60$ to discretize

the DAE, which are given in Table 2. It illustrates that the condition number of the Jacobian matrix of the FVM discretized model is about 10 times smaller than the condition number of the Jacobian matrix of the FDM discretized model, which makes solving such a FVM discretized model easier than solving a FDM discretized model.

TABLE 2. Condition number of the Jacobian matrix D_F from FVM and FDM, 1st time step, $h = 60$

Newton iter.	1	2	3	4
FVM	1.56e+07	1.57e+07	1.57e+07	1.57e+07
FDM	1.24e+08	1.25e+08	1.25e+08	1.25e+08

Computational results in Figure 9 and Table 2 show that the FVM offers a big advantage over the FDM. When using the same mesh size for discretization, FVM gives a more accurate model than the FDM discretization. Moreover, the Jacobian matrix from the FVM discretized model has a better condition number than the Jacobian matrix from the FDM discretized model, which makes it easier to simulate the FVM discretized model. To get the same model accuracy, the size of the FVM discretized model is smaller than the size of the FDM discretized model, and it is therefore computationally cheap. For the comparison of the finite element method (FEM) with FDM for the gas network simulation, we refer to an early study in [28], where the authors preferred FDM due to the comparable accuracy with FEM and less computational time.

5.2. Change of flow direction. The basis of our modeling is a directed graph, with the implicit assumption that the gas flow follows that direction. However, the flow direction may change due to the change of operation conditions of the gas network. In this part, we show that the direction is just a theoretical construction but that the gas is allowed to flow in either direction meaning that the mass flow can be negative and does not influence the performance of our methods. The change of the flow direction does not change the mathematical formulation of algebraic constraints. Therefore, the structure of the system stays unchanged with respect to the change of the flow direction. This means that we do not require the foreknowledge of the flow direction.

Note that the nonlinear term $\frac{q|q|}{p}$ which describes the damping law of gas networks is still differentiable with respect to the change of the flow direction. For details of the regularity of the solution (p, q) of the PDE (5), we refer to Theorem 3.2 given in [9]. It is also pointed out by Theorem 4.3 in [9] that the forward PDE operator of (5) is Fréchet differentiable with Lipschitz continuous derivative. Therefore, we can still apply Newton's method to solve such a nonlinear system of equations.

First, we test two different cases, which corresponds to two different flow direction profiles of the network, cf. Figure 2. Case 1 corresponds to $p_s^{(1)} = p_s^{(2)} = 30$ bar, and $q_d = 30$ kg/s, while case 2 corresponds to $p_s^1 = 30$ bar, $p_s^2 = 20$ bar, and $q_d = 30$ kg/s. We plot the mass flow at the supply pipe 1 and 2 in Figures 12–13.

Figure 12 shows that the mass flow at both supply nodes approaches the steady state after oscillation for a short while, and both input mass flows have a positive sign. This represents that both supply nodes inject gas flow into the network to supply gas to the demand node 6. After changing the operation condition of the

network, e.g., changing the pressure at supply nodes, the mass flow is redistributed as shown in Figure 13. The mass flow at supply node 10 becomes negative after a few seconds and remains negative after the network reaches steady state. For this case, the supply pipe at node 10 acts as a demand pipe since gas flows out of the network there. For both cases, the equality $q_s^1 + q_s^2 = \sum q_d$ holds, which can be easily seen by looking at the steady state solution.

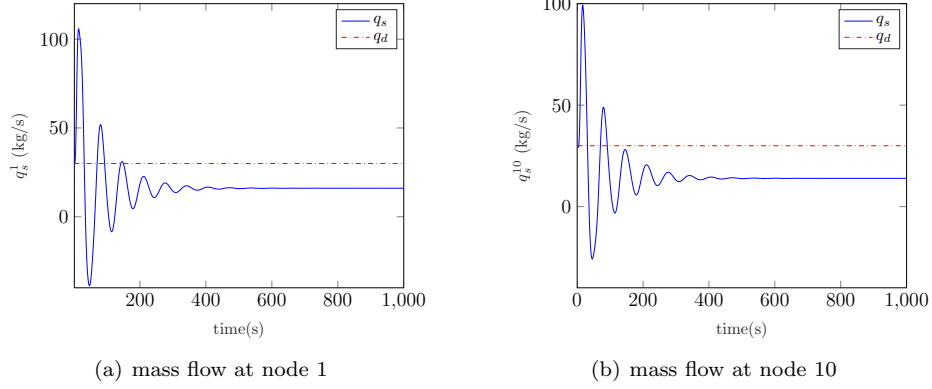


FIGURE 12. Mass flow at supply nodes for case 1

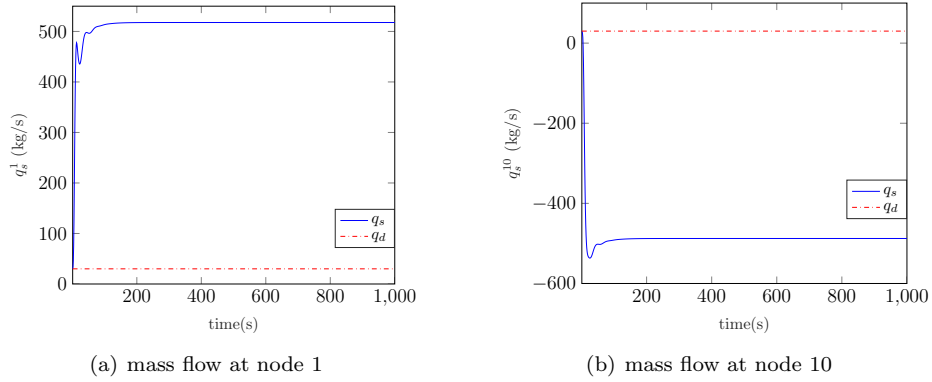


FIGURE 13. Mass flow at supply nodes for case 2

We also apply two different cases to a more complicated network given in Figure 10 to test the robustness of our methods. Case 1 corresponds to $p_s^{55} = p_s^{56} = 50.5$ bar, $p_s^{57} = 50.8$ bar while case 2 has $p_s^{55} = p_s^{56} = 50.5$ bar, and $p_s^{57} = 50.0$ bar. The demand of gas at the demand nodes is the same for both cases. We show the mass flow at the pipe that connects node 31 and 37, which also connects two sub-networks. The mass flow for the pipe $31 \rightarrow 37$ for different cases is shown in Figure 14. The initial conditions of the gas network for the simulation of the two different cases are the same.

The simulation results in Figure 14 show that the flow direction at pipe $31 \rightarrow 37$ changes for the above two cases. The steady state of the mass flow for the two cases shows that the flow can travel in a direction opposite to the prescribed flow direction, and the inflow at node 31 is equal to the outflow at node 37 for the steady state. The imbalance between the inflow and outflow in the transient process is necessary

to build the pressure profile of the network. Computational results in Figures 12–14 show that the gas can flow in the opposite direction as the directed graph suggests. We do not need to introduce another set of variables and switch to another model when the flow changes direction.

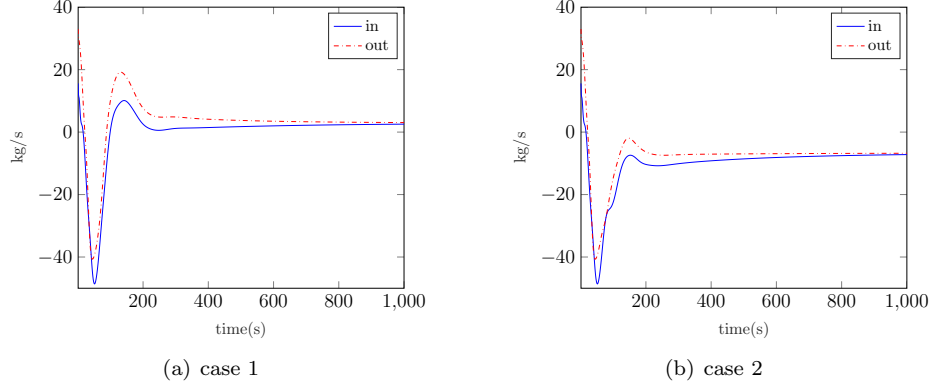


FIGURE 14. Mass flow for the pipe 31 → 37

5.3. Preconditioning performance. As introduced in the previous section, the biggest challenge for applying Algorithm 1 to simulate a gas network lies in the effort spent to solve the linear system at each Newton iteration. For large-scale networks, we need smaller mesh sizes to discretize such networks and this results in larger sizes of the DAEs. Therefore, we need to employ iterative solvers to compute the solution of such a large-scale linear system at each Newton iteration, while preconditioning is essential to accelerate the convergence of such iterative solvers. In this part, we study the performance of the preconditioner (22).

We test the performance of the preconditioner for the network in Figure 5 using different mesh sizes for the FVM discretization. At each Newton (outer) iteration, we solve a linear system by applying an (inner) Krylov solver, e.g., the IDR(s) solver [41, 36], and this is called Newton-Krylov method. Note that the Newton-Krylov method is an inexact Newton method, and at each Newton iteration, we apply the IDR(s) method to solve the linear system up to an accuracy ε_{tol} , i.e.,

$$\|F(x_m) + DF(x_m)(x - x_m)\| \leq \varepsilon_{tol} \|F(x_m)\|,$$

where ε_{tol} is related to the forcing term for an inexact Newton’s method [25]. Since the Newton-Krylov method is inexact, we show its convergence with respect to different tolerances of the Krylov solver, i.e., $\|F\|_2$ with respect to different settings of ε_{tol} . We use the “true” residual computed by using a direct method, i.e., the `backslash` operator implemented in MATLAB for comparison. We report the computational results for the FVM discretization with mesh sizes of 50 and 40, and the time step size τ is set to be 1. For the convergence rate of the inexact Newton method with respect to ε_{tol} , we refer to [7].

The computational results of the nonlinear residual $\|F\|_2$ in Figure 15 for two different mesh sizes show that the accuracy of the inner iteration loop can be set relatively low while the convergence of the outer iteration can still be comparable with more accurate inner loop iterations. The convergence properties of the Newton iteration for the first time step are the same when the inner loop is solved accurately, or when the inner loop is solved up to an accuracy of 10^{-6} or 10^{-4} . If the inner

loop is solved up to an accuracy of 10^{-3} , only one more Newton iteration is needed. Moreover, the convergence behavior of the Newton iteration for different inner loop solution tolerances are the same for the 10-th time step. If lower inner loop accuracy is used, less computational effort is needed, which reduces the computational complexity. The number of IDR(4) iterations for different inner loop tolerances are reported in Figure 16–17.

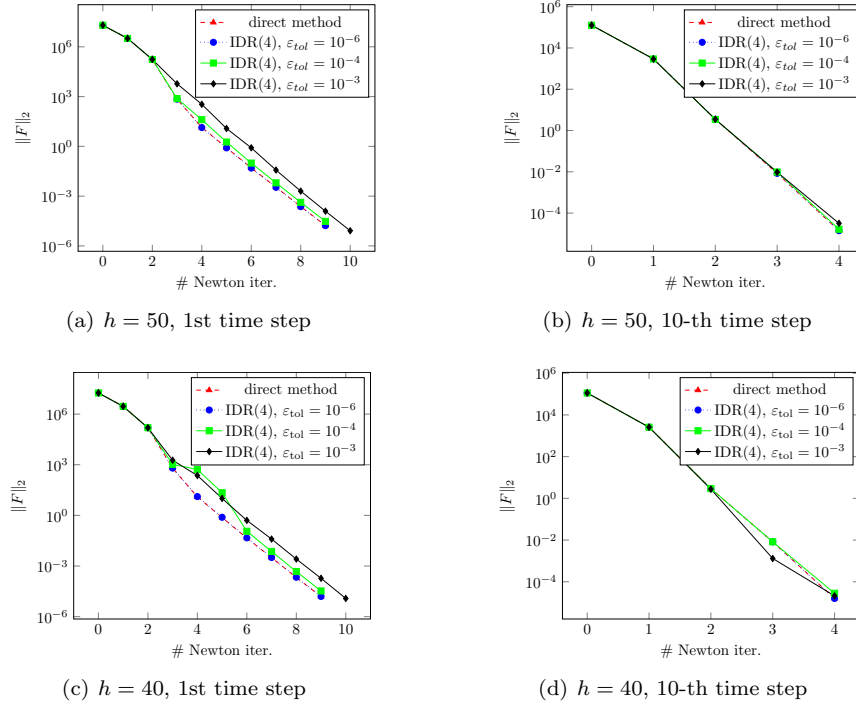


FIGURE 15. Nonlinear residual at the first and tenth time step

The computational results in Figure 16(a) show that the total number of IDR(4) iterations (47) for $\varepsilon_{tol} = 10^{-6}$ is almost twice the total number of IDR(4) iterations (24) for $\varepsilon_{tol} = 10^{-3}$. This demonstrates that the computational work for the first time step can be reduced to almost 50% since the most time consuming part inside each Newton iteration is the IDR(4) solver. Similar results are shown in Figure 16(b). As the system gets closer to steady state, less Newton iterations are needed, and the IDR(4) solver also needs less iterations, as shown in Figure 17. At this stage, IDR(4) with $\varepsilon_{tol} = 10^{-3}$ still needs less work than IDR(4) with $\varepsilon_{tol} = 10^{-6}$, but this is no longer as significant.

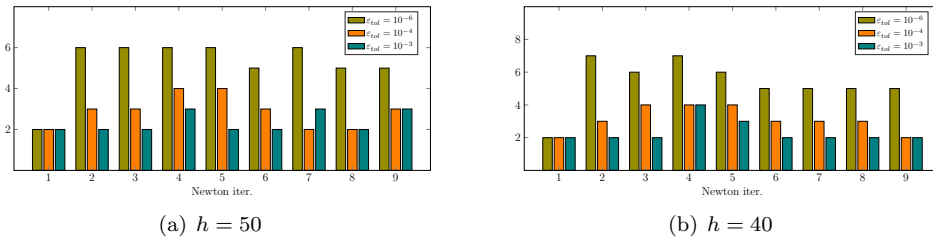


FIGURE 16. Number of IDR(4) iterations at the first time step

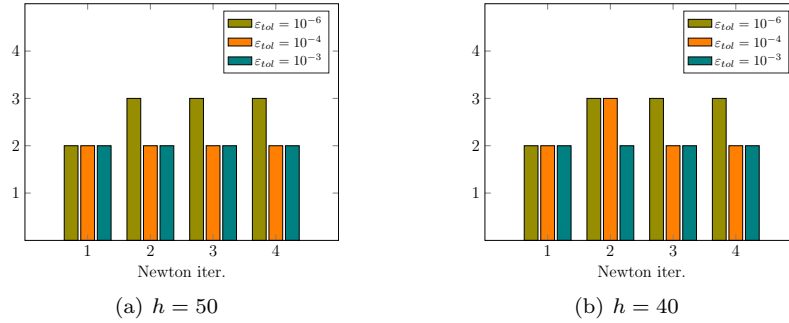


FIGURE 17. Number of IDR(4) iterations at the 10-th time step

We have already observed that the performance of our modeling is robust and convergence of the preconditioned Krylov solver happens quickly. Next, we report the computational time for computing the preconditioner P_1 and applying the preconditioned IDR(4) solver of the first Newton step for the first time step for different FVM discretization mesh sizes. We solve the Jacobian system up to an accuracy of 10^{-4} , and the computational results are given by Table 3. Here “-” represents running out of memory, $\#D_F$ represents the size of the Jacobian system, t_{S^1} denotes the time to compute the Schur complement in P_1 , and all the timings are measured in seconds.

TABLE 3. Computational time for the 1st Newton iteration

h	$\#D_F$	t_{S^1}	IDR(4)	backslash
40	1.03e+05	3.85	0.25	0.13
20	2.01e+05	8.12	0.52	0.36
10	3.97e+05	17.84	1.06	1.18
5	7.91e+05	38.44	2.13	1054.62
2.5	1.58e+06	81.42	4.34	-

The computational results in Table 3 show the advantage of our preconditioner in solving the large-scale Jacobian system over the direct solver. The time to solve the preconditioned system using the IDR(4) solver scales linearly with the system size, and is much smaller than the time to apply the direct solver when the mesh sizes are smaller than 20. For large-scale Jacobian systems, the direct solver either takes up too much CPU time or fails to solve the Jacobian system due to running out of memory. For smaller Jacobian systems, the direct solver shows the advantage over preconditioned Krylov solvers. This is primarily because there is a big overhead when applying the preconditioned IDR(4) solver while the `backslash` operator is highly optimized for smaller systems. The time to compute the Schur complement in preconditioner P_1 scales almost linearly with the system sizes.

6. Conclusions. In this paper, we studied the modeling and simulation of gas networks. We applied the finite volume method (FVM) to discretize the incompressible isothermal Euler equation, and compared it with the finite difference method (FDM). Numerical results show the advantage of the FVM over the FDM. To model gas networks, we introduced the smoothed gas network concept, which represents the topology of the network interconnection and reduces the size of the algebraic

constraints of the resulting system of differential algebraic equations (DAEs) compared with current research. To simulate such a DAE system, we proposed the *direction following* (DF) ordering of the edges of the smoothed network. Through such a DF ordering, we exploited the structure of the system matrix and proposed an efficient preconditioner to solve the DAE. Numerical results show the advantage of our algorithms.

REFERENCES

- [1] N. Banagaaya, S. Grundel and P. Benner, Index-aware MOR for gas transport networks with many supply inputs, in *IUTAM Symposium on Model Order Reduction of Coupled Systems* (eds. J. Fehr and B. Haasdonk), Springer International Publishing, Cham, 2020, 191–207.
- [2] J. Bang-Jensen and G. Z. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer-Verlag, London, 2008.
- [3] P. Benner, S. Grundel, C. Himpe, C. Huck, T. Streubel and C. Tischendorf, Gas network benchmark models, in *Differential-Algebraic Equations Forum*, Springer, Berlin, Heidelberg, 2018.
- [4] M. Benzi, G. H. Golub and J. Liesen, Numerical solution of saddle point problems, *Acta Numer.*, **14** (2005), 1–137.
- [5] A. Bermúdez, X. López and M. E. Vázquez-Cendón, Finite volume methods for multi-component Euler equations with source terms, *Comput. Fluids*, **156** (2017), 113–134.
- [6] M. Chaczykowski, Sensitivity of pipeline gas flow model to the selection of the equation of state, *Chem. Eng. Res. Des.*, **87** (2009), 1596–1603.
- [7] R. Dembo, S. Eisenstat and T. Steihaug, Inexact Newton methods, *SIAM J. Numer. Anal.*, **19** (1982), 400–408.
- [8] H. Egger, A robust conservative mixed finite element method for isentropic compressible flow on pipe networks, *SIAM J. Sci. Comput.*, **40** (2018), A108–A129.
- [9] H. Egger, T. Kugler and N. Strogies, Parameter identification in a semilinear hyperbolic system, *Inverse Probl.*, **33** (2017), 055022.
- [10] H. Elman, D. Silvester and A. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, New York, 2014.
- [11] A. Fügenschuh, et al., Physical and technical fundamentals of gas networks, in *Evaluating Gas Network Capacities*, Society for Industrial and Applied Mathematics, Philadelphia, 2015, 17–43.
- [12] T. G. Grandón, H. Heitsch and R. Henrion, A joint model of probabilistic/robust constraints for gas transport management in stationary networks, *Comput. Manag. Sci.*, **14** (2017), 443–460.
- [13] S. Grundel, N. Hornung, B. Klaassen, P. Benner and T. Clees, Computing surrogates for gas network simulation using model order reduction, in *Surrogate-Based Modeling and Optimization*, Applications in Engineering, Springer, New York, 2013, 189–212.
- [14] S. Grundel, N. Hornung and S. Roggendorf, Numerical aspects of model order reduction for gas transportation networks, in *Simulation-Driven Modeling and Optimization*, Springer Proceedings in Mathematics & Statistics, 153, 2016, 1–28.
- [15] S. Grundel and L. Jansen, Efficient simulation of transient gas networks using IMEX integration schemes and MOR methods, in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, 4579–4584.
- [16] S. Grundel, L. Jansen, N. Hornung, T. Clees, C. Tischendorf and P. Benner, Model order reduction of differential algebraic equations arising from the simulation of gas transport networks, in *Progress in Differential-Algebraic Equations*, Differential-Algebraic Equations Forum, Springer Berlin Heidelberg, 2014, 183–205.
- [17] M. Gugat, F. M. Hante, M. Hirsch-Dick and G. Leugering, Stationary states in gas networks, *Netw. Heterog. Media*, **10** (2015), 295–320.
- [18] F. M. Hante, G. Leugering, A. Martin, L. Schewe and M. Schmidt, *Challenges in Optimal Control Problems for Gas and Fluid Flow in Networks of Pipes and Canals: From Modeling to Industrial Applications*, Springer Verlag, Singapore, 2017, 77–122.
- [19] A. Herrán-González, J. M. D. L. Cruz, B. D. Andrés-Toro and J. L. Risco-Martín, Modeling and simulation of a gas distribution pipeline network, *Appl. Math. Model.*, **33** (2009), 1584–1600.

- [20] M. Herty, [Modeling, simulation and optimization of gas networks with compressors](#), *Netw. Heterog. Media*, **2** (2007), 81–97.
- [21] M. Herty, [Coupling conditions for networked systems of Euler equations](#), *SIAM J. Sci. Comput.*, **30** (2008), 1596–1612.
- [22] M. Herty, J. Mohring and V. Sachers, [A new model for gas flow in pipe networks](#), *Math. Methods Appl. Sci.*, **33** (2010), 845–855.
- [23] C. Huck, L. Jansen and C. Tischendorf, [A topology based discretization of PDAEs describing water transportation networks](#), *Proc. Appl. Math. Mech.*, **14** (2014), 923–924.
- [24] C. Johnson and J. Pitkäranta, [An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation](#), *Math. Comp.*, **46** (1986), 1–26.
- [25] C. Kelley, [Solving Nonlinear Equations with Newton's Method](#), Society for Industrial and Applied Mathematics, Philadelphia, 2003.
- [26] A. Osadacz, [Simulation of transient gas flows in networks](#), *Internat. J. Numer. Methods Fluids*, **4** (1984), 13–24.
- [27] A. Osadacz, [Simulation and Analysis of Gas Networks](#), Gulf Publishing, Houston, TX, 1987.
- [28] A. J. Osadacz and M. Yedroudj, [A comparison of a finite element method and a finite difference method for transient simulation of a gas pipeline](#), *Appl. Math. Model.*, **13** (1989), 79–85.
- [29] J. W. Pearson, [On the development of parameter-robust preconditioners and commutator arguments for solving Stokes control problems](#), *Electron. Trans. Numer. Anal.*, **44** (2015), 53–72.
- [30] J. Pestana and A. J. Wathen, [Natural preconditioning and iterative methods for saddle point systems](#), *SIAM Rev.*, **57** (2015), 71–91.
- [31] M. Porcelli, V. Simoncini and M. Tani, [Preconditioning of active-set Newton methods for PDE-constrained optimal control problems](#), *SIAM J. Sci. Comput.*, **37** (2015), S472–S502.
- [32] Y. Qiu, [Preconditioning Optimal Flow Control Problems Using Multilevel Sequentially Semiseparable Matrix Computations](#), Ph.D thesis, Delft Institute of Applied Mathematics, Delft University of Technology, 2015.
- [33] T. Rees, [Preconditioning Iterative Methods for PDE-Constrained Optimization](#), Ph.D thesis, University of Oxford, 2010.
- [34] S. Roggendorf, [Model Order Reduction for Linearized Systems Arising from the Simulation of Gas Transportation Networks](#), Master's thesis, Rheinischen Friedrich-Wilhelms-Universität Bonn, Germany, 2015.
- [35] Y. Saad, [Iterative Methods for Sparse Linear Systems](#), Society for Industrial and Applied Mathematics, Philadelphia, 2003.
- [36] P. Sonneveld and M. B. van Gijzen, [IDR\(s\): A family of simple and fast algorithms for solving large nonsymmetric systems of linear equations](#), *SIAM J. Sci. Comput.*, **31** (2008), 1035–1062.
- [37] M. C. Steinbach, [On PDE solution in transient optimization of gas networks](#), *J. Comput. Appl. Math.*, **203** (2007), 345–361.
- [38] M. Stoll and T. Breiten, [A low-rank in time approach to PDE-constrained optimization](#), *SIAM J. Sci. Comput.*, **37** (2015), B1–B29.
- [39] W. Q. Tao and H. C. Ti, [Transient analysis of gas pipeline network](#), *Chem. Eng. J.*, **69** (1998), 47–52.
- [40] E. F. Toro and S. J. Billett, [Centred TVD schemes for hyperbolic conservation laws](#), *IMA J. Numer. Anal.*, **20** (2000), 47–79.
- [41] M. B. van Gijzen and P. Sonneveld, [Algorithm 913: An elegant IDR\(s\) variant that efficiently exploits biorthogonality properties](#), *ACM Trans. Math. Software*, **38** (2011), 5:1–5:19.
- [42] A. J. Wathen, [Preconditioning](#), *Acta Numer.*, **24** (2015), 329–376.
- [43] M. Wathen, C. Greif and D. Schötzau, [Preconditioners for mixed finite element discretizations of incompressible MHD equations](#), *SIAM J. Sci. Comput.*, **39** (2017), A2993–A3013.
- [44] J. Zhou and M. A. Adewumi, [Simulation of transients in natural gas pipelines using hybrid TVD schemes](#), *Internat. J. Numer. Methods Fluids*, **32** (2000), 407–437.
- [45] A. Zlotnik, M. Chertkov and S. Backhaus, [Optimal control of transient flow in natural gas networks](#), in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, 4563–4570.

E-mail address: qiuyue@shanghaitech.edu.cn

E-mail address: grundel@mpi-magdeburg.mpg.de

E-mail address: martin.stoll@mathematik.tu-chemnitz.de

E-mail address: benner@mpi-magdeburg.mpg.de