

Efficient Numerical Methods for Gas Network Modeling and Simulation*

Yue Qiu[†], Sara Grundel[‡], Martin Stoll[§], Peter Benner[¶]

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 nonlinear differential algebraic equation (DAE). By introducing the concept of *long pipes*, we can reduce the dimension of the algebraic constraints in the resulting DAEs. We introduce a so-called *flow direction following* (FDF) ordering technique to order the *long pipes* of the network, and we obtain a block lower-triangular matrix structure of the (1,1) block for the system matrix of the DAE model. We further exploit such a matrix structure in the DAE model and 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 advantage of our methods.

Keywords: gas networks modeling, isothermal Euler equation, differential algebraic equation (DAE), preconditioning

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 the transient gas network dates back to the 80s [1, 2], the standard is to only compute stationary solutions of the gas network. This is also reasonable as the variation in a classically operated gas transportation networks sees no need for a truly transient simulation. However as we move from classical energy source 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 [1–5], optimization and control [6–8], time splitting schemes for solving the parabolic flow equations [9], discretization methods [10, 11], and model sensitivity study [12] to mention a few. It is obvious that efficient simulation techniques are needed both for design and for control. The early research in simulating gas flow in networks of pipes from the 80s and 90s, cf. [1–3] either made assumptions to simplify the mathematical model [1, 2], or made use of analogies between electrical networks and fluids for the network simulation [3]. Recent work on transient simulation of gas pipe networks either focus on the stationary or quasi-stationary computations [5, 13], using model order reduction technique to reduce the size of the network model [4, 14, 15], applying the explicit in time scheme to simplify the simulation of nonlinear dynamics of the network [16], or exploiting the model hierarchy

*This work is funded by the European Regional Development Fund (ERDF/EFRE: ZS/2016/04/78156) within the center dynamic systems (CDS).

[†]Corresponding author. Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39108, Magdeburg, Germany. E-mail: qiu@mpi-magdeburg.mpg.de, y.qiu@gmx.us

[‡]Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39108, Magdeburg, Germany. E-mail: grundel@mpi-magdeburg.mpg.de

[§]Technische Universität Chemnitz, Faculty of Mathematics, Reichenhainer Straße 41, 09107 Chemnitz, Germany. E-mail: martin.stoll@mathematik.tu-chemnitz.de

[¶]Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39108, Magdeburg, Germany. E-mail: benner@mpi-magdeburg.mpg.de

for model refinement in the simulation [17]. For other related work on gas networks, we refer to the active research groups of TRR154 [18].

The objective of this paper is to study the modeling of the dynamics for the gas networks and investigate fast numerical methods for the simulation of transient dynamics of gas networks. We apply the isothermal incompressible Euler equation to model the dynamics of gas transported through the network. We discretize the isothermal incompressible Euler equation using finite volume method (FVM) for all the pipelines in the network. We then assemble the sub-models for all the pipelines into a global model for the network. The obtained gas pipeline network model is represented by a nonlinear differential algebraic equation (DAE), which follows the rules of mass conservation, momentum conservation, and the conservation laws in the networks. We propose the concept of *long pipes*, which reduces the size of the algebraic constraints in the DAE, and this reduction also reduces the dimension of the DAE to be solved. We discretize the nonlinear DAE in time using the implicit Euler method. At each time step, we need to solve a large-scale nonlinear equation. We introduce the so-called *flow direction following* (FDF) ordering technique to order all the *long pipes* to obtain a block lower-triangular matrix structure for the system matrix of the DAE. By further exploiting the system structure, 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 the gas dynamics modeling of each pipeline of the network in Section 2, and we apply finite volume method (FVM) to discretize the incompressible isothermal Euler equation in Section 2. In Section 3, we introduce the details of gas networks modeling starting from assembling all pipelines. This results in a set of nonlinear differential algebraic equation (DAE) 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.

2.1 1D Isothermal Euler Equation

The dynamics of gas transported along pipes is described by the Euler equation, which represents the law of the conservation of mass, the conservation of momentum, and the conservation of energy. In this paper, we assume that the temperature is identical throughout the gas network. 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 change much. This assumption greatly reduces the complexity of modeling and is widely used in the simulation of gas networks [5, 13, 14, 19, 20]. For the heterogeneous temperature setting, current literature only focuses on the gas dynamics, i.e., the dynamics of the discretized PDE over a certain spatial domain, but not on the dynamics of the network. We refer to [21, 22] for further details.

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 and $\varphi = \rho v$ 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. 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 the

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 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, then $\gamma(T) = \gamma(T_0) = \gamma_0$, 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, we have

$$\frac{\gamma_0}{a^2}\frac{q^2 z_0(p)}{p} = \frac{v^2}{z_0(p)\gamma_0}p,$$

and the speed of sound in the gas or the acoustic wave velocity $v_s \approx \sqrt{z_0(p)\gamma_0}$. For the subsonic flow case, $v \ll v_s$, which is widely used in the gas transportation since supersonic gas flow also causes noise pollution, it is studied in [19] that

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

Therefore, the inertia term can be neglected and this neglect greatly simplifies the model, which is standard in the study of gas networks [4, 7, 8]. In this paper, we also use this neglect. Meanwhile, at the current stage, we assume that the elevation of pipes is homogeneous. The gravity term in (3b) vanishes. Note that the gravity term does not change the structure of the model after discretization of (3a) - (3b). This will be illustrated 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 [23].

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 ($z_0(p) = 1$) over spatial domain $[0, L]$ with homogeneous elevation. According to (4a) - (4b) we have

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

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

Here we denote γ_0 as c and $\lambda(q)$ as λ for short. 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 [24]. In this section, we apply the finite volume method (FVM) to discretize (5a) - (5b) together with the boundary conditions (6). For such an FVM discretization, we partition the domain as shown in Figure 1.

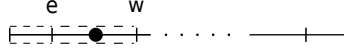


Figure 1: finite volume cells partition

In Figure 1, the left boundary of the control volume is denoted by ‘e’ while the right boundary of the control volume is denoted by ‘w’. For the isothermal incompressible Euler equation (5a) - (5b), we have two variables, i.e., the pressure p , and the mass flow q . Together with the boundary condition (6), we use two different control volume partitions for the pressure and mass flow nodes, which are shown in Figure 2.



Figure 2: control volumes partition for p and q

To apply the finite volume method (FVM) to discretize the PDE and the boundary condition, we integrate (5a) - (5b) over each control volume. To be specific, we integrate (5a) over the pressure control volume in Figure 2(a), and integrate (5b) over the mass flow control volume in Figure 2(b).

To integrate (5a) over the i -th control pressure control volume C_i , we have

$$\int_{C_i} \left(\partial_t p + \frac{c}{a} \partial_x q \right) dx = 0.$$

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 represent 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. This partition is shown in Figure 3.

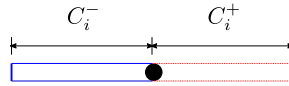


Figure 3: Separation of control volume C_i

Therefore, we get

$$\int_{C_i} \partial_t p \, dx + \int_{C_i^-} \frac{c}{a} \partial_x q \, dx + \int_{C_i^+} \frac{c}{a} \partial_x q \, dx = 0.$$

Furthermore by applying the midpoint rule, we get

$$\frac{h_{i-1} + h_i}{2} \partial_t p_i + \frac{c}{a_{i-1}} \left(q_i - \frac{q_{i-1} + q_i}{2} \right) + \frac{c}{a_i} \left(\frac{q_i + q_{i+1}}{2} - q_i \right) = 0,$$

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 junction nodes \mathcal{N}_j of the network. Here the supply nodes represent the set of nodes in the network where gas is injected into the network, and the demand nodes are a set of nodes where the gas is extracted from the network. In order to distinguish the junction nodes used in other works [15], we introduce the following definition of junction nodes.

Definition 3.1. *The junction nodes inside a network \mathcal{G} are a set of nodes \mathcal{N}_j that are neither supply nodes nor demand nodes, but nodes that connect at least three edges (pipes) together.*

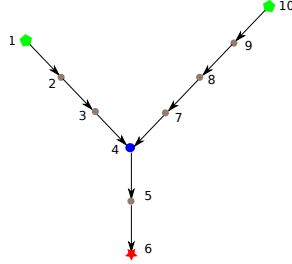


Figure 4: A typical gas network

An example network is shown in Figure 4. Here, node 1 and 10 denote the supply nodes, node 6 represents the demand node, all the other nodes connect pipes and redistribute gas in the network. According to Definition 3.1, only node 4 in Figure 4 is a junction node and the other nodes such as node 2 and 3 are not junction nodes. With this definition of the junction node, the number of junction nodes inside a network is reduced, and the number of algebraic constraints is also reduced, since the nodal conditions introduced later are only applied to junction nodes.

3.1 Nodal Conditions

There are two types of constraints concerning the connection of pipes, i.e., the pressure constraint, and the mass flow constraint. These two types of constraints describe the dynamics of gas at the junction nodes.

The so-called pressure nodal condition is given by

$$p_{\text{in}}^l = p_{\text{out}}^r, \quad (12)$$

where p_{in}^l denotes the pressure on the left boundary of the inflow pipes at the junction node i , and p_{out}^r represents the pressure on the right boundary of the outflow pipes at the same junction node i . The pressure nodal condition states that the pressure at the end of the outflow pipes should equal to the pressure at the beginning of the inflow pipes that connect to the same junction 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

$$\sum q_{\text{in}}^i = \sum q_{\text{out}}^i. \quad (13)$$

Here it states that the inflow at the junction node i should equal to the outflow at the same junction node i .

3.2 Network Assembling

In Section 2.2, we have obtained the discretized model in (11) to describe the dynamics of the gas transported along pipes through the gas network. Together with the nodal conditions in Section 3.1, we can obtain the global network model that describes the dynamics of the gas network. Before this, we would like to introduce the concept of the so-called *long pipes*, which reduces the computational complexity of the network simulation.

Definition 3.2. *The pipe in a gas network is called a long pipe, if it starts with a supply node, ends up with a junction node or a demand node, or it starts with a junction node, ends up with another junction node, or it starts with a junction node, and ends up with a demand node.*

The advantage of introducing the concept of *long pipes* is that it will reduce the computational cost for the network simulation, which will be shown by the numerical experiments. Here, we use the network depicted in Figure 4 as a simple example to illustrate the *long pipes*. There are 3 *long pipes* in the network. The first one is the pipe that starts with node 1 and ends up with node 4, the second is the pipe starts with node 10 and ends up with node 4, while the third is the pipe that starts with node 4 and ends up with node 6. The network in Figure 4 has 9 pipes that are connected, and it is natural to use the *long pipes* instead of the “natural pipes” to represent the network since the *long pipes* concept represents the network topology.

The discretized model (11) describes the dynamics of gas transported along one single pipe. To obtain such a model, we need two boundary conditions (6) for each pipe, i.e., the prescribed pressure at the input node, and the prescribed mass flow at the demand node. However, for the gas network, there are not enough boundary conditions. Take the network in Figure 4 for example, we only have the prescribed pressure at the supply nodes 1, 10, and the prescribed mass flow at the demand node 6. However, we have three *long pipes*, while we need 2 boundary conditions for each *long pipe*. Therefore, we need to introduce extra variables to get enough boundary conditions. We need extra equations by applying (12) - (13) for the extra introduced variables to obtain a proper model. We take the network in Figure 4 as an example to introduce such a procedure.

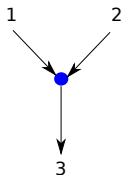


Figure 5: *long pipes* of network in Figure 4

The *long pipe* 1 and 2 are supply pipes, and the pressure at the inflow node is prescribed. Therefore, we need to introduce the mass flow at the outflow node for each pipe, which is represented by q^1 and q^2 , respectively. The *long pipe* 3 is the demand pipe, and the mass flow at the outflow node is given. We need to introduce the variable p^3 to represent the pressure at the inflow node of *long pipe* 3. We can apply the nodal condition (12) without introducing a new variable p^3 , which reduces the number of introduced variables and the computational cost. By using the single pipe model (11), we obtain the mathematical model for network in Figure 4,

$$\underbrace{\begin{bmatrix} \mathcal{M}_1 & & & & \\ & \mathcal{M}_2 & & & \\ & & \mathcal{M}_3 & & \\ & & & 0 & \\ & & & & 0 \end{bmatrix}}_{\mathbf{M}} \frac{\partial}{\partial t} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ q^1 \\ q^2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathcal{K}_1 & & \mathcal{B}_q^1 & & \\ & \mathcal{K}_2 & & & \mathcal{B}_q^2 \\ \bar{\mathcal{B}}_p^3 & & \mathcal{K}_3 & & \\ e_1 & e_2 & e_3 & 1 & 1 \\ & & & & 0 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ q^1 \\ q^2 \end{bmatrix} + \underbrace{\begin{bmatrix} \mathcal{B}_p^1 \\ & \mathcal{B}_p^2 \\ & & p_s^1 \\ & & & p_s^2 \end{bmatrix}}_{\mathbf{B}_p} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \mathcal{B}_q^3 \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{B}_q} q_d^3 + \underbrace{\begin{bmatrix} \mathcal{G}_1(u_1, p_s^1) \\ \mathcal{G}_2(u_2, p_s^2) \\ \mathcal{G}_3(u_3, p^3) \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{g}^*}. \quad (14)$$

Here u_1, u_2, u_3 represent the variables to be resolved for pipe 1, 2, 3, respectively, and $\bar{\mathcal{B}}_p^3 = [0, 0, \dots, 1, 0, \dots] \otimes \mathcal{B}_p^3$. The mass flow nodal condition is represented by the 4th block row in (14), and the pressure nodal condition is given by the 5th block row and also the (3, 1) block of \mathbf{K} in (14). 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 (12) - (13).

Note that the matrix \mathbf{K} in (14) is not unique. This is because that we make $p^3 = p_e^1$, where p_e^1 is the pressure at the end node of *long pipe* 1, and the second pressure nodal condition is obtained via making $p_e^1 = p_e^2$ at the fifth block row in (14). We can also make $p^3 = p_e^2$, and this in turn moves $\bar{\mathbf{B}}_p^3$ 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 the pressure nodal conditions. 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, 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 3.1. *Suppose that the network $\mathcal{G} = (\mathcal{E}, \mathcal{N})$ is a connected graph, and has n_s supply pipes, n_d demand pipes, and n_j junction pipes. Here junction pipes are long pipes that both start with and end up with junction nodes. Then the following relation between the number of extra variables n_e and the number of extra algebraic constraints n_a hold,*

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

Proof. As stated before, we only introduce extra variables for the mass flow of each *long pipe* when necessary, 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 flow at the demand pipes are already prescribed. The number of extra variables are the mass flows at the end of each pipe that is not a demand pipe.

The algebraic constraints are obtained via applying nodal conditions at the junction nodes. Suppose that the junction node i has n_{in}^i injection pipes, and n_{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 network assembling. We have one algebraic constraint to prescribe the mass flow nodal condition for junction node i . Therefore, we need n_{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.

$$n_a = \sum_i n_{\text{in}}^i,$$

algebraic constraints. Meanwhile,

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

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

Proposition 3.1 states that the total number of extra variables equal to the total number of algebraic constraints. This is very important for us to simulate the network model in the form of (14), which will be shown in the next section.

4 Fast Numerical Methods for Simulation

By introducing the concept of *long pipes* and assembling all the *long pipes* of the network, we obtain the model to represent the gas network with such a structure as in (14). In this section, we will introduce fast numerical algorithms for the simulation of such a model.

4.1 Numerical Algorithms to Solve Differential Algebraic Equations

Here, we reuse notations of (14) and simplify them. We obtain the general mathematical model,

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

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 nonlinear differential algebraic equation (DAE). Meanwhile, the size of the DAE (15) is also huge, since it contains several discretized isothermal Euler equations (5) over the network domain. To solve/simulate such a DAE model is challenging. 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 [4], or reducing the so-called tractability index [15]. However, these methods are problem dependent, and demand deep insight into the network. Here, we propose a fast numerical method that is general for the gas network without decoupling the DAE or reducing the DAE tractability index.

To simulate the DAE model (15), we discretize in time using the implicit Euler method, and at time step k we have,

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

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

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

at each time step k to compute the solution x^k . To solve such a nonlinear equation for the simulation of gas networks, some related work [16, 27] treat the nonlinear term $f(x, u^k)$ explicitly, i.e., $f(x^k, u^k) \approx f(x^{k-1}, u^k)$ since the input u^k is known. This explicit approximation of the nonlinear term reduces the computational complexity, and yields a linear system. However, this approximation typically does not accurately represent the nonlinear dynamics of the network. Here, we treat this nonlinear term implicitly and apply Newton's method to solve the nonlinear system (16) to study the nonlinear dynamics of the network. Newton's method is described by Algorithm 1.

Algorithm 1 Newton's method to solve (16)

- 1: **Input:** maximal 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: $x_m \leftarrow x$, $m \leftarrow m + 1$
 - 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 generalized minimal residual (GMRES) method [28] or induced dimension reduction (IDR(s)) method [29] are then appropriate to solve such a system. To accelerate the convergence of such a Krylov subspace method, we need to apply the preconditioning technique by exploiting the structure of the Jacobian matrix $D_F(x)$.

4.2 Preconditioning Technique

The Jacobian matrix

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

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

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

Here \bar{M} is block-diagonal, and the second block row of A comes from the algebraic constraints of the networks by applying the nodal conditions introduced in Section 3. Meanwhile, the size of A_{11} is much bigger than the size of A_{22} since A_{11} comes from the discretization of isothermal Euler equations over all the *long pipes* of the network, while the size of A_{22} is equal to $n_s + n_j$ according to Proposition 3.1. Moreover, the partial derivative of the nonlinear term $\frac{\partial}{\partial x} f(x, u)$ has the following 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 (14). This can also be verified by checking the structure of the nonlinear term in (14). By additionally introducing the concept of *flow direction following* (FDF) ordering of the *long pipes* of the network, we can further exploit the structure of $D_F(x)$ for fast computations.

Before applying the *flow direction following* ordering, we assume that the directed graph that we use to represent the gas network is already re-directed so that there are no directed cycles inside the network. When there are directed cycles, this re-direction can be made by changing the prescribed flow direction at the corresponding edges of the cycle to eliminate the directed cycle. This is feasible because the directions of the edges in the graph are set to represent the pre-defined flow direction inside the network, which makes the modeling easier, while the real gas flow direction at some edges may be opposite to the pre-defined direction. Numerical experiments in the next section illustrates this. However, by re-directing the graph to eliminate cycles, we can exploit the structure of the mathematical model to make fast computations, which reduces the computational complexity for the simulation of such a gas network.

Definition 4.1 (*Flow direction following (FDF) ordering*). *Given a directed graph such that all long pipes are directed in such a way that we do not have cycles in the graph and we can order the pipes along the directions that point away from supply and towards demand nodes. This means for any path one can take starting from a supply node to a demand node following the direction of the graph, the index of the pipes have to increase along that path.*

After the FDF ordering, the pipe that is in the downstream direction of the flow automatically gets a bigger number for the indexing of all the *long pipes* of the gas network. Note that the FDF ordering is not unique, and to perform such an ordering of all the pipes, we can start from any of the supply pipes. Then we have the following proposition to illustrate the structure of the partial derivative of the nonlinear term (19).

Proposition 4.1. *By applying the FDF ordering, D_f in (19) has a block lower-triangular structure.*

Proof. The i -th block in f has the structure,

$$f_i = \begin{bmatrix} 0 \\ g(x^{(i)}, p_{\text{in}}) \end{bmatrix},$$

where the structure of $g(x^{(i)}, p_{\text{in}})$ is given by (11). Here, $x^{(i)} = [p^{(i)T} \quad q^{(i)T}]$, and

$$p_{\text{in}} = \begin{cases} p_s, & \text{when } i\text{-th pipe is a supply pipe} \\ p_{\text{out}}^{(j)}, & \text{when } i\text{-th pipe is connected by a junction node with pipe } j \end{cases}.$$

Then the diagonal block of $\frac{\partial}{\partial x_i} f_i$ becomes,

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

while

$$\frac{g(x^{(i)}, p_{\text{in}})}{\partial p_{\text{in}}} = \begin{cases} 0, & \text{when } i\text{-th pipe is a supply pipe} \\ \frac{\partial g(x^{(i)}, p_{\text{in}})}{\partial p_{\text{out}}^{(j)}}, & \text{when } i\text{-th pipe is connected by a junction node with pipe } j \end{cases},$$

is the (i, j) block of D_f . Here $p_{\text{out}}^{(j)}$ is the pressure at the end of the pipe j . According to Definition 4.1, we have $i > j$. Therefore, the nonzero off-diagonal block lies in the lower-triangular part. \square

Similar to Proposition 4.1, we can also show that the $(1, 1)$ block of A in (18) has a lower-triangular block structure.

Proposition 4.2. *After the FDF ordering, A_{11} in (18) also has a block lower-triangular structure.*

Proof. For the i -th block row of A_{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 (12) is applied for the i -th pipe. According to Definition 4.1, we have $i > j$, and this completes the proof. \square

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

Theorem 4.3. *By applying the FDF ordering to all the long pipes of the network, the $(1, 1)$ block of the Jacobian matrix (17) has a block lower-triangular structure.*

Proof. According to (17) (18) (19), the $(1, 1)$ block of the Jacobian matrix $D_F(x)$ is,

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

According to Proposition 4.1 and Proposition 4.2, A_{11} and D_f are block lower-triangular matrices after the FDF ordering. Since \bar{M} is a block diagonal matrix, this completes the proof. \square

Next, we use a benchmark network in [30] 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 FDF ordering. The network parameters are also given in [30], and we use the concept of *long pipes* in this paper to discretize the network. 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 FDF ordering are given by Figure 7.

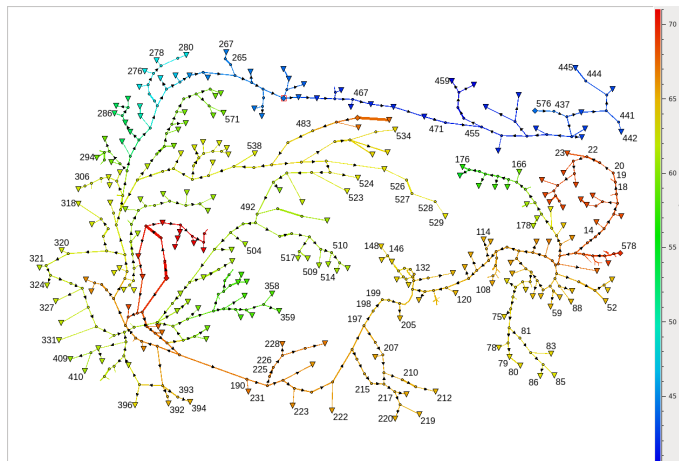


Figure 6: Big benchmark network in [30]

Figure 7 shows that the Jacobian matrix $D_F(x)$ is a sparse matrix. After applying the FDF 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 200348×200348 block lower-triangular matrix while the size of the $(2, 2)$ block is 417×417 . In general for a certain network after discretization, the size of the $(2, 2)$ block of the Jacobian matrix is fixed since it equals the number of algebraic constraints. According to Proposition 3.1, it equals $n_s + n_j$ where n_s are the number of supply pipes and n_j is the number of junction pipes.

These two variables are fixed for a certain network. The size of the $(1, 1)$ block equals two times the grid points of the network. Therefore, it is much bigger than $n_s + n_j$, which is clearly shown in Figure 7.

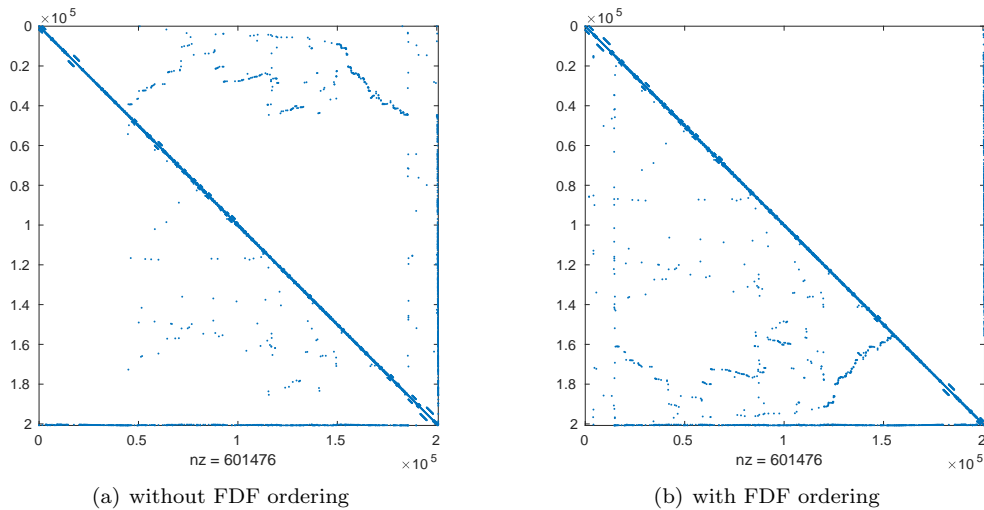


Figure 7: Sparsity pattern of J without and with FDF ordering

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 to simulate the discretized gas network model, we need to apply Algorithm 1, while 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 the generalized saddle-point structure [31]. This enables us to make use of the preconditioning techniques designed for the generalized saddle-point systems to solve the Jacobian system. Generalized saddle-point systems come from many applications, such as computational fluid dynamics [32], PDE-constrained optimization [33], optimal flow control [34]. Many efforts have been dedicated to the efficient numerical solution of such systems using preconditioning techniques [35–39], we recommend [31, 40] for a general survey of the preconditioning generalized saddle-point systems.

We can compute a block LU factorization by

$$D_F(x) = \begin{bmatrix} D_{F_{11}} & \\ D_{F_{21}} & S \end{bmatrix} \begin{bmatrix} I & D_{F_{11}}^{-1} D_{F_{12}} \\ & I \end{bmatrix}. \quad (20)$$

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 4.3, $D_{F_{11}}$ has a block lower-triangular structure, and the size of $D_{F_{22}}$ is much smaller than $D_{F_{11}}$. Therefore, we can compute the Schur complement S by block forward substitution, and apply the following 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 $\lambda(P^{-1} D_F(x)) = \{1\}$. Moreover, the preconditioned matrix $P^{-1} D_F(x)$ has minimum polynomial of degree 2, so that a method like generalized minimum residual (GMRES) [28] would converge in at most two steps [31].

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 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.

Since P_1 is a good preconditioner for $D_F^1(x_1)$, and if the Jacobian matrix $D_F(x)$ at the other Newton iterations and other time steps is close to $D_F^1(x_1)$, then P_1 is also a good preconditioner for solving the Jacobian systems at other Newton iterations of other time steps. This is true for the 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, and it will be shown 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 in the term of total computational time.

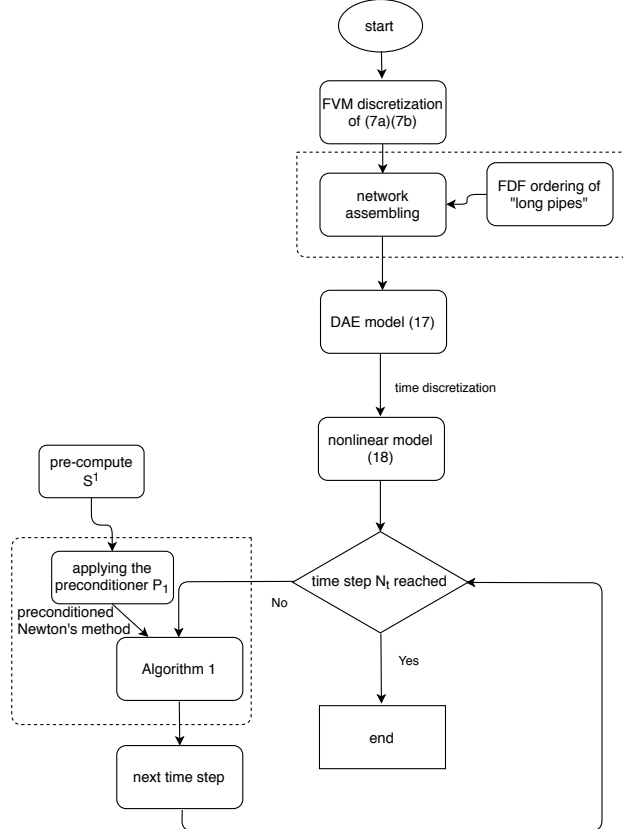


Figure 8: Computational diagram for gas network simulation

By applying the preconditioner P_1 in (22), we show the computational diagram to illustrate the process of gas network simulation in Figure 8.

5 Numerical Results

In this section, we report the performance of our numerical algorithms for the simulation of the gas networks. We apply our numerical algorithms to the benchmark problems of several gas networks given in [4, 14–16] 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 5.1. Parameter settings for this pipeline network are given in [4].



Figure 9: Pipeline network in [15]

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 10, where the x -axis represents the mesh sizes in meters.

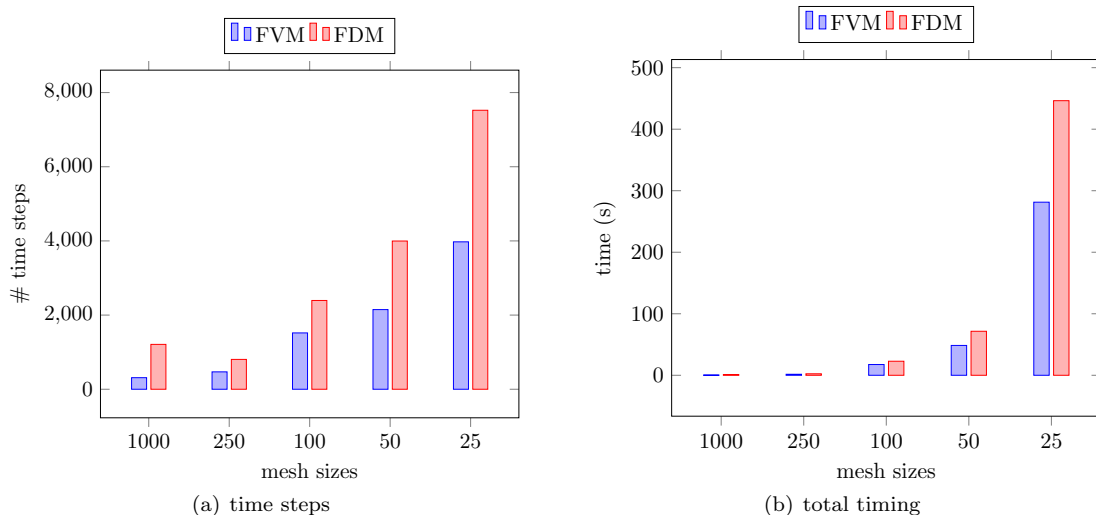


Figure 10: Comparison of FVM and FDM for a single pipe network

Figure 10(a) shows the number of time steps that `ode15s` uses to simulate the ODEs given by FVM and FDM discretization over the time horizon $[0, 10^5]$. It is clear that for every mesh size, we need less time steps for the ODE given by the FVM discretization than the ODE given by the FDM discretization while the cost of the simulation for each time step is the same for the two ODEs since the same mesh size for discretization yields the ODEs of the same size. This makes the total computation time for the simulation of ODE given by the FVM discretization shorter than that of the ODE given by the FDM discretization, which is also shown in Figure 10(b).

The results in Figure 10 show that the ODE model given by the FVM discretization is faster to be solved by the `ode15s` routine in MATLAB than the ODE model given by the FDM discretization.

However, the background mechanism is still not clear since `ode15s` behaves like a black-box. 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 11, where the network parameters are given in [15].

We use the FVM and FDM method to discretize the network in Figure 11 and apply the computational method depicted in Figure 8. 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 12.

The mass flow at node 57 computed by using different discretized DAE models (16) is plotted in Figure 15(a), and they have similar dynamical behavior. However, when we look at the dynamics of the mass flow at the first 5 seconds, we can see quite a big difference in Figure 15(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.

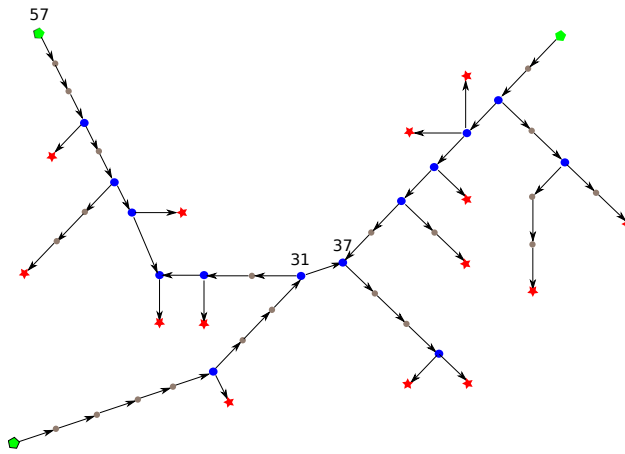


Figure 11: Medium size network

The computational results given by Figure 10 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.

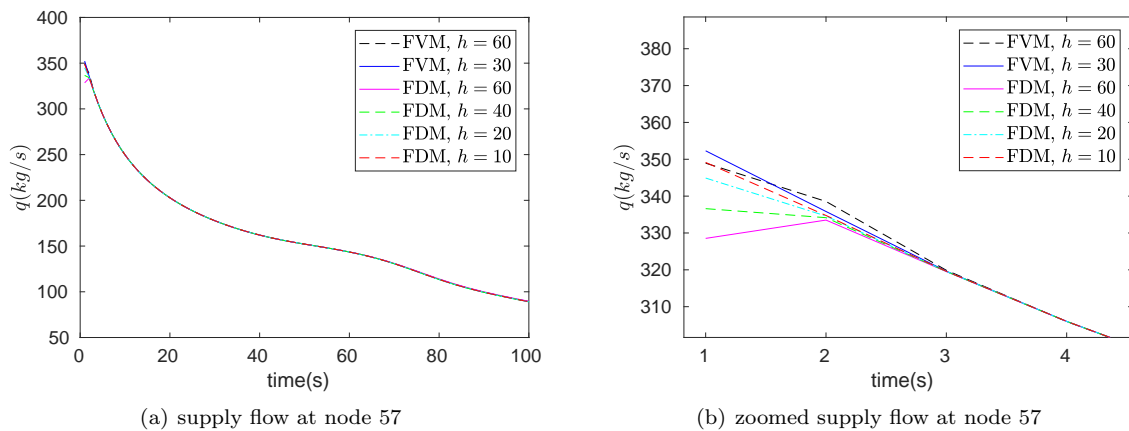


Figure 12: Comparison of FVM and FDM for a medium network

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 Figure 13. 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.

Computational results in Figure 10 - 13 show that the finite volume method has a bigger advantage over the finite difference method. 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 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 [11], where the authors preferred FDM due to the comparable accuracy with FEM and less computational time.

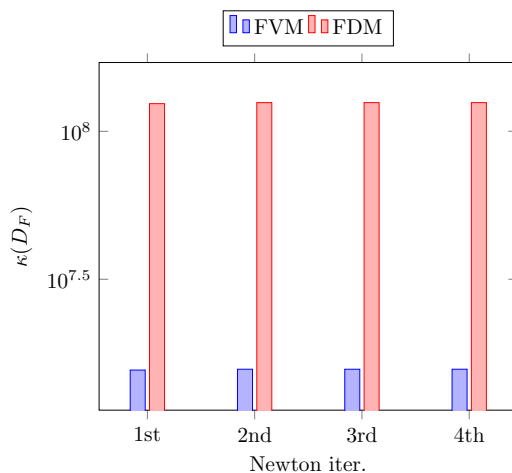


Figure 13: Condition number of the Jacobian matrix, 1st time step, $h = 60$

5.2 Robustness of Modeling

The flow direction inside the gas network plays an important role for the modeling and fast simulation of such a network. This is because the setup of the algebraic constraints, i.e., applying the nodal conditions (12) - (13), and the *flow direction following* (FDF) ordering are closely related with the flow direction. However, the flow direction may change due to the change of operation conditions of the gas networks. In this part, we show that we only need a prescribed flow direction for our numerical methods, and the flow direction change does not influence the performance of our methods. This is because the FDF ordering at the pre-processing stage only needs a prescribed flow direction and helps us to rearrange the algebraic constraints so that we can exploit the system structure for fast computation. The change of the flow direction does not change the mathematical formulation of algebraic constraints. Therefore, the structure of the system stay unchanged with respect to the change of the flow direction. This means that we do not require the foreknowledge of the flow direction, which sometimes is also difficult to know for the large network, and our methods are robust with respect to the change of flow direction.

First, we test two different cases, which corresponds to two different flow direction profiles of the network, cf. Figure 4. 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 Figure 14 - 15.

Figure 14 shows that the mass flow at both supply nodes approaches to 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 15. The mass flow at the 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 it gets gas flow from the network. For both cases, the equality $q_s^1 + q_s^2 = \sum q_d$ holds at the steady state.

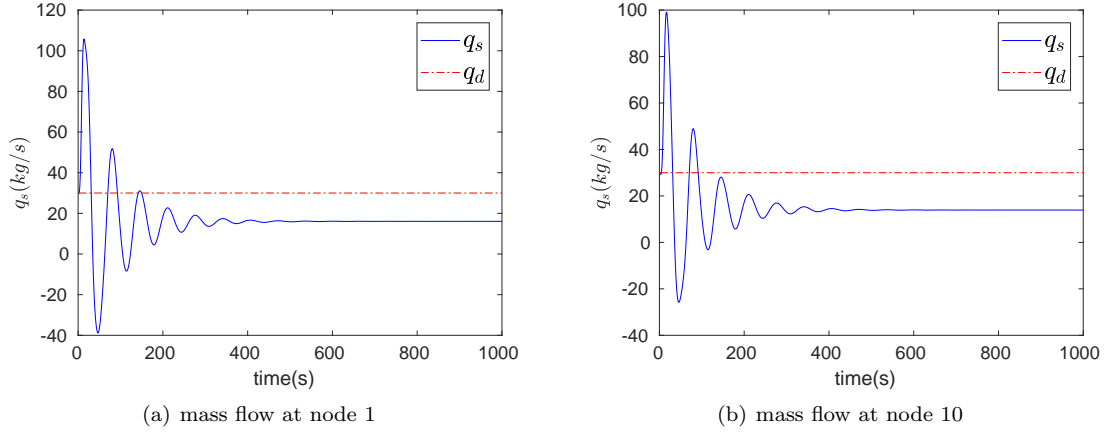


Figure 14: Mass flow at supply nodes for case 1

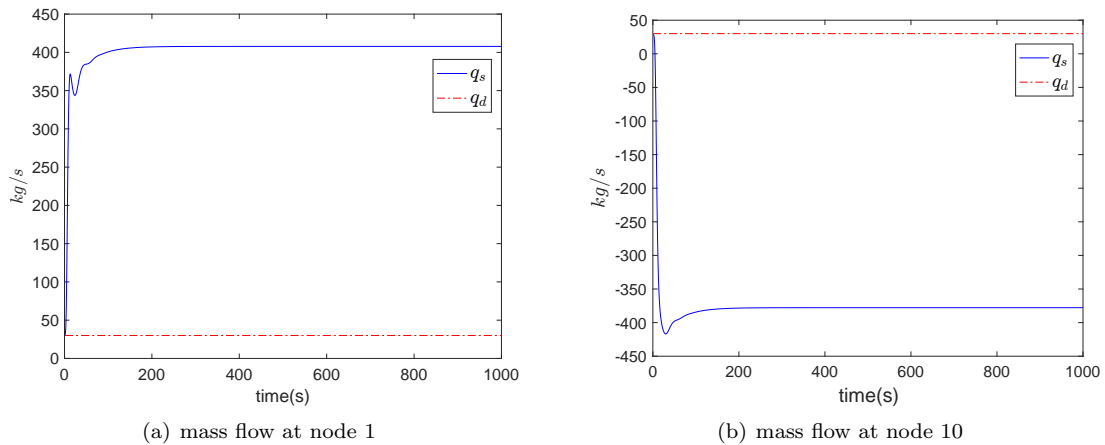


Figure 15: Mass flow at supply nodes for case 2

We also apply two different cases to a more complicated network given in Figure 11 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 are 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 are given by Figure 16. The initial conditions of the gas network for the simulation of the two different cases are set the same.

The simulation results in Figure 16 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 show that the flow can travel in a direction opposite to the prescribed flow direction, and the inflow at node 31 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 Figure 14 - 16 show that our methods are robust with respect to the change of flow direction. We do not need the detailed information of the flow direction profile of the network, instead we can just assume a predefined flow direction of the network and use this predefined flow direction for the modeling and simulation. The computations will update the flow direction automatically.

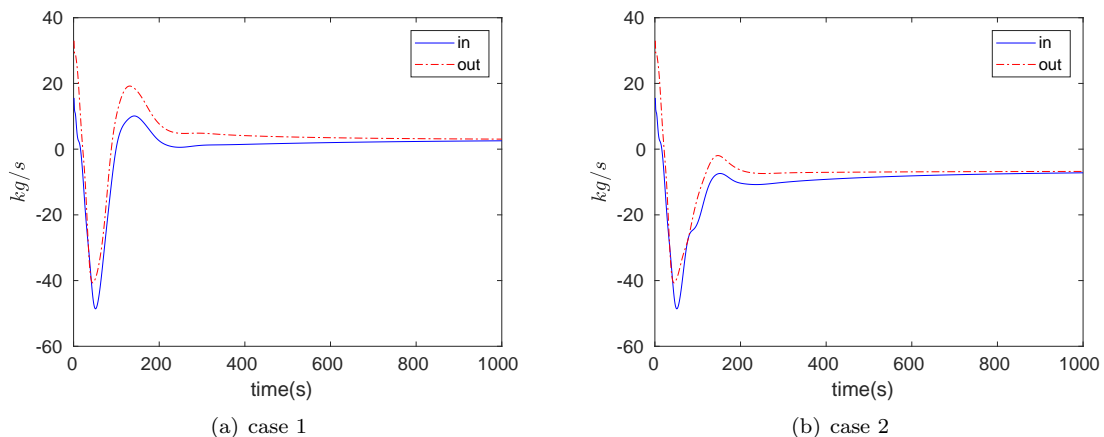


Figure 16: Mass flow for the pipe 31 → 37

The mechanism of the robustness of the modeling with respect to gas flow direction contains two parts. First, the algebraic constraints for the modeling are invariant under the change of flow direction in the network. Second, the boundary conditions given by (6) also suit the case when the flow changes direction. Therefore, the additional variables, which make all *long pipes* in the network have both left and right boundary conditions, are also invariant under the change of flow direction. We do not need to introduce another set of variables and switch to another model when the flow changes direction. This is the key for us to apply the FDF ordering at the pre-processing stage to get a block lower-triangular structure of the (1, 1) block of the system matrix.

5.3 Convergence Comparison

After the FDF ordering, we apply Algorithm 1 to solve the nonlinear equation (16) at each time step. Newton’s method requires the computation of the Jacobian matrix at each iteration, which is typically expensive. We try to reduce the cost of computing the Jacobian matrix, and we approximate the nonlinear term $\frac{q_m^k |q_m^k|}{q_m^k}$ at the m -th Newton iteration of time step k as,

$$\frac{q_m^k |q_{m-1}^k|}{p_{m-1}^k} \approx \frac{q_m^k |q_m^k|}{p_m^k}.$$

This approximation avoids computing the partial derivatives of the nonlinear term, and results in a diagonal coefficient matrix for the linearization. We apply this approximation to solve the nonlinear equation iteratively, and this is the Picard iteration. Here, we study the convergence of both Newton’s method and the Picard method for solving (16). We plot the 2-norm of the nonlinear residual, i.e., $\|F\|_2$ of the first and 50-th time step at each Newton and Picard iteration. Both Newton and Picard iteration start with the same initial condition and the time step size for both methods are set as $\tau = 1$.

We first study the convergence for the simulation of the network shown in Figure 11. We discretize this network using the finite volume method with mesh size $h = 50$, and both Newton’s method and the Picard method are stopped when $\|F\|_2 \leq 10^{-5}$. The results given in Figure 17 show that both the Newton and Picard iteration have a fast convergence rate. However, the Picard method takes more than 2 times the number of iterations to reach the same stopping criterion. This means that we need to solve more than 2 times the linear systems for the Picard method than the Newton’s method, while solving such a linear system is the most time consuming part of such a nonlinear iteration. Moreover, the sizes of the linear systems at each Newton and Picard iteration are the same. The additional cost from the Picard method is much bigger than the cost saved from the simplification of the derivatives computation. This makes the Picard method not as practical as the Newton’s method for solving such a nonlinear system.

Next, we test on a bigger network given in Figure 6. We also discretize such a network with the finite volume method with $h = 50m$, and we stop both the Newton and Picard iteration when $\|F\|_2 \leq 10^{-4}$. The results are given by Figure 18. Again, we observe similar convergence behavior

for Newton’s method and the Picard method with the convergence results shown in Figure 17. The Picard method needs more than double the number of linear system solves than Newton’s method. When we need more accurate simulations of a gas network, smaller mesh sizes are necessary increasing the difference in the computational effort.

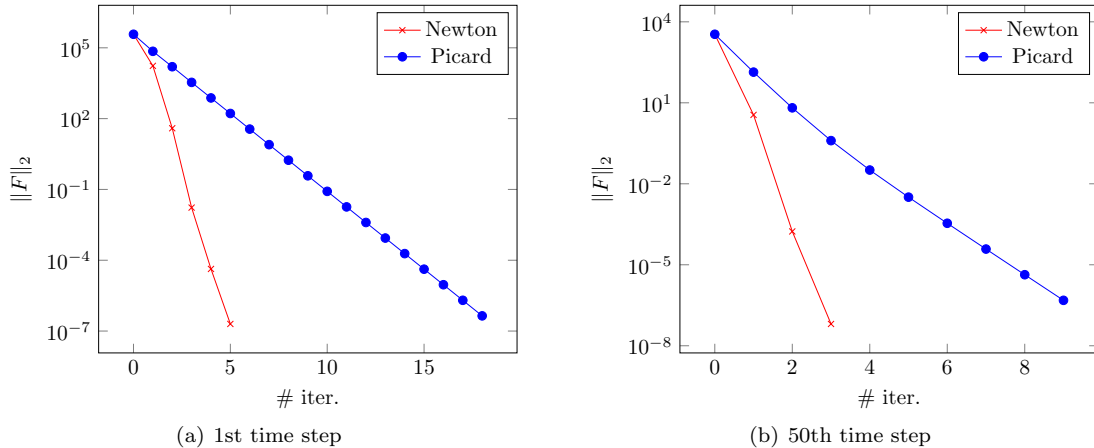


Figure 17: Convergence comparison for network in Figure 11

Computational results in Figure 17 and 18 show that for the simulation of gas networks, Newton’s method is superior to the Picard method in both the computational complexity and the convergence rate.

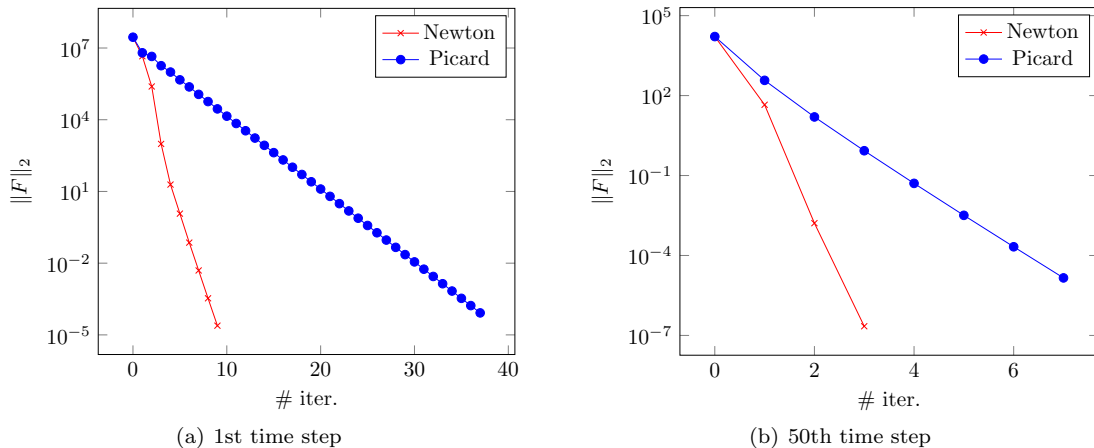


Figure 18: Convergence comparison for network in Figure 6

5.4 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 size 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 on the network in Figure 6 using different mesh sizes for the finite volume discretization. At each Newton (outer) iteration, we solve a linear system by applying an (inner) Krylov solver, e.g., the IDR(s) solver [29, 41], 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 with the forcing term for inexact Newton's method [42]. Here, we show the convergence of the Newton-Krylov method with respect to different ε_{tol} , i.e., $\|F\|_2$ with respect to different settings of ε_{tol} . We use the “true” residual computed by using 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 [43].

The computational results of the nonlinear residual $\|F\|_2$ in Figure 19 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 if the inner loop is solved accurately, or 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 accuracy are the same for the 10-th time step. If lower inner loop accuracy is used, less computational effort is needed. This reduces the computational complexity. The number of IDR(4) iterations for different inner loop accuracy are reported in Figure 20 - 21.

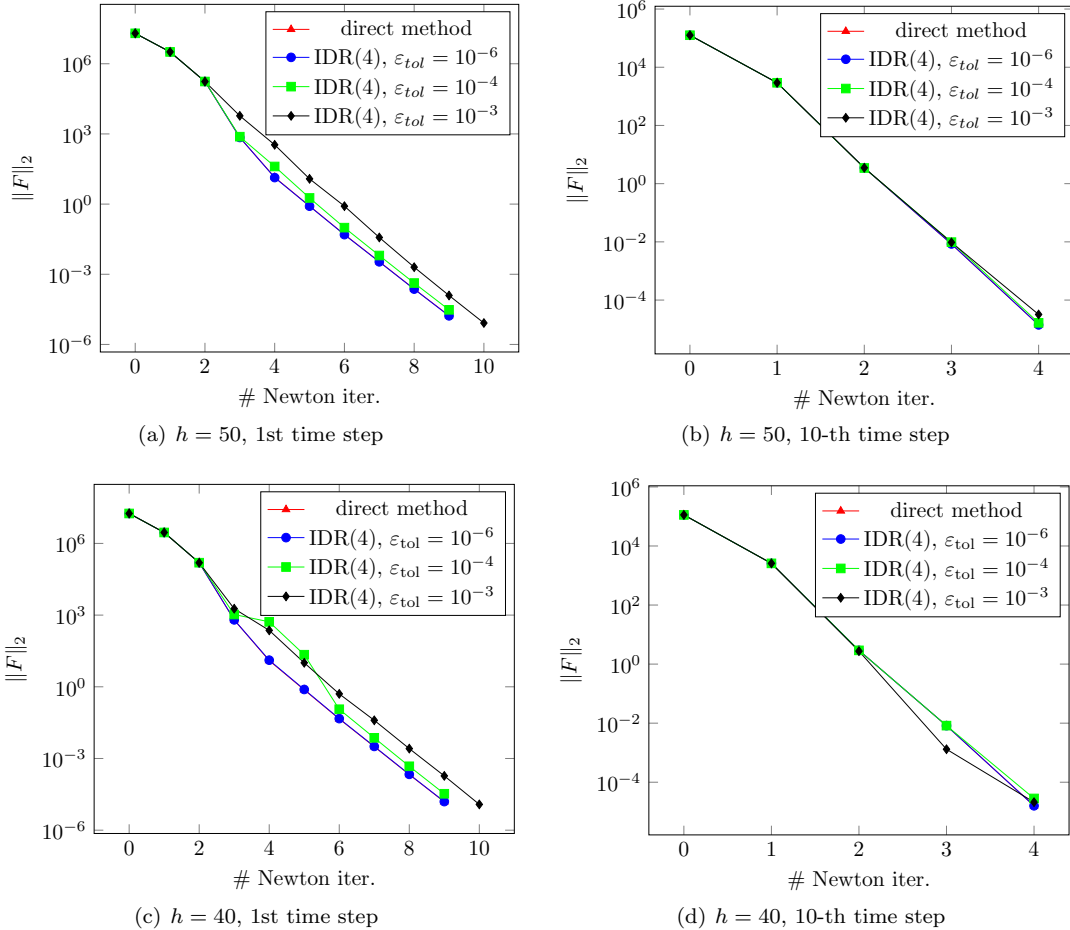


Figure 19: Nonlinear residual at the first and tenth time step

The computational results in Figure 20(a) show that the total number of IDR(4) iterations (47) for $\varepsilon_{tol} = 10^{-6}$ is almost twice of the total number of IDR(4) iterations (24) for $\varepsilon_{tol} = 10^{-3}$. This represents 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 by Figure 20(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 21. At this stage, IDR(4) with $\varepsilon_{\text{tol}} = 10^{-3}$ still needs less work than the IDR(4) with $\varepsilon_{\text{tol}} = 10^{-6}$ but is no longer as significant.

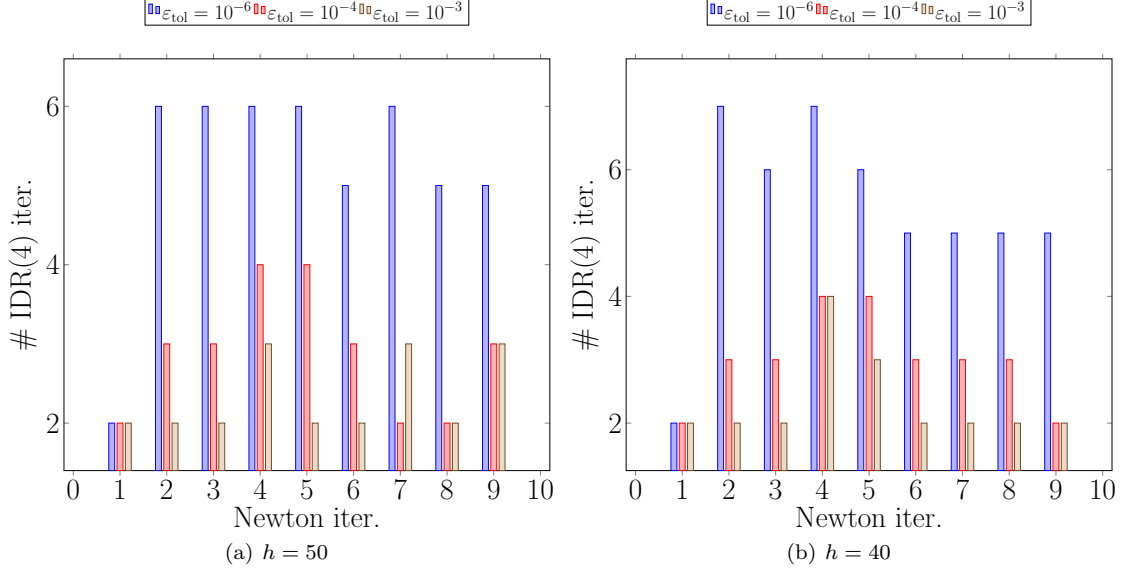


Figure 20: Number of IDR(4) iterations at the first time step

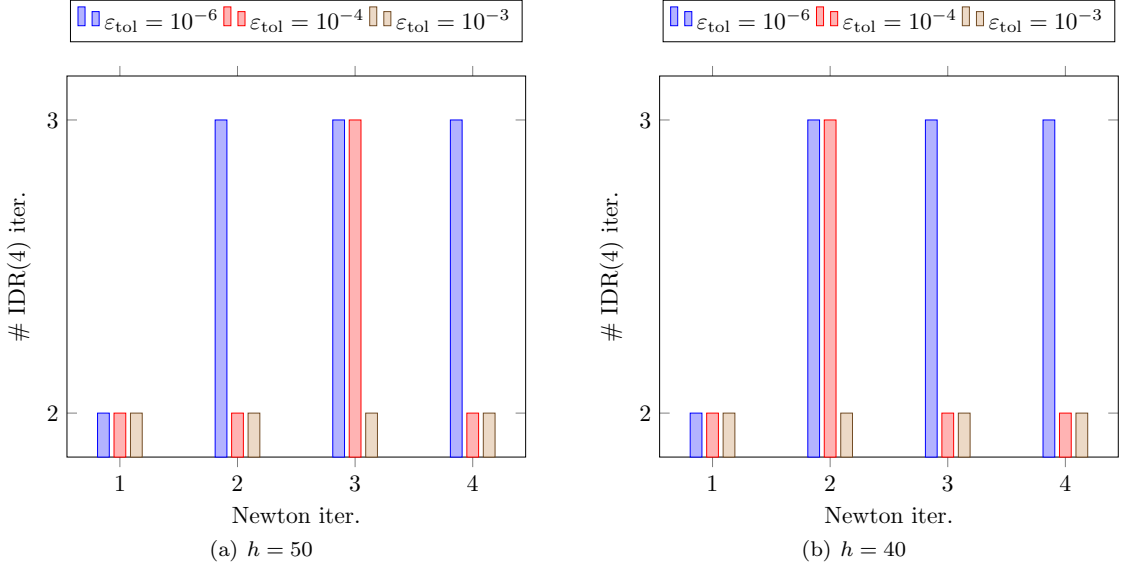


Figure 21: Number of IDR(4) iterations at the 10-th time step

We have already observed that the performance of our modeling is robust and convergence of the preconditioned Krylov solver happens quickly. There are no other alternatives currently found in the literature. 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 1. 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 time are measured in seconds.

The computational results in Table 1 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 that there is a big overhead when applying the preconditioned IDR(4) solver while the `backslash` operator is highly optimal for smaller systems. The time to compute the Schur complement in preconditioner P_1 scales almost linearly with the system sizes.

Table 1: Computational time for the 1st Newton iteration

h	$\#D_F$	t_{S^1}	IDR(4)	<code>backslash</code>
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	-

6 Conclusions

In this paper, we studied the modeling and simulation of pipeline gas networks. We applied the finite volume method (FVM) to discretize the incompressible isothermal Euler equation, and compared with the finite difference method (FDM). Numerical results show the advantage of the FVM over the FDM. To model the gas networks, we introduced the concept of *long pipes*, which represents the topology of the network interconnection and reduces the size of the algebraic constraints of the resulting differential algebraic equation (DAE). To simulate such a DAE system, we proposed the *flow direction following* (FDF) ordering of the *long pipes* of the network. Through such an FDF 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] A. Osiadacz. Simulation of transient gas flows in networks. *Internat. J. Numer. Methods Fluids*, 4(1):13–24, 1984.
- [2] A. Osiadacz. *Simulation and analysis of gas networks*. Gulf Publishing, Houston, TX, 1987.
- [3] W. Q. Tao and H. C. Ti. Transient analysis of gas pipeline network. *Chem. Eng. J.*, 69(1):47–52, 1998.
- [4] S. Grundel, N. Hornung, and S. Roggendorf. Numerical aspects of model order reduction for gas transportation networks. In *Simulation-Driven Modeling and Optimization*, volume 153 of *Springer Proceedings in Mathematics & Statistics*, pages 1–28. 2016.
- [5] M. Gugat, F. M. Hante, M. Hirsch-Dick, and G. Leugering. Stationary states in gas networks. *Netw. Heterog. Media*, 10(2):295–320, 2015.
- [6] M. C. Steinbach. On PDE solution in transient optimization of gas networks. *J. Comput. Appl. Math.*, 203(2):345–361, 2007.
- [7] 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)*, pages 4563–4570, Dec 2015.

- [8] 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*, pages 77–122. Springer Verlag, Singapore, 2017.
- [9] J. Zhou and M. A. Adewumi. Simulation of transients in natural gas pipelines using hybrid TVD schemes. *Internat. J. Numer. Methods Fluids*, 32(4):407–437, 2000.
- [10] H. Egger. A robust conservative mixed finite element method for isentropic compressible flow on pipe networks. *SIAM J. Sci. Comput.*, 40(1):A108–A129, 2018.
- [11] A. J. Osiadacz 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(2):79–85, 1989.
- [12] M. Chaczykowski. Sensitivity of pipeline gas flow model to the selection of the equation of state. *Chem. Eng. Res. Des.*, 87(12):1596 – 1603, 2009.
- [13] M. Herty, J. Mohring, and V. Sachers. A new model for gas flow in pipe networks. *Math. Methods Appl. Sci.*, 33(7):845–855, 2010.
- [14] 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, pages 189–212. 2013.
- [15] 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, pages 183–205. 2014.
- [16] 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)*, pages 4579–4584, 2015.
- [17] P. Domschke, A. Dua, J. J. Stolwijk, J. Lang, and V. Mehrmann. Adaptive refinement strategies for the simulation of gas flow in networks using a model hierarchy. *Electron. Trans. Numer. Anal.*, 48:97–113, 2018.
- [18] SFB Transregio 154: Mathematical modelling, simulation and optimization using the example of gas networks. <https://trr154.fau.de/index.php/en/>.
- [19] A. Herrán-González, J. M. De La Cruz, B. De Andrés-Toro, and J. L. Risco-Martín. Modeling and simulation of a gas distribution pipeline network. *Appl. Math. Model.*, 33(3):1584 – 1600, 2009.
- [20] A. Fügenschuh, B. Geißler, R. Gollmer, A. Morsi, J. Rövekamp, M. Schmidt, K. Spreckelsen, and M.C. Steinbach. *Chapter 2: Physical and technical fundamentals of gas networks*, pages 17–43. Society for Industrial and Applied Mathematics, Philadelphia, 2015.
- [21] K. S. Shterev and S. K. Stefanov. Pressure based finite volume method for calculation of compressible viscous gas flows. *J. Comput. Phys.*, 229(2):461 – 480, 2010.
- [22] K. S. Shterev. Iterative process acceleration of calculation of unsteady, viscous, compressible, and heat-conductive gas flows. *Internat. J. Numer. Methods Fluids*, 77(2):108–122, 2015.
- [23] P. Benner, S. Grundel, C. Himpe, C. Huck, T. Streubel, and C. Tischendorf. Gas network benchmark models. Technical Report 184, Humboldt-Universität zu Berlin (TRR154), 2017.
- [24] H. Egger, T. Kugler, and N. Strogies. Parameter identification in a semilinear hyperbolic system. *Inverse Probl.*, 33(5):055022, 2017.
- [25] 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(3):443–460, Jul 2017.

- [26] M. Herty. Modeling, simulation and optimization of gas networks with compressors. *Netw. Heterog. Media*, 2(1):81–97, 2007.
- [27] U. M. Ascher, S. J. Ruuth, and B. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM J. Numer. Anal.*, 32(3):797–823, 1995.
- [28] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, 2003.
- [29] 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(2):1035–1062, 2008.
- [30] 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.
- [31] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.
- [32] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers*. Oxford University Press, New York, June 2014.
- [33] T. Rees. *Preconditioning Iterative Methods for PDE-Constrained Optimization*. PhD thesis, University of Oxford, 2010.
- [34] Y. Qiu. *Preconditioning Optimal Flow Control Problems Using Multilevel Sequentially Semiseparable Matrix Computations*. PhD thesis, Delft Institute of Applied Mathematics, Delft University of Technology, 2015.
- [35] J. W. Pearson. On the development of parameter-robust preconditioners and commutator arguments for solving Stokes control problems. *Electron. Trans. Numer. Anal.*, 44:53–72, 2015.
- [36] M. Porcelli, V. Simoncini, and M. Tani. Preconditioning of active-set Newton methods for PDE-constrained optimal control problems. *SIAM J. Sci. Comput.*, 37(5):S472–S502, 2015.
- [37] M. Wathen, C. Greif, and D. Schötzau. Preconditioners for mixed finite element discretizations of incompressible MHD equations. *SIAM J. Sci. Comput.*, 39(6):A2993–A3013, 2017.
- [38] J. Pestana and A. J. Wathen. Natural preconditioning and iterative methods for saddle point systems. *SIAM Rev.*, 57(1):71–91, 2015.
- [39] M. Stoll and T. Breiten. A low-rank in time approach to PDE-constrained optimization. *SIAM J. Sci. Comput.*, 37(1):B1–B29, January 2015.
- [40] A. J. Wathen. Preconditioning. *Acta Numer.*, 24:329–376, 2015.
- [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(1):5:1–5:19, 2011.
- [42] C. Kelley. *Solving Nonlinear Equations with Newton’s Method*. Society for Industrial and Applied Mathematics, Philadelphia, 2003.
- [43] R. Dembo, S. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19(2):400–408, 1982.