

# Numerical Aspects of Model Order Reduction for Gas Transportation Networks

Sara Grundel, Nils Hornung, and Sarah Roggendorf

**Abstract** The chapter focuses on the numerical solution of parametrized unsteady Eulerian flow of compressible real gas in pipeline distribution networks. Such problems can lead to large systems of nonlinear equations that are computationally expensive to solve by themselves, more so if parameter studies are conducted and the system has to be solved repeatedly. The stiffness of the problem adds even more complexity to the solution of these systems. Therefore, we discuss the application of model order reduction methods in order to reduce the computational costs. In particular, we apply two-sided projection via proper orthogonal decomposition with the discrete empirical interpolation method to exemplary realistic gas networks of different size. Boundary conditions are represented as inflow and outflow elements, where either pressure or mass flux is given. On the other hand, neither thermal effects nor more involved network components such as valves or regulators are considered. The numerical condition of the reduced system and the accuracy of its solutions are compared to the full-size formulation for a variety of inflow and outflow transients and parameter realizations.

**Keywords** Gas network simulation • Model order reduction • Proper orthogonal decomposition • Discrete empirical interpolation method • Stiff initial-value problems

**MSC code:** 65K05 (Numerical Analysis, mathematical programming methods)

---

S. Grundel  
Max Planck Institute Magdeburg, Magdeburg, Germany  
e-mail: [sara.grundel@mpi-magdeburg.mpg.de](mailto:sara.grundel@mpi-magdeburg.mpg.de)

N. Hornung (✉) • S. Roggendorf  
Fraunhofer SCAI, Sankt Augustin, Germany  
e-mail: [nils.hornung@scai.fraunhofer.de](mailto:nils.hornung@scai.fraunhofer.de); [sarah.roggendorf@scai.fraunhofer.de](mailto:sarah.roggendorf@scai.fraunhofer.de)

# 1 Introduction

Gas as an energy source is being transported from producer or supplier to consumer along pipelines via short or long distances. Size and complexity of these transportation networks will thus vary notably. Maybe the most typical question with regard to such gas transportation problems is if the supply can satisfy consumer demands. Often this question is coupled with the goal to operate the network efficiently and to account for variations either in supply, demand or in the network properties itself. The latter problem might, e.g., result from the difficulty to measure pipeline properties such as roughness exactly. Since gas is compressible such that the network may hold strongly varying amounts, transient effects can become important.

All these questions and goals can in principle be answered by numerical simulation provided that the method of simulation is accurate enough while being efficient at the same time. Efficiency and computational costs depend on network size, nonlinearity, and stiffness of the underlying equations. In order to manage size while dealing with nonlinearity we demonstrate how a nonlinear model order reduction method can be applied, which is tailored to the problem in that the general form of the original equations is respected in the reduced order systems.

Certain important assumptions are made in order to simplify the problem, the most prominent of which is the neglect of all network elements except for simple pipelines, junctions, inflow and outflow elements. Moreover, we assume a single phase of an ideal gas flowing through the pipes, disregard most temperature effects, and make further assumptions that allow us to rewrite the problem as a system of ordinary differential equations. Some of these assumptions are made in favor of a simple exposition of the topic, while others originate from numerical considerations.

Transient simulation of gas networks is a very active field of research. Model order reduction for this system is treated on a more basic level by [3]. Many different approaches on how to efficiently compute transient behavior are known in the literature [6, 9, 10]. The main goal of this chapter is to introduce the reader to basic concepts of model order reduction involving its practical application. Thus we present three example networks of different complexity that form the foundation of a comprehensive treatment of a nonlinear reduced order method, including numerical tests with a focus on stiffness, accuracy, and computational cost.

How the laws of continuum mechanics can be applied to model gas flow through a network of pipes is explained in Section 2, including three different example problems to be used in later sections. Basic notions and definitions of stiffness are discussed in Section 3 with the application and example problems in mind. Section 4 introduces basic concepts of model order reduction with a focus on nonlinear methods suitable for gas transport networks, while Section 5 presents and discusses results for the example cases. The final section of this chapter summarizes the main topics and touches further interesting questions.

## 2 Problems in Network Simulation

We consider the simulation of gas flowing through a system of branching pipelines with influx and outflux defined at determined locations. First the network model itself is described, including continuum mechanics for the flow through a single pipe and mass conservation at pipeline junctions. Subsequently, we present example networks later used to empirically analyze the simulation of resulting full and reduced order systems.

### 2.1 Pipe Network Modeling

A gas transportation network can be described as a directed graph  $\mathcal{G} = (\mathcal{E}, \mathcal{N})$ , where  $\mathcal{N}$  is the set of nodes and  $\mathcal{E}$  is the set of directed edges. Those edges are denoted by tuples of nodes. We distinguish between so-called supply nodes  $\mathcal{N}_s$ , demand nodes  $\mathcal{N}_d$ , and interior nodes  $\mathcal{N}_0$ , where  $\mathcal{N} = \mathcal{N}_s \cup \mathcal{N}_0 \cup \mathcal{N}_d$ . If we choose not to consider more involved components, each edge represents a pipe and can thus be specified by length, width, and roughness. Nodes, on the other hand, can be imagined as the points where pipes start, end, or meet. The physics of a fluid moving through a single pipe can be modeled by the isothermal Euler equations averaged over the pipe's cross-section  $A = \frac{\pi}{4}D^2$  with inner pipe diameter  $D$ , see [11].

Now several simplifications are applied. Discarding terms related to kinetic energy, we first get

$$\partial_t \rho + \partial_x q = 0, \quad (1a)$$

$$\partial_t q + \partial_x p + g\rho \partial_x h = -\frac{\lambda(q)}{2D} \rho v |v|, \quad (1b)$$

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

Here  $\rho$  denotes the fluid's density,  $p$  the pressure, and  $q$  the volumetric flux. Together they form the set of unknown dynamical variables and depend on space and time  $(t, x)$ . Velocity is denoted by  $v = \frac{q}{\rho}$ . The system consists of three equations, two for mass and momentum conservation (1a,1b) and one that specifies material properties (1c). Momentum conservation (1b) includes a term on the left-hand side to incorporate gravity  $g$  as a conservative body force depending on height  $h$ , as well as a friction term on the right-hand side.

As an approximation to the friction coefficient  $\lambda(q)$ , we use

$$\lambda = \left( 2 \log \left( \frac{D}{\kappa} \right) + 1.138 \right)^{-2},$$

where  $D$  is again the diameter of the pipe and  $\kappa$  a parameter describing its roughness. Notice that this approximation, called the Swamee-Jain equation [12], neglects the dependence of friction on flux  $q$  if we assume, for simplicity again, that roughness  $\kappa$  does not depend on  $q$ . The field  $\gamma = RT$  embodies gas properties by its dependence on a given gas temperature  $T$  and on the universal gas constant  $R$ . For further simplification we assume that the temperature  $T \equiv T_0$  is constant in time and space and that we deal with an ideal gas (i.e.,  $z \equiv 1$ ). Thus,  $\gamma = RT_0$  is constant and we can rewrite the constitutive assumption as  $p = \gamma\rho$ .

Without change of notation, volumetric flux is now substituted by mass flux  $q \leftarrow Aq$ . Along with the relation  $v = \frac{q}{\rho}$  the simplified isothermal Euler equations take the form

$$\partial_t \rho + \frac{1}{A} \partial_x q = 0, \quad (2a)$$

$$\frac{1}{A} \partial_t q + \partial_x p + g\rho \partial_x h = -\frac{\lambda}{2DA^2} \frac{q|q|}{\rho}, \quad (2b)$$

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

or substituting  $\rho$  according to (2c),

$$\begin{aligned} \partial_t p &= -\frac{\gamma}{A} \partial_x q, \\ \partial_t q &= -A \partial_x p - \frac{Ag}{\gamma} p \partial_x h - \frac{\lambda\gamma}{2DA} \frac{q|q|}{p}. \end{aligned} \quad (3)$$

Within the pipe network, (3) is valid for each edge. Any edge stands for a pipe parametrized along its given length  $L$ , which means the interval  $[0, L]$  establishes the domain of definition of the according partial differential equation. The full system of equations additionally encompasses consistency conditions for each demand node as well as input in terms of supply pressures given at supply nodes.<sup>1</sup> If we assemble all pipes that end in node  $i$  into the set  $I_R^i$  and, accordingly, all pipes that start in node  $i$  into the set  $I_L^i$ , then the demand consistency conditions are given by

$$0 = \sum_{l \in I_L^i} q_l(L_l, t) - \sum_{k \in I_R^i} q_k(0, t) + d_i(t) \quad (4)$$

for every node  $i$ .<sup>2</sup> If

---

<sup>1</sup>We always identify flux conditions with demand, i.e., with outflux, and pressure boundary conditions with supply, i.e., with influx. Without further modification, this (somehow arbitrary) identification can be relaxed such that demand can also be modeled as pressure conditions and supply as mass or volumetric fluxes.

<sup>2</sup>The direction given to the edges serves the sole purpose of topology definition and is independent of the direction of the flux within the pipe that results from the laws of continuum mechanics.

$$d_i(t) \equiv 0, \quad (5)$$

this condition reduces to the well-known Kirchhoff law and is valid at any junction of pipes. Strictly speaking, (4) holds for all nodes in  $\mathcal{N}_0 \cup \mathcal{N}_d$ , whereas for nodes in  $\mathcal{N}_0$  we additionally have (5).

Let  $n_E$  be the number of edges and assume they are ordered such that every edge has an index in  $[1, 2, \dots, n_E]$ . Once we discretize (3) following [4] and add given supply pressures, the resulting overall differential algebraic system of equations is given by

$$\partial_t \frac{p_R^k + p_L^k}{2} = - \frac{\gamma}{A_k} \frac{q_R^k - q_L^k}{L_k} \quad \forall k \in [1, \dots, n_E], \quad (6a)$$

$$\begin{aligned} \partial_t \frac{q_R^k + q_L^k}{2} = & -A_k \frac{p_R^k - p_L^k}{L_k} - \frac{A_k g}{2\gamma} (p_R^k + p_L^k) \frac{h_R^k - h_L^k}{L_k} \\ & - \frac{\lambda_k \gamma}{4D_k A_k} \frac{(q_R^k + q_L^k) |q_R^k + q_L^k|}{p_R^k + p_L^k} \quad \forall k \in [1, \dots, n_E], \end{aligned} \quad (6b)$$

$$0 = \sum_{l \in I_L^i} q_l^l - \sum_{k \in I_R^i} q_k^k - d_i(t) \quad \forall i \in \mathcal{N}_0 \cup \mathcal{N}_d, \quad (6c)$$

$$0 = p_i(t) - s_i(t) \quad \forall i \in \mathcal{N}_s. \quad (6d)$$

The vector  $q_R$  is the vector of fluxes at the end of the pipes, and the vector  $q_L$  is the vector of fluxes at the beginning of the pipes. Except for those at the beginning and end, we do not take any values along the pipes. This means, if the numerical error by this discretization exceeds our needs because the pipes are too long, we have to add artificial nodes (junctions) to the network such that all pipes are short enough to yield accurate enough results.

For a more compact description, we write the system in matrix notation

$$|B_S^T| \partial_t p_s + |B_0^T| \partial_t p_d = -M_L^{-1} q_-, \quad (7a)$$

$$\partial q_+ = M_A (B_S^T p_s + B_0^T p_d) + g(q_+, p_s, p_d), \quad (7b)$$

$$0 = B_0 q_+ + |B_0| q_- - d(t), \quad (7c)$$

$$0 = p_s - s(t), \quad (7d)$$

where

$$M_L = \text{diag}(\dots, \frac{L_k A_k}{4\gamma}, \dots), \quad (8)$$

$$M_A = \text{diag}(\dots, -\frac{A_k}{L_k}, \dots), \quad (9)$$

and the  $k$ -th component of the function  $g$  is given by

$$g_k(q_+, \rho_d, \rho_s) = -\frac{A_k g}{2\gamma} \ell_k(p_d, p_s) \frac{h_R^k - h_L^k}{L_k} - \frac{\lambda_k \gamma}{4D_k A_k} \frac{(q_+^k) |q_+^k|}{\ell_k(\rho_d, \rho_s)}, \quad (10)$$

where  $\ell_k$  is the  $k$ -th entry of the vector-valued function  $\ell$

$$\ell(p_d, p_s) = |B_0^T| p_d + |B_S^T| p_s.$$

Both matrices  $M_L$  and  $M_A$  are diagonal and invertible. The matrix

$$B = \begin{bmatrix} B_0 \\ B_S \end{bmatrix}$$

denotes the incidence matrix of the underlying directed graph, where  $B_0$  corresponds to the demand nodes and junctions and  $B_S$  corresponds to the supply nodes. In addition, the notation  $q_- = q_R - q_L$  and  $q_+ = q_R + q_L$  has been introduced here. To eliminate  $q_-$ , we multiply (7a) by  $|B_0| M_L$  and then use (7c) to substitute  $|B_0| q_-$ ,

$$\begin{aligned} & |B_S^T| \partial_t p_s + |B_0^T| \partial_t p_d = -M_L^{-1} q_- \\ \Rightarrow & |B_0| M_L |B_S^T| \partial_t p_s + |B_0| M_L |B_0^T| \partial_t p_d = -|B_0| q_- \\ \Rightarrow & |B_0| M_L |B_S^T| \partial_t p_s + |B_0| M_L |B_0^T| \partial_t p_d = B_0 q_+ - d(t). \end{aligned}$$

We also replace  $p_s$  according to (7d) to obtain

$$|B_0| M_L |B_0^T| \partial_t p_d = B_0 q_+ - d(t) - |B_0| M_L |B_S^T| \partial_t s(t), \quad (11a)$$

$$\partial_t q_+ = M_a B_0^T p_d + g(q_+, s(t), p_d) + M_a B_S^T s(t) \quad (11b)$$

with  $g$  as in (10). The structure that (11) implies can be seen more clearly in block matrix notation

$$\begin{bmatrix} |B_0| M_L |B_0^T| & 0 \\ 0 & M_a^{-1} \end{bmatrix} \begin{bmatrix} \partial_t p_d \\ \partial_t q_+ \end{bmatrix} = \begin{bmatrix} 0 & B_0 \\ B_0^T & 0 \end{bmatrix} \begin{bmatrix} p_d \\ q_+ \end{bmatrix} + \begin{bmatrix} -d(t) - |B_0| M_L |B_S^T| \partial_t s \\ M_a^{-1} g + B_S^T s \end{bmatrix}. \quad (12)$$

Compare this derivation of an ordinary differential equation to the very similar approach of [4], where more details are conveyed. From now on we consider (12) of the size  $|\mathcal{N}| - |\mathcal{N}_s| + n_E$  (the number of nodes minus the number of supply nodes plus the number of edges). It is an ordinary differential equation in descriptor form where the matrix  $E$ ,

$$E = \begin{bmatrix} |B_0| M_L |B_0^T| & 0 \\ 0 & M_a^{-1} \end{bmatrix},$$

is positive definite and symmetric. Furthermore, the equation depends on several parameters given by pipe lengths  $L$ , diameters  $D$ , friction coefficients  $\lambda$ , height differences  $\Delta h$ , and gas properties  $\gamma$  totaling  $4 \times n_E + 1$ -many parameters.

*Remark 1.* We are now dealing with an ordinary differential equation of the form

$$E\dot{x} = Tx + f(x, u) + Ku.$$

This means we have been able to decouple the system in such a way that it is no longer written in the form of a differential algebraic equation. During the process of reformulation, the time derivative of the input signal  $s$ , which is the pressure at the supply nodes, has been introduced. Since this pressure is usually given explicitly as a slowly changing function of time we can calculate its derivative in most practical applications.

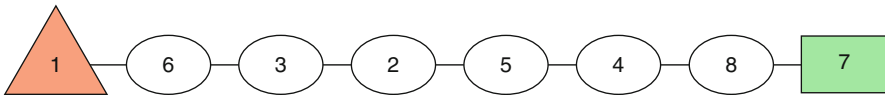
## 2.2 Example Problems

In this chapter we are going to numerically analyze three example networks. The first consists in a single pipe, the second in a small connected network of 57 pipes, and the third in a larger network with a higher number of elements forming several connected components, the largest of which includes 669 pipes. Network topology is given by a graph which is mainly defined by a list of edges, i.e., ordered tuples of nodes indicating the direction of the edge. For simplicity, the total set of  $N_N$  nodes is just described by their numbering, i.e., by a set of integers of the form  $\{n \in \mathbb{N} : n \leq N_N\}$ . The set of edges given as a list automatically implies a numbering of the edges which we later make use of to set up our equations. Length, height difference, width, and roughness are given as a parameter vector for each pipe. Furthermore, one or several demand fluxes are provided by functions of time. Similarly, one or several time-dependent supply pressures are given.

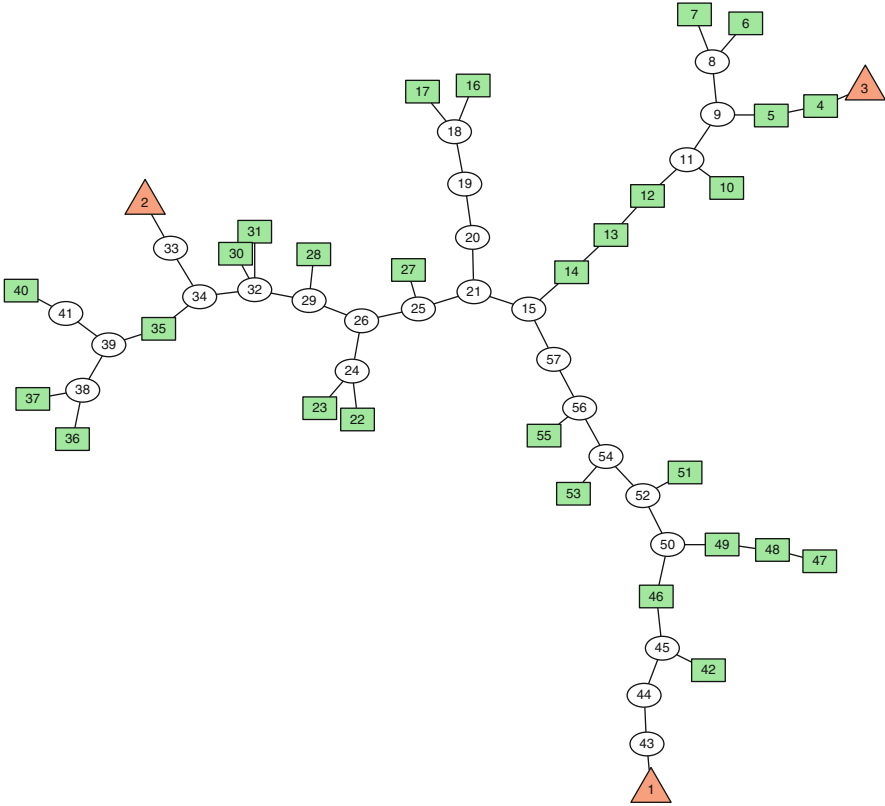
The topologies of the three networks are visualized in Figures 1, 2 and 3. Supply nodes are marked by triangles, and nodes with nonzero demand are marked by rectangles. Tables 1 and 4 show the corresponding parameter vectors.

At the supply nodes a supply function of the following form is given

$$s(t) = \alpha_s \times s(0) \times \left( 0.5 \times \left( \cos \left( \frac{\pi \times (t - T_1^s)}{T_2^s - T_1^s} \right) - 1 \right) \right) + s(0), \quad (13)$$



**Fig. 1** Network 1, a single pipe divided into subsections of different properties



**Fig. 2** Network 2, a small network with several supply and demand notes

where  $\alpha_s$  is the portion the supply pressure drops during the given time period  $T_2^s - T_1^s$ . Tables 2 and 3 show the supply pressures at time zero for all supply nodes. Similarly, the demand functions at nodes with nonzero demand are of the form

$$d(t) = d(0) + 0.5 \times \alpha_d \times \left( 1 - \cos \left( \frac{\pi \times (t - T_1^d)}{T_2^d - T_1^d} \right) \right) \times d(0), \quad (14)$$

where  $\alpha_d$  denotes the percentages the demand grows. The corresponding values of  $d(0)$ ,  $T_1^d$ , and  $T_2^d$  are listed in Tables 2 and 5 for all nodes with nonzero demand. This means that the first example, the single pipe, yields a system of ordinary differential equations of size 14 with 22 parameters, the second network is of size 110 with 29 parameters, and the last network's largest subnet is of size 1341.



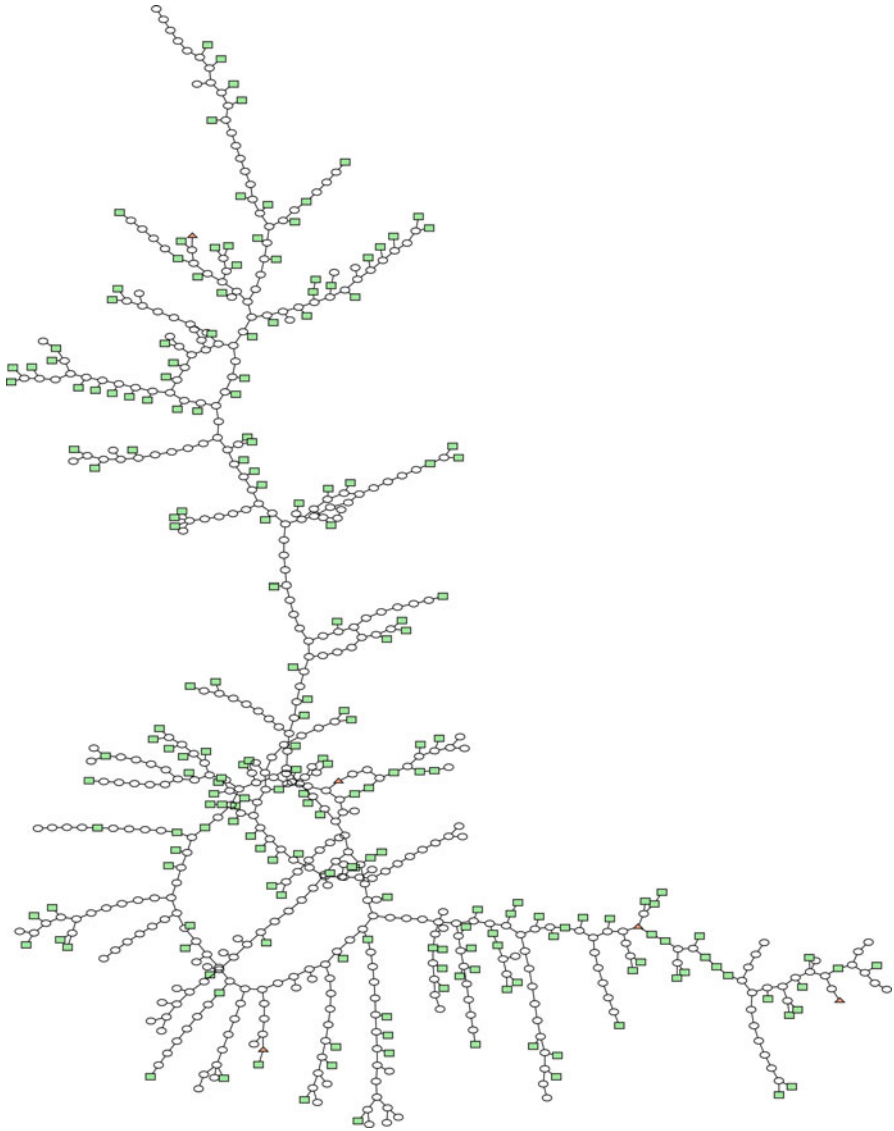


Fig. 3 Connected component of network 3, of larger scale

### 3 Stiffness in Ordinary Differential Equations

A simple idea in principle, stiffness can be conceived as the intuition that stability is more critical than accuracy or, much simpler, that explicit integration methods might fail. To put this into a mathematical framework can become rather difficult,

**Table 1** Parameters for network 1

Pipe element	Length [m]	Diameter [m]	Friction coefficient	Height difference [m]
(8,7)	100.000	1.000	0.012	0.000
(1,6)	100.000	0.900	0.008	0.000
(3,2)	1,000.000	0.500	0.014	0.000
(6,3)	1,000.000	0.500	0.014	0.000
(5,2)	1,000.000	0.500	0.014	0.000
(5,4)	1,000.000	0.500	0.014	0.000
(8,4)	1,000.000	0.500	0.014	0.000

**Table 2** Supply and demand for network 1

Node	1	Node	7
$s(0)$ [Pa]	5,000,000	$d(0)$ [kg/s]	219.230
$T_1^s$ [s]	500,000	$T_1^d$ [s]	100,000
$T_2^s$ [s]	1,000,000	$T_2^d$ [s]	500,000

**Table 3** Supply for network 2

Node	1	2	3
$s(0)$ [Pa]	5,400,000	2,700,000	2,700,000
$T_1^s$ [s]	700,000	700,000	700,000
$T_2^s$ [s]	1,000,000	1,000,000	1,000,000

though. As a means to analyze stability, we introduce concepts and definitions of stiffness in linear constant-coefficient ordinary differential equations in the following subsection as well as a discussion of practical implications for three example cases in subsequent subsections.

### 3.1 Concepts and Definitions

If a linear constant-coefficient ordinary differential equation is given as

$$\dot{x} = Ax,$$

stiffness is often quantified via a ratio of the minimum and maximum real part of the eigenvalues under the assumption that all eigenvalues of  $A$  be negative. Such a ratio is linked with the behavior of the differential equation for  $t \rightarrow \infty$  (and can be too liberal a condition). The Lipschitz constant of the linear function, which coincides with the largest singular value of  $A$ , gives another (too conservative) criterion of stiffness that is tied to the limit  $t \rightarrow t_0$ .

A more realistic measure of stiffness for linear systems is suggested by [7] in terms of a pseudo spectral analysis. We do not repeat details here. It is, however, important to realize that the two notions of stiffness differ more strongly from each

**Table 4** Parameters for network 2

Pipe	$L$ [m]	$D$ [m]	$\lambda$	$(h_R - h_L)$ [m]
(1,43)	1.0	1.0	0.0120	0.0
(57,56)	1, 661.0	0.8	0.0125	-22.0
(56,55)	1, 550.0	0.8	0.0125	6.0
(56,54)	1, 530.0	0.8	0.0125	6.0
(54,53)	750.0	0.8	0.0125	-3.0
(54,52)	4, 089.0	0.8	0.0125	0.0
(52,51)	229.0	0.8	0.0125	5.0
(52,50)	1, 135.0	0.8	0.0125	3.0
(50,49)	222.0	0.8	0.0125	1.0
(50,46)	1, 948.0	1.0	0.0120	67.0
(49,48)	732.0	0.8	0.0125	-3.0
(48,47)	10.0	0.8	0.0125	0.0
(46,45)	15, 134.0	1.0	0.0120	78.0
(45,44)	1.0	1.0	0.0120	0.0
(45,42)	14, 088.0	1.0	0.0120	-81.0
(44,43)	3.0	1.0	0.0120	0.0
(41,40)	2, 258.0	0.9	0.0122	4.0
(41,39)	2, 010.0	0.9	0.0122	4.0
(39,38)	1, 948.0	0.9	0.0122	2.0
(39,35)	3, 533.0	0.9	0.0122	2.0
(38,37)	35.0	0.9	0.0122	2.0
(38,36)	111.0	0.9	0.0122	-3.0
(35,34)	1, 930.0	0.9	0.0122	5.0
(34,33)	81.0	0.9	0.0122	1.0
(34,32)	1640.0	0.9	0.0122	8.0
(33,2)	34.0	0.9	0.0122	-1.0
(32,31)	1, 666.0	0.9	0.0122	19.0
(32,30)	763.0	0.9	0.0122	16.0

Pipe	$L$ [m]	$D$ [m]	$\lambda$	$(h_R - h_L)$ [m]
(32,29)	7, 700.0	0.9	0.0122	22.0
(29,28)	1, 350.0	0.9	0.0122	-12.0
(29,26)	6, 300.0	0.9	0.0122	-7.0
(26,25)	343.0	0.6	0.0132	-10.0
(26,24)	1, 455.0	0.6	0.0132	-8.0
(25,27)	189.0	0.6	0.0132	-1.0
(25,21)	2, 486.0	0.6	0.0132	64.0
(24,23)	392.0	0.6	0.0132	-1.0
(24,22)	814.0	0.6	0.0132	3.0
(21,20)	18.0	0.6	0.0132	1.0
(19,18)	621.0	0.6	0.0132	-24.0
(18,17)	1, 818.0	0.6	0.0132	-47.0
(18,16)	16.0	0.6	0.0132	0.0
(15,14)	1, 040.0	0.6	0.0188	34.0
(14,13)	8, 296.0	0.6	0.0188	-29.0
(13,12)	1, 295.0	0.6	0.0188	2.0
(12,11)	521.0	0.6	0.0188	0.0
(11,10)	470.0	0.7	0.0128	14.0
(11,9)	1, 507.0	0.7	0.0181	-3.0
(9,8)	789.0	0.7	0.0128	4.0
(9,5)	800.0	0.7	0.0161	1.0
(8,7)	275.0	0.7	0.0128	9.0
(8,6)	1, 305.0	0.7	0.0128	2.0
(5,4)	11, 866.0	0.7	0.0161	-7.0
(4,3)	3, 212.0	0.7	0.0161	-6.0
(20,19)	1, 000.0	0.5	0.0137	0.0
(57,15)	1, 000.0	0.8	0.0125	0.0

other the further away  $A$  is from a normal matrix, where nonnormality can, e.g., be measured by

$$\|A^*A - AA^*\|_F.$$

Here  $\|\cdot\|_F$  denotes the Frobenius norm. As a consequence, both concepts of stiffness are appropriate for normal matrices. The matrices arising from the systems presented in the following are not normal, though, such that we need to expect different stiffness measures.

The whole issue of stiffness becomes even more complicated once we deal with nonlinear equations and source terms, which are present in the systems that we consider. Linear stiffness theory for constant-coefficient ordinary differential

**Table 5** Demand for network 2

Node	4	5	6	7	10	12	13	14	16	17
$d(0)$ [kg/s]	0.092	0.028	0.538	0.434	0.807	0.035	0.045	0.452	0.471	1.773
$T_1$ [s]	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000
$T_2$ [s]	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
Node	22	23	27	28	30	31	35	36	37	40
$d(0)$ [kg/s]	0.599	0.133	0.040	0.018	0.478	0.678	0.573	0.093	0.710	0.598
$T_1$ [s]	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000
$T_2$ [s]	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000
Node	42	46	47	48	49	51	53	55		
$d(0)$ [kg/s]	0.318	0.561	0.498	0.493	0.233	0.172	0.278	0.359		
$T_1$ [s]	100,000	100,000	100,000	100,000	100,000	100,000	100,000	100,000		
$T_2$ [s]	500,000	500,000	500,000	500,000	500,000	500,000	500,000	500,000		

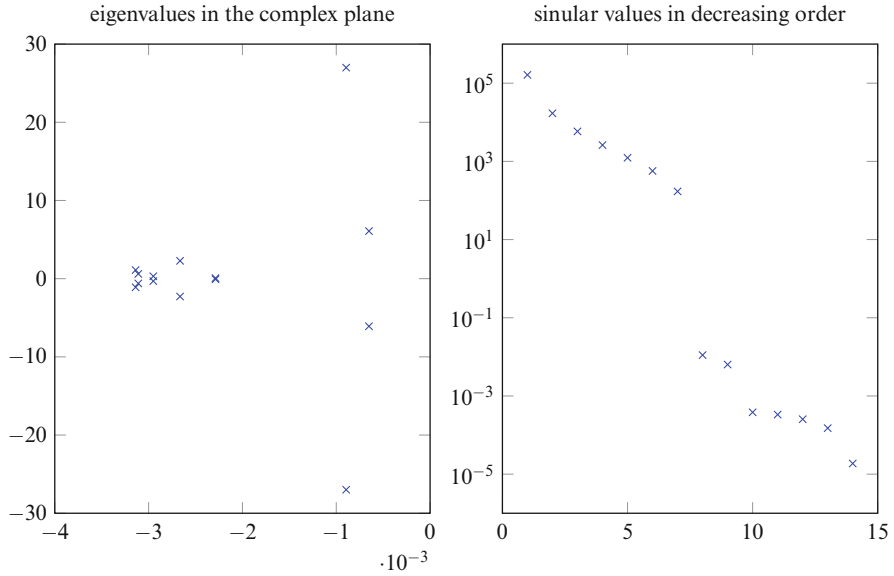
equations can be applied to a linearization of the nonlinear system: Via the Jacobian matrix we get a constant-coefficient linear equation that can be evaluated at a given time  $t_0$ . However, information may get lost in the process of linearization. In the following we empirically evaluate and show the stiffness of the ordinary differential equation (12) for the single pipe and the small network example from Section 2.2.

### 3.2 Single Pipe System

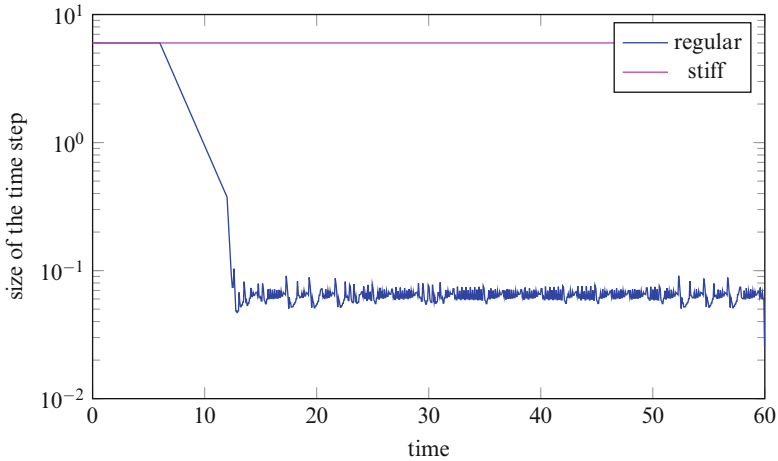
The single pipe introduced in Section 2.2 comprises segments of varying properties and can hence be written as a system of ordinary differential equations of size 14.

In this first excursion in stiffness we are going to compute the common notions of stiffness that are given by the eigenvalues and the singular values of  $E^{-1}J_f(0, x_0)$ , where  $x_0$  denotes a stationary solution at time  $t = 0$ . This stationary solution can in principle be chosen arbitrarily and will serve as a starting value later. The Jacobian's singular values and eigenvalues are plotted in Figure 4. Notice that these plots tend to look very similar for other values of  $t$  and  $x$ . The analysis of singular values and eigenvalues leads to a liberal stiffness estimate of 4 and a conservative estimate of  $1.6 \times 10^5$ . Because the difference between both estimates comprises several orders of magnitude we cannot conclude the extent of stiffness, hence the problem at hand might either become rather stiff or only mildly so.

By a simple numerical test we notice, though, that the step size of a regular solver for non-stiff problems (here MATLAB's `ode15s`) is chosen much smaller than by the corresponding stiff solver (here MATLAB's `ode23s`). This difference can be interpreted as a sign of more than only mild stiffness. The step sizes chosen by the different MATLAB solvers are shown in Figure 5, where we can

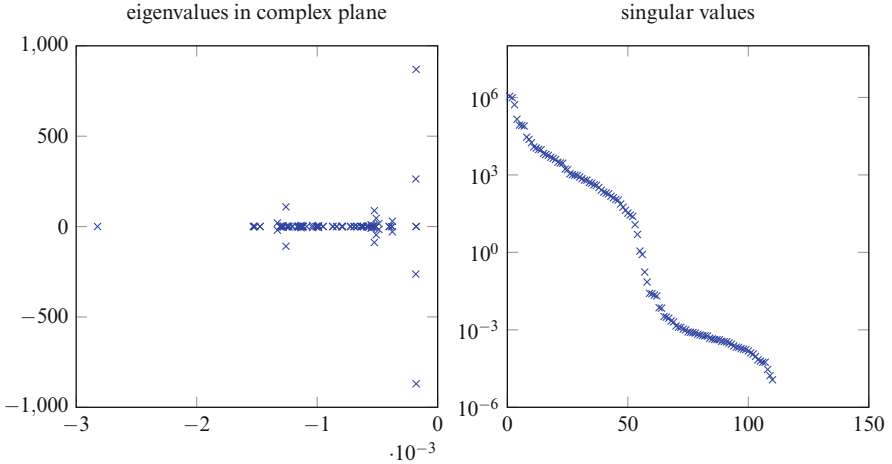


**Fig. 4** The eigenvalues and the singular values of the linearized system, network 1



**Fig. 5** The time steps chosen by MATLAB’s ode23 and ode15s solvers for ordinary differential equations, network 1

observe a difference of two orders of magnitude for this simple example. Even more important, we must expect that we deal with serious stiffness in this class of gas transportation problems due to the fact that, in spite of those small step sizes, the solution found by MATLAB’s non-stiff ode15 blows up in finite time whereas the solution given by ode23s does not.



**Fig. 6** The eigenvalues and singular values of the linearized system, network 2

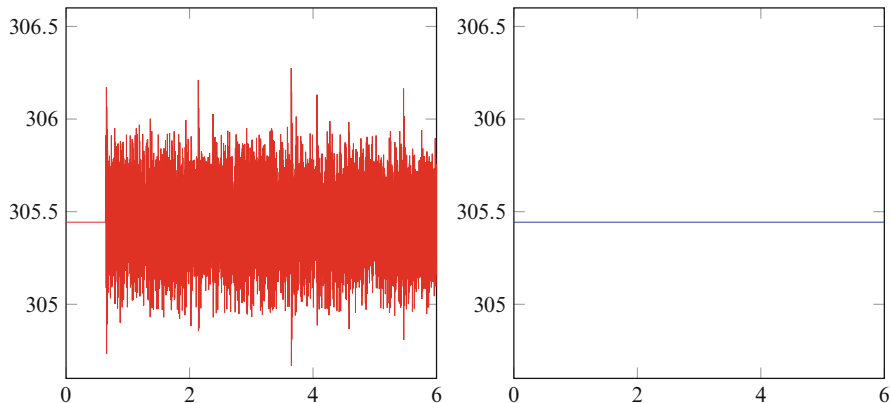
### 3.3 Small Networks

For a network of pipes the stiffness of the system becomes clearer yet. Looking at the eigenvalue and singular value plot of the Jacobian matrix, which is shown in Figure 6, we again see that the stiffness estimates vary strongly by several orders of magnitude, this time between 16 and  $1.1 \times 10^6$ . As above this result eludes an exact stiffness analysis of the system of equations. Since both estimates grow higher than in the previous example, stiffness can most probably be assumed to increase with the complexity and size of the network. Conducting the same experiment as before, we compare the resulting MATLAB step sizes as in Section 3.2. We do not include a plot as it shows the same peculiarities as Figure 5, only that the time step of the non-stiff method decreases even more, oscillates around  $10^{-3}$  s and therefore differs by four orders of magnitude from the stiff solver.

Regarding the first six seconds, the solutions computed via the two different solvers are depicted in Figure 7. We can notice again that these systems seem to require a stiff or, possibly even better, a dedicated solver.

## 4 Model Order Reduction

Seeing that these network problems can become arbitrarily large and stiff, we would be interested in creating a reduced order model which is of small order and hopefully not stiffer than the full system. Since the system is nonlinear, we apply the model order reduction method called proper orthogonal decomposition as this is typically the method of choice for nonlinear systems.



**Fig. 7** The time steps chosen by MATLAB's `ode23` and `ode15s` solvers for ordinary differential equations, network 2

#### 4.1 Proper Orthogonal Decomposition

The model order reduction method called proper orthogonal decomposition (POD) starts with multiple snapshots  $\{y_j^k\}_{j=1}^m \subset X$ ,  $1 \leq k \leq p$ , for a given Hilbert space  $X$  (typically  $\mathbb{R}^n$  or a function space like  $\mathcal{L}^2$ ). One is interested in finding a subspace spanned by an orthonormal basis  $\psi_1, \dots, \psi_\ell$  within the Hilbert space  $X$  that solves the minimization problem

$$\min \sum_{k=1}^p \sum_{j=1}^m \alpha_j \left\| y_j^k - \sum_{i=1}^{\ell} \langle y_j^k, \psi_i \rangle_X \psi_i \right\|_X^2 \quad (15)$$

s. t.  $\{\psi_i\}_{i=1}^{\ell} \subset X$  and  $\langle \psi_i, \psi_j \rangle_X = \delta_{ij}$ ,  $1 \leq i, j \leq \ell$ .

This minimization can also be written as a maximization problem due to the orthogonal nature of the elements  $\psi_i$

$$\max \sum_{k=1}^p \sum_{j=1}^m \alpha_j \sum_{i=1}^{\ell} \langle y_j^k, \psi_i \rangle_X^2 \quad (16)$$

s. t.  $\{\psi_i\}_{i=1}^{\ell} \subset X$  and  $\langle \psi_i, \psi_j \rangle_X = \delta_{ij}$ ,  $1 \leq i, j \leq \ell$ .

**Theorem 1.** Let  $X$  be a separable Hilbert space,  $\mathcal{R} : X \rightarrow X$  a summation operator on  $X$  defined by

$$\mathcal{R} : X \rightarrow X, \quad \mathcal{R}\psi \mapsto \sum_{k=1}^p \sum_{j=1}^m \alpha_j \langle \psi, y_j^k \rangle_X y_j^k.$$

The following assertions are all true:

- (a)  $\mathcal{R}$  is linear compact self-adjoint and nonnegative.
- (b) A basis of  $X$  of eigenfunctions  $\psi_j$  exists such that

$$\mathcal{R}\psi_j = \lambda_j\psi_j$$

and  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d > \lambda_{d+1} = \dots = 0$ .

- (c) The first  $\ell$  of these eigenfunctions solve the minimization problem (15) and the maximization problem (16).

For a proof of Theorem 1 see [13].

In the remainder of this chapter, let us assume that  $X = \mathbb{R}^n$  and the inner product is defined by  $\langle x, y \rangle_X = x^T E y$  for a given positive definite symmetric matrix  $E$ . We will denote by  $Y_k$  the matrix of snapshots  $\{y_j^k\}_{j=1}^m$ . This means  $Y$  is an  $n \times m$  matrix.

**Corollary 1.** *The eigenvectors of the operator  $\mathcal{R}$  equate to the eigenvectors of the matrix*

$$(Y_1 D Y_1^T + \dots + Y_p D Y_p^T) E, \quad (17)$$

where  $D = \text{diag}(\alpha_1, \dots, \alpha_n)$ .

So in order to find the best approximation space of size  $\ell$  for a given set of snapshots, all we have to do is compute the first  $\ell$  eigenvectors of (17). Notice that the matrix (17) is an  $n \times n$  matrix, the size of the state space  $X$ . This matrix is not symmetric. If we, however, knew the eigenvalues and eigenvectors of the following symmetric system

$$E^{1/2} (Y_1 D Y_1^T + \dots + Y_p D Y_p^T) E^{1/2}, \quad (18)$$

we could compute the eigenvectors of (17). In fact, these eigenvectors are obtained if we multiply the eigenvectors of (18) by  $E^{-1/2}$ . Since  $E$  is usually a large matrix and its square root is hard to compute, we are going to describe a faster alternative. The matrix (18) can be written as  $\hat{Y}\hat{Y}^T$  for

$$\hat{Y} = [\hat{Y}_1 \ \hat{Y}_2 \ \dots \ \hat{Y}_p],$$

where  $\hat{Y}_i = E^{1/2} Y_i D^{1/2}$ . Since this matrix is symmetric and positive definite its eigenvalue and singular value decompositions are equivalent. Assuming we know the singular value decomposition of  $\hat{Y}$ ,

$$\hat{Y} = U S V^T, \quad (19)$$

we also know the singular value decomposition of  $\hat{Y}\hat{Y}^T = U S^2 U^T$  and the singular value decomposition of  $\hat{Y}^T \hat{Y} = V S^2 V^T$ . This means, if we compute  $V$  we can recover



$U$  by (19), with  $U = \hat{Y}S^{-1}V$ . The matrix  $U$  is the matrix of singular vectors or eigenvectors of  $\hat{Y}\hat{Y}^T$  (recall that  $\hat{Y}\hat{Y}^T$  equals (18)). Hence, we get the eigenvector matrix of (17) by  $E^{-1/2}U = E^{-1/2}\hat{Y}S^{-1}V$ . If  $\phi_1, \dots, \phi_\ell$  are the singular vectors of  $\hat{Y}\hat{Y}^T$  we can, consequently, compute the eigenvectors  $\psi_i$ ,

$$\psi_i = \frac{1}{\sqrt{\sigma_i}} [Y_1 D^{1/2}, Y_2 D^{1/2}, \dots, Y_p D^{1/2}] \phi_i,$$

where  $\sigma_i$  and  $\phi_i$  are the singular values and singular vectors of  $\hat{Y}^T\hat{Y}$ . The objective function is then given by

$$\begin{aligned} & \sum_{k=1}^p \sum_{j=1}^m \alpha_j \left\| y_j^k - \sum_{i=1}^{\ell} \langle y_j^k, \psi_i \rangle_X \psi_i \right\|_X^2 \\ &= \sum_{k=1}^p \sum_{j=1}^m \alpha_j \left( \|y_j^k\|_X^2 - 2 \langle y_j^k, \sum_{i=1}^{\ell} \langle y_j^k, \psi_i \rangle_X \psi_i \rangle + \left\| \sum_{i=1}^{\ell} \langle y_j^k, \psi_i \rangle_X \psi_i \right\|_X^2 \right) \\ &= \sum_{k=1}^p \sum_{j=1}^m \alpha_j \left( \|y_j^k\|_X^2 - \sum_{i=1}^{\ell} \langle y_j^k, \psi_i \rangle_X^2 \right) \\ &= \sum_{i=1}^n \sigma_i - \sum_{i=1}^{\ell} \sigma_i = \sum_{i>\ell} \sigma_i. \end{aligned}$$

which follows from the fact that  $\psi_i$  are eigenfunctions of the operator  $\mathcal{R}$  [13]. Not to mention, this result can be utilized to determine  $\ell$ , i.e., to decide where to cut off eigenvalue computations. In practical applications, a heuristic choice for  $\ell$  can, e.g., be implied by the requirement that  $\mathcal{E} \geq 99\%$  where

$$\mathcal{E} = \frac{\sum_{i=1}^{\ell} \sigma_i}{\sum_{i=1}^n \sigma_i} = \frac{\sum_{i=1}^{\ell} \sigma_i}{\sum_{k=1}^p \sum_{j=1}^m \alpha_j \|y_j^k\|_X^2}. \quad (20)$$

We now need to understand how to use the described subspace of  $X$  in order to create a reduced order model of a dynamical system given in the form

$$E\dot{x} = Tx + f(x, u) + Ku,$$

where the function  $f$  depends on the dynamic variable  $x$  and the time-dependent input function  $u$ . We are now going to briefly address the matter that the system matrices  $E, T, K$ , and the function  $f$  additionally depend on parameters. The basic idea is to generate snapshots for certain sampled parameter values. This parameter sampling can be picked as a uniform grid. An alternative way to set up a sampling in a semi-optimized way by using a Greedy algorithm is, e.g., described by [5].

---

**Algorithm 1** POD
 

---

**Require:**  $Y_1, \dots, Y_p \in \mathbb{R}^{N \times n}$ , inner product matrix  $E$ , diagonal weight matrix  $D$ , reduction order  $\ell$

**Ensure:** metamodel  $\psi_1, \dots, \psi_\ell$

- 1:  $B_{jk} \leftarrow D^{1/2} Y_j^T E Y_k D^{1/2}, j, k \in \{1, \dots, p\}$
  - 2:  $B \leftarrow$  matrix with  $B_{ij}$  blocks
  - 3:  $\sigma_i, \phi_i \leftarrow$  singular values and singular vectors of  $B$
  - 4: **for**  $i \leftarrow 1 : \ell$  **do**
  - 5:      $\psi_i \leftarrow \frac{1}{\sqrt{\sigma_i}} [Y_1 D^{1/2}, \dots, Y_p D^{1/2}] \phi_i$
- 

In fact, (15) can be seen as the discrete version of a continuous minimization problem. Seen in this light, the objective function would read

$$\sum_{k=1}^p \int_0^T \left\| y^k(t) - \sum_{i=1}^{\ell} \langle y^k(t), \psi_i \rangle_X \psi_i \right\|_X^2,$$

which, in turn, *implies* the minimization problem (15) by using a quadrature rule to compute the integral. A classical choice of  $\alpha_i$  are trapezoidal weights and the inner product is typically given by  $\langle x, y \rangle = x^T E y$  with the inner product matrix denoted by  $E$ . This allows us to compute  $\psi_1, \dots, \psi_\ell$  by the methods discussed above and summarized in Algorithm 1. We then determine the matrix  $W$  as  $W = [\psi_1, \dots, \psi_\ell]$  and project the large scale system by Galerkin projection onto

$$W^T E W \dot{\hat{x}} = W^T T W x + W^T f(W \hat{x}, u) + W^T K u, \quad (21)$$

where  $W^T E W = I$  by the mutual orthogonality of  $\psi_i$  with respect to said inner product.

## 4.2 Nonlinearity and Problem Structure

In the following, we are going to describe the rest of the algorithm that is used in the numerical experiments of Section 5. There are mainly two questions left to be answered carefully which were not mentioned in the theoretical derivation of POD.

1. Do we need to consider the special structure of the problem when we set up the reduced order model? Pressures and fluxes are not necessarily of a similar magnitude such that a direct application of the POD matrix as in (21) might neglect important effects [8]. We therefore reduce the pressure and the flux vector separately.
2. How do we handle the nonlinearity of the function? The term  $W^T f(W \hat{x})$  contributes  $\ell$ -dimensional function values with an  $\ell$ -dimensional vector-valued

argument. However, a higher-dimensional vector must be evaluated to compute  $W^T f$ . To obtain a fast simulation, this evaluation needs to be avoided.

Let us first discuss implications of the special structure of our problem. For the convenience of the reader, the ordinary differential equation (12) is repeated here:

$$\begin{bmatrix} |B_0| M_L |B_0^T| & 0 \\ 0 & M_a^{-1} \end{bmatrix} \begin{bmatrix} \partial_t p_d \\ \partial_t q_+ \end{bmatrix} = \begin{bmatrix} 0 & B_0 \\ B_0^T & 0 \end{bmatrix} \begin{bmatrix} p_d \\ q_+ \end{bmatrix} + \begin{bmatrix} -d(t) - |B_0| M_L |B_S^T| \partial_t s \\ g(q, p, s) + B_S^T s \end{bmatrix}.$$

For given input functions  $s(t)$ ,  $d(t)$ , and given parameters, we solve this differential equation within the time interval  $[0, T]$  in time steps of size  $\Delta t > 0$ . This means we obtain pressure values  $p(0), p(\Delta t), \dots, p(T)$  and flux values  $q(0), q(\Delta t), \dots, q(T)$ , representing snapshots that we assemble into matrices

$$Y_i^p \text{ (pressures) and } Y_i^q \text{ (fluxes),} \quad (22)$$

where  $i$  indicates the snapshots of different given parameter values. They are used to calculate projection matrices  $W_p$  and  $W_q$  separately for pressures and fluxes by Algorithm 1, such that the reduced order system obtained via Galerkin projection of the full order ordinary differential equation onto the matrix

$$\begin{bmatrix} W_p & 0 \\ 0 & W_q \end{bmatrix}$$

reads

$$\begin{bmatrix} W_p^T |B_0| M_L |B_0^T| W_p & 0 \\ 0 & W_q^T M_a^{-1} W_q \end{bmatrix} \begin{bmatrix} \partial_t p_d \\ \partial_t q_+ \end{bmatrix} = \begin{bmatrix} 0 & W_p^T B_0 W_q \\ W_q^T B_0^T W_p & 0 \end{bmatrix} \begin{bmatrix} p_d \\ q_+ \end{bmatrix} + \begin{bmatrix} -W_p^T d(t) - W_p^T |B_0| M_L |B_S^T| \partial_t s \\ W_q^T f(W_p p, W_q q, s) + W_p^T B_S^T s \end{bmatrix}. \quad (23)$$

Since all involved matrices can be precomputed, the only issue left is the computation of the nonlinear term  $W_q f(W_p p, W_q q, s)$ . So as to approximate this nonlinear term by a function which no longer requires any higher order operation, the discrete empirical interpolation method (DEIM in short, see [2]) is employed. If we precompute a snapshot matrix  $F$  that consists of function evaluations of  $f$  at a number of given values of  $p$  and  $q$ , matrices  $P$  and  $U$  can be computed via Algorithm 2, such that the resulting reduced (i.e., approximate) nonlinear function evaluation amounts to

$$W_q^T f(W_p p, W_q q, s) = W_q^T U (P^T U)^{-1} P^T f(W_p p, W_q q, s).$$

---

**Algorithm 2** DEIM
 

---

**Require:**  $F \in \mathbb{R}^{N \times n}$ , reduction order  $m$

**Ensure:** metamodel  $\psi_1, \dots, \psi_\ell$

- 1:  $[U, S, V] \leftarrow \text{svd}(F)$
  - 2:  $\text{ind} \leftarrow \text{index of the maximal entry of } U(:, 1)$
  - 3:  $P \leftarrow e_{\text{ind}}$  coordinate basis vector
  - 4: **for**  $i \leftarrow 1 : m$  **do**
  - 5:      $\text{ind} \leftarrow \text{index of maximal entry of } U(:, i+1) - U(P^T U)P^T U(:, i+1)$
  - 6:      $P \leftarrow [P, e_{\text{ind}}]$
- 

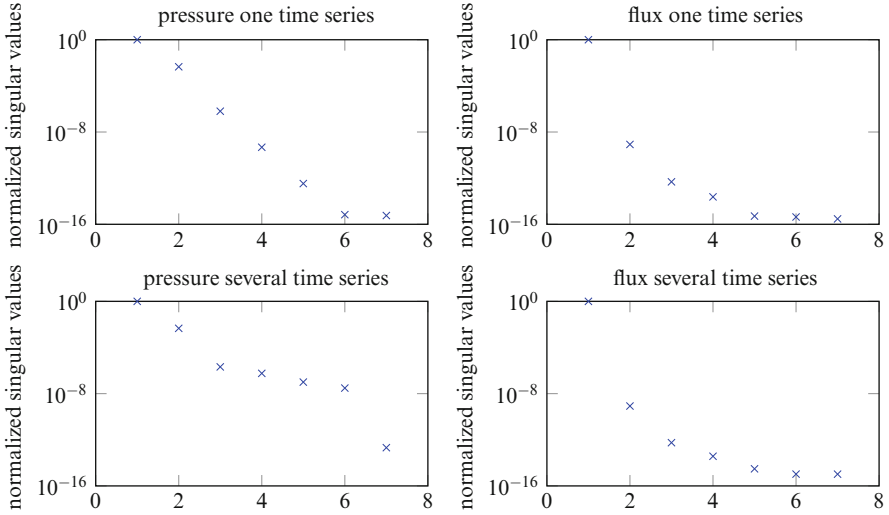
We immediately see that the matrix  $W_q^T U (P^T U)^{-1}$  is of size  $r_1 \times m$ , where  $m$  is the reduction order for DEIM and  $r_1$  the reduction order for the flux component in POD. The function  $P^T f(W_p p, W_q q, s)$  is truly a function of order  $m$  since the matrix  $P$  is a matrix of zeros and ones, where exactly  $m$  entries of  $f$  are picked. Owing to the structure of the function  $f$ , namely the  $k$ -th entry only depending on the  $k$ -th entry of  $W_q q$  and  $B_0^T W_p p + B_S^T s$ , this computation is straightforward.

## 5 Numerical Examples

The purpose of the following section is to experimentally show the extent of reduction of the system we can achieve for our three network models (Section 2.2), while still being able to reproduce the general dynamical behavior of the gas within the network.

### 5.1 The Single Pipe

Our first example is one of the most trivial networks possible and serves as proof of concept. To begin with, a single solution trajectory, which we again call a snapshot, is calculated for the parameters given in Section 2.2. Its singular values are shown in the top panel of Figure 8. By the methods explained in Section 4.2, the snapshot is used to set up a reduced order model. In short, this means we can determine projection matrices  $W_1$  and  $W_2$  via POD as well as projection and picking matrices  $U$  and  $P$  via DEIM. The resulting reduced order model parametrically depends on length, diameter, and friction coefficient of each pipe segment as well as on the field  $\gamma$  that characterizes properties of the gas flowing through it. In order to better understand the behavior of the system for different choices of these parameters, we make use of a Latin hypercube sampling of size 10, where we vary all 22 parameters in a cube of  $\pm 5\%$  of the original parameter set. In this particular test, the order of the reduced model is 4, which results from the choice  $r_1 = r_2 = m = 2$ . For the



**Fig. 8** The singular values for one time series as well as for 10 time series, each the result of a different parameter set. The values that are shown along the y-axis are ordered decreasing in size, along the x-axis

different test networks, which only differ by the said parameter values, we compute the reduced order model as well as, for reference, the full model and compute the relative error in the pressure and the absolute error in the flux, displaying the respective maximal error in Table 6. Furthermore, the relative error at a distance from  $t = 0$  is also given as the maximal difference over all pressure and flux components. We also display the error of the reduced order system compared to the full system of the original parameter distribution. This shows that, when we vary the parameters, we still stay within the same order of magnitude of the error.

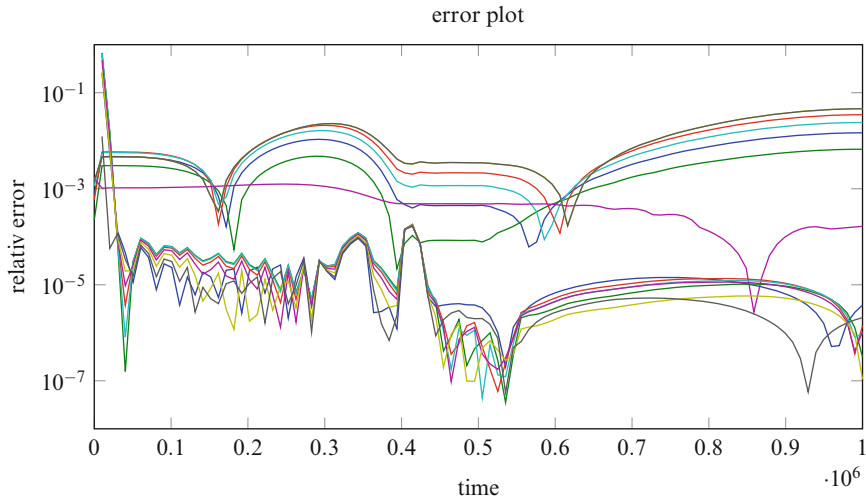
Figure 8 emphasizes this result. Here we see that the singular value decay in the flux looks similar, whether we have one time series or several time series for several parameters. The difference becomes noticeable in the left column that refers to pressure. However, in both cases the third singular value, after which DEIM cuts off the series of singular values, is of order  $10^{-8}$ .

Of course, the input and output functions of this problem are not varied. And due to the nature of the problem we cannot expect to find a reduced system of small order representing all possible input and output functions. The difficulty consists in that the partial differential equation describing the physics in the pipe, the isothermal Euler equation, is a transport dominated partial differential equation. Thus shocks might travel through the system and we can, therefore, not expect the solution to lie in a low-dimensional subspace.

The following numerical experiment, however, shows that, if we assume the supply pressures and demand nodes only vary within a certain range of given values, we are able to still use the reduced system. We are now going to repeat

**Table 6** Numerical error, network 1

Sample	1	2	3	4	5	6	7	8	9	10	original parameters
Absolute error (flux) [kg/s]	0.0393	0.4779	0.3269	0.2307	0.5856	0.1450	0.4893	0.2796	0.2631	0.2666	0.2479
Relative error (pressure)	0.0711	0.0214	0.0783	0.0301	0.0737	0.0263	0.0777	0.0571	0.0660	0.2401	0.021
Relative error (both, $\tau \in [10^5, 10^6]$ )	0.0711	0.0214	0.0783	0.0301	0.0737	0.0263	0.0777	0.0480	0.0660	0.2401	

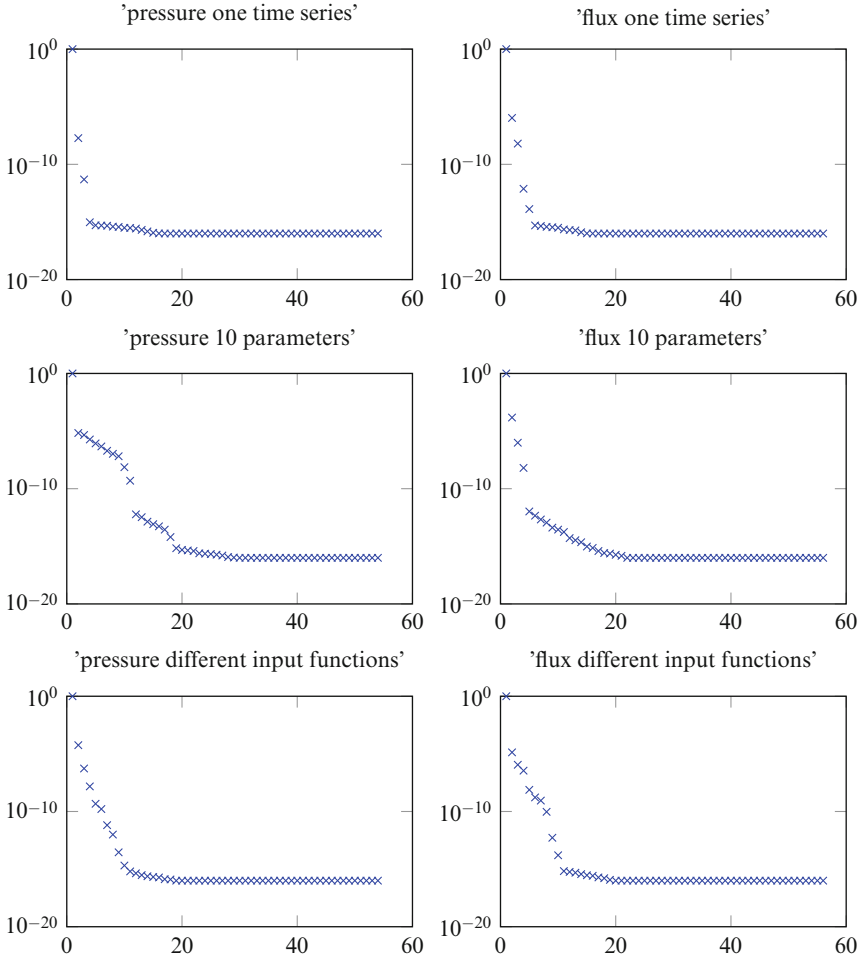


**Fig. 9** The maximal relative error over the 10 sampled scenarios

the comparison of the full and reduced simulation. Again, we pick 10 simulations, varying the percentage the supply pressure and the demand flux change over time within  $\pm 10\%$ . Remember the reduced order was created with these values all varying by 2% only. The relative error plot, where the maximum is taken over all 10 scenarios, is displayed in Figure 9.

## 5.2 Network of 57 Pipes

Having a manageable size but realistic structure, the following example supports the idea that model order reduction already helps to reduce simulation time and stability of the simulation for networks of smaller scale. We test parameter variations as well as changes of the input function receiving robust and accurate results. The system is first solved for the parameters and input function given in Section 2.2,  $T = 1 \times 10^6 \text{ s} \approx 280 \text{ h}$ . We store 100 time steps for a system size of 110, of which 54 dimensions refer to pressure and 56 to flux. By and large, we gather a snapshot matrix of dimension  $100 \times 110$  in this fashion. The singular value decay of  $Y E Y^T$  is illustrated in the top panel of Figure 10, where the singular values are displayed for the fluxes and the pressures independently again. On the subject of the criteria given in (20), we can conclude that a reduced order of size 1 for pressures and of size 1 for fluxes is sufficient to obtain a “good” approximation in that case with  $\mathcal{E} \geq 0.99999$  for the pressures as well as for the fluxes. We, however, select  $r_1 = r_2 = 3$ . Typically, the reduction order for DEIM has to equal at least this



**Fig. 10** Singular values of the snapshot matrices

reduced order (3 in our case). It turns out that this minimum choice is enough for the example at hand, see the error values listed in Table 7. In fact,  $r_1 = 3$ ,  $r_2 = 3$ ,  $m = 3$  yield a reasonably accurate reduced order system.

The projection matrices are applied to create a parametric reduced order model. The parameter space of this parametrized model possesses a dimension of size 29. We allow this parameter vector to vary in a box around  $\pm 5\%$  of its given values again. We pick 10 values in that box by Latin hypercube sampling to obtain 10 scenarios. For each scenario we compute the reduced solution and the full solution. We realize here that in order to run MATLAB's ode solver `ode15s` we need to have a very accurate starting value for the full model. Therefore, we take the original initial value and compute a truly stationary solution by Newton's method to be used as initial value. For the reduced order system this step is not necessary,



**Table 7** Numerical error, network 2

Sample	1	2	3	4	5	6	7	8	9	10
Absolute error (flux) [kg/s]	10.6854	3.4017	2.3185	10.6690	4.3558	4.0095	9.1439	14.2883	12.9632	5.0359
Relative error (pressure)	0.0296	0.0426	0.0421	0.0340	0.0300	0.0224	0.0388	0.0312	0.0429	0.0171
Relative error (pressure, $t \in [10^3 \text{ s}, 10^6 \text{ s}]$ )	0.0296	0.0348	0.0221	0.0916	0.0346	0.0096	0.0388	0.0831	0.0573	0.0216
Time (full model) [s]	1,388	904	1,029	1,320	1,060	878	941	940	811	914
Time (reduced model) [s]	2	1.7	1.6	1.8	1.7	1.7	1.8	1.4	1.1	1

which in turn means that the solution of the full order system becomes even slower in comparison. However, this procedure also implies that the solutions near the initial time are not as close to each other as later because they do not start from the same initial value. We thus compare the maximal relative error only after the first 1000 s have already passed. This error is shown in Table 7. We, furthermore, compare the timings, for the solution of the reduced system and for the solution of the full system, where the identification of a starting value is included as well as the simulation time itself.

With regard to the full solution of the 10 scenarios from above, creating a snapshot matrix as in Section 4.1, we can compute the singular values that are plotted in the middle panel of Figure 10. The singular values show the size of a linear subspace to be used in order to create a good approximation of the space in which the solutions lie.

Furthermore, we have conducted some test with varying input functions. As in the case of the single pipe, we vary the percentages the demand and supply values change during the simulated time in a box of  $\pm 10\%$ . We can match the general behavior even though the errors between the full and the reduced system can be large. However, measurements often differ from the computed solution even stronger than our reduced model, see, e.g., [1]. We, furthermore, display the singular value decay for the snapshot matrix created by 10 such scenarios (last panel of Figure 10).

### 5.3 Network of 669 Pipes

A system of 669 elements is rather large such that computations of a single time series can already take from several hours up to weeks with standard solvers (it is possible to accelerate the implementation, though). The goal here is to show that we can predict the behavior of the system with the model order reduction method as described above up to a certain extent. We only compute the solution of this large system for the first 100 s. From the results of this computation we calculate the projection matrix  $W$  as well as the DEIM matrices  $P$  and  $U$  with reduction order  $r_1 = 4, r_2 = 2, m = 4$ . These matrices are needed to set up the reduced system. We then run the reduced system for different values of  $d(0)$ , which are a random perturbation of the original  $d(0)$ . We cannot compare the reduced time series to the time series of the full system, as the latter is too expensive to compute. Since we run the reduced system in time until it should arrive at a new stationary solution, we can assess the extent of “stationarity” of our result from the right-hand side of the system. In Table 8 we compare the value of the right-hand side of the reduced system to the value of the right-hand side of the full system at the state we arrived at following the trajectory of the reduced system. Even though these values are not numerically zero, they are reputedly closer to zero than if we evaluated the right-hand side at the initial configuration. Compared to the initial configuration, these values are hence better starting values to use a Newton-type method to find a truly stationary solution.

**Table 8** Value of the right-hand side, network 3

Sample	1	2	3	4	5	6	7	8	9	10
$f_r(r_E, x_E)$	0.4	0.3	2.7	0.9	1.3	0.4	0.6	1.4	0.6	1.5
$f(t_E, Wx_E)$	18	22	29	19	20	17	26	22	18	22
$f(t_E, x_0)$	$10^6$	$10^6$	$10^6$	$10^6$	$10^6$	$10^6$	$10^6$	$10^6$	$10^6$	$10^6$

The simulation time for simulating those reduced order systems amounts to approximately 25 s. If we use the end value obtained from the reduced simulation and apply a Newton method to find a stationary solution such that  $|f(x_S)| \leq 10^{-3}$ , the Newton method takes around 10 to 100 s. Whereas, if we use  $x_0$  to start our Newton method, it takes about 10,000 s. So in the presented case, the reduced method can certainly help to speed up the computation of a feasible stationary solution.

## 6 Conclusions

For gas transportation problems through networks of pipelines, from supplier to consumer, the underlying continuum mechanics, practical simplifications, and a numerical model have been compiled and explained to the reader. Notably, a state-of-the-art transformation of the resulting discretized differential algebraic system of equations to a system of ordinary differential equations has been presented. Among the properties of this system, stiffness has been spotlighted since it affects the choice of suitable numerical solvers strongly. Empirical evidence is shown that stiffness plays an important role and that it even grows with the complexity of the systems.

With the goal of exhaustive studies of varying input and output in mind, we indicate how to apply model order reduction methods for nonlinear systems efficiently, making use of POD with DEIM. Special attention is given to the preservation of the general form of the underlying ordinary differential equation lest pressure and flux values may be mixed. A parameter domain is defined by varying network properties within a limited range to indicate uncertainty. The domain is used to obtain empirical results that illustrate possible gain in simulation speed by order reduction, while the induced loss in accuracy is discussed.

The numerical behavior of three networks of different complexity is examined in this light. Results indicate that a speed-up of 10 to 1000 can be achieved, where we have to anticipate certain inaccuracy within the obtained solutions. In some cases inaccuracy can be of an order of magnitude such as 10% of the typical solution values. The stiffness of the reduced system has been monitored and not found to be increased by the applied model order reduction technique.

One of the most obvious extensions to the case presented here can be found in the treatment of more types of network elements, as well as in the inclusion of more involved temperature effects. From a numerical point of view, the development of dedicated solvers would probably be interesting.

## References

1. Alamian, R., Behbahani-Nejad, M., Ghanbarzadeh, A.: A state space model for transient flow simulation in natural gas pipelines. *J. Nat. Gas Sci. Eng.* **9**, 51–59 (2012)
2. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32**(5), 2737–2764 (2010). doi:[10.1137/090766498](https://doi.org/10.1137/090766498)
3. Grundel, S., Hornung, N., Klaassen, B., Benner, P., Clees, T.: Computing surrogates for gas network simulation using model order reduction. In: *Surrogate-Based Modeling and Optimization*, pp. 189–212. Springer, New York (2013)
4. Grundel, S., Jansen, L., Hornung, N., Clees, T., Tischendorf, C., Benner, P.: Model order reduction of differential algebraic equations arising from the simulation of gas transport networks. In: Schöps, S., Bartel, A., Günther, M., ter Maten, E.J.W., Müller, P.C. (eds.) *Progress in Differential-Algebraic Equations, Differential-Algebraic Equations Forum*, pp. 183–205. Springer, Berlin/Heidelberg (2014). doi:[10.1007/978-3-662-44926-4\\_9](https://doi.org/10.1007/978-3-662-44926-4_9)
5. Haasdonk, B.: Convergence rates of the POD–greedy method. *ESAIM: M2AN* **47**(3), 859–873 (2013). doi:[10.1051/m2an/2012045](https://doi.org/10.1051/m2an/2012045). <http://dx.doi.org/10.1051/m2an/2012045>
6. Herty, M., Seaid, M.: Simulation of transient gas flow at pipe-to-pipe intersections. *Int. J. Numer. Methods Fluids* **56**(5), 485–506 (2008). doi:[10.1002/flid.1531](https://doi.org/10.1002/flid.1531). <http://dx.doi.org/10.1002/flid.1531>
7. Higham, D., Trefethen, L.: Stiffness of ODEs. *BIT Numer. Math.* **33**(2), 285–303 (1993). doi:[10.1007/BF01989751](https://doi.org/10.1007/BF01989751)
8. Hinze, M., Kunkel, M.: Discrete empirical interpolation in POD model order reduction of drift-diffusion equations in electrical networks. In: *Scientific Computing in Electrical Engineering SCEE 2010*, pp. 423–431. Springer (2012)
9. Kiuchi, T.: An implicit method for transient gas flows in pipe networks. *Int. J. Heat Fluid Flow* **15**(5), 378–383 (1994)
10. Osiadacz, A.: Simulation of transient gas flows in networks. *Int. J. Numer. Methods Fluids* **4**(1), 13–24 (1984). doi:[10.1002/flid.1650040103](https://doi.org/10.1002/flid.1650040103). <http://dx.doi.org/10.1002/flid.1650040103>
11. Schmidt, M., Steinbach, M.C., Willert, B.M.: High detail stationary optimization models for gas networks. *Optimization and Engineering*, **16**(1), 131–164 (2015). doi:[10.1007/s11081-014-9246-x](https://doi.org/10.1007/s11081-014-9246-x)
12. Swamee, P.K., Jain, A.K.: Explicit equations for pipe-flow problems. *J. Hydraul. Div.* **102**(5), 657–664 (1976)
13. Volkwein, S.: Optimal control of a phase-field model using proper orthogonal decomposition. *ZAMM – J. Appl. Math. Mech./ Z. Angew. Math. Mech.* **81**(2), 83–97 (2001). doi:[10.1002/1521-4001\(200102\)81:2<83::AID-ZAMM83>3.0.CO;2-R](https://doi.org/10.1002/1521-4001(200102)81:2<83::AID-ZAMM83>3.0.CO;2-R)