

On the Loewner Framework for Model Reduction of Burgers' Equation



Athanasios C. Antoulas, Ion Victor Gosea and Matthias Heinkenschloss

Abstract This paper addresses issues that originate in the extension of the Loewner framework to compute reduced order models (ROMs) of so-called quadratic-bilinear systems. The latter arise in semi-discretizations of fluid flow problems, such as Burgers' equation or the Navier-Stokes equations. In the linear case, the Loewner framework is data-driven and constructs a ROM from measurements of the transfer function; it does not explicitly require access to the system matrices, which is attractive in many settings. Research on extending the Loewner framework to quadratic-bilinear systems is ongoing. This paper presents one extension and provides details of its implementation that allow application to large-scale problems. This extension is applied to Burgers' equation. Numerical results show the potential of the Loewner framework, but also expose additional issues that need to be addressed to make it fully applicable. Possible approaches to deal with some of these issues are outlined.

Keywords Model reduction · Loewner framework · Rational interpolation
Burgers' equation

A. C. Antoulas

Department of Electrical and Computer Engineering, Rice University,
6100 Main Street MS-380, Houston, TX 77005-1892, USA
e-mail: aca@rice.edu

I. V. Gosea · A. C. Antoulas

Max Planck Institute for Dynamics of Complex Technical Systems Magdeburg,
Sandtorstrasse 1, 39106 Magdeburg, Germany
e-mail: gosea@mpi-magdeburg.mpg.de

M. Heinkenschloss (✉)

Department of Computational and Applied Mathematics, Rice University,
6100 Main Street MS-134, Houston, TX 77005-1892, USA
e-mail: heinken@rice.edu

© Springer Nature Switzerland AG 2019

R. King (ed.), *Active Flow and Combustion Control 2018*,
Notes on Numerical Fluid Mechanics and Multidisciplinary Design 141,
https://doi.org/10.1007/978-3-319-98177-2_16

1 Introduction

We study the extension of the Loewner framework to compute reduced order models (ROMs) of so-called quadratic-bilinear systems that arise in semi-discretizations of fluid flow problems, such as Burgers' equation or the Navier-Stokes equations. The attractive feature of the Loewner framework is that it is data-driven and constructs a ROM from measurements of transfer functions related to the system. In the linear case it does not explicitly require access to the matrices (such as $\mathbf{E}$, $\mathbf{A}$, etc. defined below) of the system. While the Loewner framework for linear systems is fairly developed, see e.g., the tutorial paper by Antoulas et al. [3], its extension to quadratic-bilinear systems is ongoing. Recently, Antoulas et al. [2, 8] have extended the Loewner framework to bilinear and quadratic-bilinear systems, but have not yet addressed several issues that need to be dealt with when applying the Loewner framework to fluid flow problems. This paper begins to address some of these issues. In particular, we present the Loewner framework from [8] in a way that allows its efficient implementation and scaling to large-scale problems. In addition, we present more extensive numerical results of an initial version of the Loewner framework to Burgers' equation. These numerical results show the potential of the Loewner framework, but also highlight open issues that need to be addressed. We will outline possible paths on how to deal with these issues.

Given

$$\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n, d \in \mathbb{R}, \tilde{\mathbf{N}} : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n, \mathbf{G} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n, \quad (1)$$

we consider single input and single output (SISO) systems of the type

$$\mathbf{E} \frac{d}{dt} \mathbf{x}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{G}(\mathbf{x}(t), \mathbf{x}(t)) + \tilde{\mathbf{N}}(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{b} \mathbf{u}(t), \quad t \in (0, T), \quad (2a)$$

$$\mathbf{y}(t) = \mathbf{c}^T \mathbf{x}(t) + d \mathbf{u}(t), \quad t \in (0, T), \quad (2b)$$

$$\mathbf{x}(0) = \mathbf{0}. \quad (2c)$$

We will make additional assumptions on (2) in Sect. 2. The system (2) is referred to as the full order model (FOM). The semi-discretized viscous Burgers' equations and the Navier-Stokes equations fit into the setting (2). Burgers' equation will be discussed in Sect. 4. We refer to, e.g., Elman et al. [6, Chap. 8], Layton [11] for the semi-discretized Navier Stokes equations, where (2a) is a system of differential algebraic equations for the velocity and pressure.

Our goal is to construct a small ($r \ll n$) dimensional model

$$\hat{\mathbf{E}} \frac{d}{dt} \hat{\mathbf{x}}(t) = \hat{\mathbf{A}} \hat{\mathbf{x}}(t) + \hat{\mathbf{G}}(\hat{\mathbf{x}}(t), \hat{\mathbf{x}}(t)) + \hat{\mathbf{N}}(\hat{\mathbf{x}}(t), \mathbf{u}(t)) + \hat{\mathbf{b}} \mathbf{u}(t), \quad t \in (0, T), \quad (3a)$$

$$\hat{\mathbf{y}}(t) = \hat{\mathbf{c}}^T \hat{\mathbf{x}}(t) + d \mathbf{u}(t), \quad t \in (0, T), \quad (3b)$$

$$\hat{\mathbf{x}}(0) = \mathbf{0}, \quad (3c)$$

with $\widehat{\mathbf{E}}, \widehat{\mathbf{A}} \in \mathbb{R}^{r \times r}$, $\widehat{\mathbf{b}}, \widehat{\mathbf{c}} \in \mathbb{R}^r$, $\widehat{\mathbf{N}} : \mathbb{R}^r \times \mathbb{R} \rightarrow \mathbb{R}^r$, $\widehat{\mathbf{G}} : \mathbb{R}^r \times \mathbb{R}^r \rightarrow \mathbb{R}^r$ such that the input-to-output map $\mathbf{u} \mapsto \widehat{\mathbf{y}}$ of the ROM (3) is a good approximation of the input-to-output map $\mathbf{u} \mapsto \mathbf{y}$ of the FOM (2).

The majority of ROM approaches use a Petrov-Galerkin projection. That is they compute matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ of rank $r \ll n$, approximate the state of the FOM (2) by $\mathbf{x}(y) \approx \mathbf{V}\widehat{\mathbf{x}}(t)$, insert this approximation into (2), and multiply the resulting Eq. (2a) from the left by $\mathbf{W}^T$. This results in a ROM (3) with

$$\begin{aligned} \widehat{\mathbf{E}} &= \mathbf{W}^T \mathbf{E} \mathbf{V}, \quad \widehat{\mathbf{A}} = \mathbf{W}^T \mathbf{A} \mathbf{V}, \quad \widehat{\mathbf{b}} = \mathbf{W}^T \mathbf{b}, \quad \widehat{\mathbf{c}} = \mathbf{V}^T \mathbf{c}, \\ \widehat{\mathbf{G}}(\widehat{\mathbf{x}}, \widehat{\mathbf{z}}) &= \mathbf{W}^T \mathbf{G}(\mathbf{V}\widehat{\mathbf{x}}, \mathbf{V}\widehat{\mathbf{z}}), \quad \widehat{\mathbf{N}}(\widehat{\mathbf{x}}, \mathbf{u}) = \mathbf{W}^T \mathbf{N}(\mathbf{V}\widehat{\mathbf{x}}, \mathbf{u}). \end{aligned} \quad (4)$$

Examples of projection based ROM approaches include proper orthogonal decomposition (POD), reduced basis (RB) methods, and Krylov subspace methods. These and other ROM approaches are described, e.g., in the books by Antoulas [1], Hes-thaven et al. [10], Quarteroni et al. [13], or the review paper by Rozza et al. [15]. Rowley and Dawson [14] review ROMs for flow problems. These methods have in common that they compute $\mathbf{V}, \mathbf{W}$ and then project the system. Thus these methods need the matrices $\mathbf{E}, \mathbf{A}$, etc., to compute the ROM matrices $\widehat{\mathbf{E}}, \widehat{\mathbf{A}}$, etc. In contrast, for linear systems, the Loewner framework computes the ROM system matrices $\widehat{\mathbf{E}}, \widehat{\mathbf{A}}$, etc., directly from measurements of the transfer function, which in some cases can be obtained from experiments. In the case where values of the transfer function can be obtained from measurements, the Loewner framework does not require knowledge of the FOM matrices $\mathbf{E}, \mathbf{A}$, etc. See the tutorial paper by Antoulas et al. [3] and Sect. 2. We study the extension of the Loewner framework to compute ROMs of (2).

To focus our presentation we consider SISO systems (2). However, the Loewner framework and the material in this paper can be extended to multiple input and multiple output (MIMO) systems using so-called tangential interpolation, but this extension is technical and beyond the scope of this paper. For the reduction of linear MIMO systems using the Loewner framework see [3].

Since we have only one input, the bilinear map $\widehat{\mathbf{N}}$ in (1) can be written in terms of a matrix $\mathbf{N} \in \mathbb{R}^{n \times n}$,

$$\widehat{\mathbf{N}}(\mathbf{x}, \mathbf{u}) = \mathbf{N}\mathbf{x}\mathbf{u} \quad \text{for all } \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}. \quad (5a)$$

Furthermore, given the bilinear map $\mathbf{G}$ in (1) there exists a matrix $\mathbf{Q} \in \mathbb{R}^{n \times n^2}$ such that the bilinear map can be written using $\mathbf{Q}$ and the Kronecker product $\otimes$ as follows

$$\mathbf{G}(\mathbf{x}, \mathbf{z}) = \mathbf{Q} \mathbf{x} \otimes \mathbf{z} \quad \text{for all } \mathbf{x}, \mathbf{z} \in \mathbb{R}^n. \quad (5b)$$

Thus, the system (2) can be written as

$$\mathbf{E} \frac{d}{dt} \mathbf{x}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{Q} \mathbf{x}(t) \otimes \mathbf{x}(t) + \mathbf{N} \mathbf{x}(t) \mathbf{u}(t) + \mathbf{b} \mathbf{u}(t), \quad (6a)$$

$$\mathbf{y}(t) = \mathbf{c}^T \mathbf{x}(t) + d \mathbf{u}(t). \quad (6b)$$

The representation (6) of quadratic bilinear systems is used in previous model reduction approaches. See, e.g., the papers by Benner and Breiten [4], Breiten and Damm [5], Gosea and Antoulas [8], and Gu [9]. However, for some applications, incl. the Burgers' equation or the Navier-Stokes equations, the construction of $\mathbf{Q}$ is impractical and the use of (6) tends to be numerically inefficient. As mentioned before, the Burgers' equation or the Navier-Stokes equations are written in the form (2). Therefore we use (6) only to introduce the ideas, but use (2) for computations.

We use the system representation (6) to review the Loewner framework in Sect. 2. In Sect. 3 we show how the Loewner framework for (6) can be efficiently applied directly to (2). This makes the Loewner framework applicable to a number of flow problems. In addition, the implementation details given in Sect. 3 can also be used for other ROM approaches that are based on the representation (6), such as [4, 5], [9]. Section 4 illustrates the performance of the Loewner approach applied to Burgers' equation. The Conclusion Sect. 5 summarizes our findings and outlines some areas of current and future research.

2 The Loewner Framework

We begin with a review of the Loewner framework for the linear case ($\mathbf{Q} = \mathbf{0}$ and $\mathbf{N} = \mathbf{0}$) because it provides important insight into basic ideas and is the foundation for its extension to the quadratic bilinear case, which will be discussed in the second part of this section. The implementation of the Loewner framework in the context of the system (2) will be discussed later in Sect. 3.

To simplify our presentation, we assume that $\mathbf{E}$ is nonsingular and that $d = 0$. These assumptions can be relaxed, especially in the linear case, see [3].

2.1 Linear Systems

We consider linear systems (6) with $\mathbf{Q} = \mathbf{0}$ and $\mathbf{N} = \mathbf{0}$. The input-to-output map of the linear system in frequency domain is given by the transfer function

$$\mathbf{H}(s) = \mathbf{c}^T (s\mathbf{E} - \mathbf{A})^{-1} \mathbf{b}. \quad (7)$$

The Loewner framework uses measurements of the transfer function (7) to construct $\widehat{\mathbf{E}}, \widehat{\mathbf{A}} \in \mathbb{R}^{r \times r}$ and $\widehat{\mathbf{b}}, \widehat{\mathbf{c}} \in \mathbb{R}^r$ such that the transfer function

$$\widehat{\mathbf{H}}(s) = \widehat{\mathbf{c}}^T (s\widehat{\mathbf{E}} - \widehat{\mathbf{A}})^{-1} \widehat{\mathbf{b}} \quad (8)$$

of the corresponding ROM linear system matches or, depending on the amount of data given, approximates the original transfer function (7) at the measurements. Specifically, we want to construct a ROM so that its transfer function (8) matches or

approximates the transfer function (7) at the left driving frequencies¹ $\mu_1, \dots, \mu_k \in \mathbb{C}$ and at the right driving frequencies $\lambda_1, \dots, \lambda_k \in \mathbb{C}$. Given distinct frequencies $\mu_1, \dots, \mu_k$ and $\lambda_1, \dots, \lambda_k$, and corresponding transfer function (7) measurements $\mathbf{H}(\mu_1), \dots, \mathbf{H}(\mu_k)$ and $\mathbf{H}(\lambda_1), \dots, \mathbf{H}(\lambda_k)$, we consider the Loewner matrix²

$$\mathbb{L} = \begin{bmatrix} \frac{\mathbf{H}(\mu_1) - \mathbf{H}(\lambda_1)}{\mu_1 - \lambda_1} & \dots & \frac{\mathbf{H}(\mu_1) - \mathbf{H}(\lambda_k)}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mathbf{H}(\mu_k) - \mathbf{H}(\lambda_1)}{\mu_k - \lambda_1} & \dots & \frac{\mathbf{H}(\mu_k) - \mathbf{H}(\lambda_k)}{\mu_k - \lambda_k} \end{bmatrix} \in \mathbb{C}^{k \times k} \quad (9a)$$

and the shifted Loewner matrix

$$\mathbb{L}_s = \begin{bmatrix} \frac{\mu_1 \mathbf{H}(\mu_1) - \mathbf{H}(\lambda_1) \lambda_1}{\mu_1 - \lambda_1} & \dots & \frac{\mu_1 \mathbf{H}(\mu_1) - \mathbf{H}(\lambda_k) \lambda_k}{\mu_1 - \lambda_k} \\ \vdots & \ddots & \vdots \\ \frac{\mu_k \mathbf{H}(\mu_k) - \mathbf{H}(\lambda_1) \lambda_1}{\mu_k - \lambda_1} & \dots & \frac{\mu_k \mathbf{H}(\mu_k) - \mathbf{H}(\lambda_k) \lambda_k}{\mu_k - \lambda_k} \end{bmatrix} \in \mathbb{C}^{k \times k}. \quad (9b)$$

The desired ROM is constructed from the Loewner and shifted Loewner matrices. There are two cases: (1) The ‘right’ amount of data is available, and (2) a redundant amount of data is provided. We focus on the second case, which is more relevant in practice.

In the second case the pencil $(\mathbb{L}_s, \mathbb{L})$ is singular and we use the singular value decomposition (SVD) to reduce the data. Consider the (short) SVDs

$$[\mathbb{L} \quad \mathbb{L}_s] = \mathbf{Y}_1 \Sigma_1 \mathbf{X}_1^*, \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_2 \Sigma_2 \mathbf{X}_2^*, \quad (10)$$

where $\Sigma_1 \in \mathbb{R}^{k \times 2k}$, $\Sigma_2 \in \mathbb{R}^{2k \times k}$, $\mathbf{Y}_1, \mathbf{X}_2 \in \mathbb{C}^{k \times k}$. A ROM is now computed as follows. The matrices $\mathbf{Y}, \mathbf{X} \in \mathbb{C}^{k \times r}$ are obtained by selecting the first r columns of the matrices $\mathbf{Y}_1$ and $\mathbf{X}_2$. The ROM is

$$\begin{aligned} \hat{\mathbf{E}} &= -\mathbf{Y}^* \mathbb{L} \mathbf{X}, \quad \hat{\mathbf{A}} = -\mathbf{Y}^* \mathbb{L}_s \mathbf{X}, \\ \hat{\mathbf{b}} &= \mathbf{Y}^* (\mathbf{H}(\mu_1), \dots, \mathbf{H}(\mu_k))^T, \quad \hat{\mathbf{c}}^T = (\mathbf{H}(\lambda_1), \dots, \mathbf{H}(\lambda_k)) \mathbf{X}. \end{aligned} \quad (11)$$

The transfer function (8) of the ROM (11) approximates the transfer function of the FOM at the driving frequencies, $\hat{\mathbf{H}}(\mu_j) \approx \mathbf{H}(\mu_j)$, $\hat{\mathbf{H}}(\lambda_j) \approx \mathbf{H}(\lambda_j)$, $j = 1, \dots, k$. Expressions for the error $|\hat{\mathbf{H}}(s) - \mathbf{H}(s)|$ are given in [3, p. 359]. The Loewner framework provides a trade-off between accuracy and complexity of the ROM by means of the singular values of the Loewner matrices $\mathbb{L}$ and $\mathbb{L}_s$.

¹The reason for the terminology ‘left’ and ‘right’ driving frequencies only becomes clear in the MIMO case and is adopted here for consistency with [3].

²We point out an important difference in notation between this paper and [3]. In [3], $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{C}^{p \times k}$ and $\mathbf{V}^T = (\mathbf{v}_1, \dots, \mathbf{v}_q) \in \mathbb{C}^{m \times q}$ are matrices (in the SISO case $m = 1$, $p = 1$ vectors) of transfer function (7) measurements, whereas in our paper $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ are the Petrov-Galerkin projection matrices in projection based ROMs (4).

The ROM (11) is in general complex. However, if data $\mu_1, \dots, \mu_k, \lambda_1, \dots, \lambda_k$, and consequently $\mathbf{H}(\mu_1), \dots, \mathbf{H}(\mu_k)$ and $\mathbf{H}(\lambda_1), \dots, \mathbf{H}(\lambda_k)$, contain also the conjugate complex data ($\{\mu_j\}_{j=1}^k = \{\bar{\mu}_j\}_{j=1}^k, \{\lambda_j\}_{j=1}^k = \{\bar{\lambda}_j\}_{j=1}^k$), then the complex ROM (11) can be transformed into a real ROM with the same transfer function, as shown in [3, p. 360].

An alternative view of the above Loewner framework, which is important for the extension of this framework, is obtained by considering the generalized controllability and observability matrices. Define

$$\Phi(s) = (s\mathbf{E} - \mathbf{A})^{-1}. \quad (12)$$

Given the left and right driving frequencies $\mu_1, \dots, \mu_k, \lambda_1, \dots, \lambda_k$, the generalized controllability matrix is

$$\mathcal{R} = [\Phi(\lambda_1)\mathbf{b}, \Phi(\lambda_2)\mathbf{b}, \dots, \Phi(\lambda_k)\mathbf{b}] \in \mathbb{C}^{n \times k} \quad (13a)$$

and the generalized observability matrix is

$$\mathcal{O}^T = [\Phi(\mu_1)^T \mathbf{c}, \Phi(\mu_2)^T \mathbf{c}, \dots, \Phi(\mu_k)^T \mathbf{c}] \in \mathbb{C}^{n \times k}. \quad (13b)$$

The Loewner matrices can be factored in terms of the generalized controllability/observability matrices as

$$\mathbb{L} = -\mathcal{O}\mathbf{E}\mathcal{R}, \quad \mathbb{L}_s = -\mathcal{O}\mathbf{A}\mathcal{R}. \quad (14)$$

Note also that the following relations hold

$$(\mathbf{H}(\lambda_1), \dots, \mathbf{H}(\lambda_k)) = \mathbf{c}^T \mathcal{R}, \quad (\mathbf{H}(\mu_1), \dots, \mathbf{H}(\mu_k))^T = \mathcal{O} \mathbf{b}. \quad (15)$$

If we define

$$\mathbf{V} = \mathcal{R}\mathbf{X} \in \mathbb{C}^{n \times r}, \quad \mathbf{W} = \mathcal{O}^* \mathbf{Y} \in \mathbb{C}^{n \times r}, \quad (16)$$

then (11), (14) and (15) show that the Loewner ROM (11) can also be written as

$$\hat{\mathbf{E}} = \mathbf{W}^* \mathbf{E} \mathbf{V}, \quad \hat{\mathbf{A}} = \mathbf{W}^* \mathbf{A} \mathbf{V}, \quad \hat{\mathbf{b}} = \mathbf{W}^* \mathbf{b}, \quad \hat{\mathbf{c}} = \mathbf{V}^* \mathbf{c}. \quad (17)$$

Thus while the Loewner ROM (11) can be obtained directly from data via the Loewner and shifted Loewner matrices (9), it could also be obtained via Petrov-Galerkin projection.

2.2 Quadratic Bilinear Systems

A starting point to extend the Loewner framework and other ROM approaches to the quadratic bilinear case is an expansion of the original system (6) using Carleman

linearization or the so-called variational equation approach. See, e.g., Rugh [16, Chap. 3]. We use the latter approach. This expansion is justified for so-called linear-analytic state equations [16, Theorem 3.1], which include (6).

Consider inputs of the form $\mathbf{a}\mathbf{u}(t)$, for a positive scalar a , and assume that the solution of (6a) can be written as a power series $\mathbf{x}(t) = \sum_{\ell=1}^{\infty} a^{\ell} \mathbf{x}_{\ell}(t)$. Inserting this representation into (6a) and equating a^{ℓ} terms leads to

$$\mathbf{E} \frac{d}{dt} \mathbf{x}_1(t) = \mathbf{A} \mathbf{x}_1(t) + \mathbf{b} \mathbf{u}(t), \quad \ell = 1, \quad (18a)$$

$$\mathbf{E} \frac{d}{dt} \mathbf{x}_2(t) = \mathbf{A} \mathbf{x}_2(t) + \mathbf{Q} \mathbf{x}_1(t) \otimes \mathbf{x}_1(t) + \mathbf{N} \mathbf{x}_1(t) \mathbf{u}(t), \quad \ell = 2, \quad (18b)$$

$$\vdots$$

with homogeneous initial data. We temporarily assume that $\mathbf{E} = \mathbf{I}$ is the identity to simplify notation. Since $\mathbf{E}$ is nonsingular we can always achieve this by multiplying (18) by $\mathbf{E}^{-1}$. The solutions of (18) are $\mathbf{x}_1(t) = \int_0^t e^{\mathbf{A}\tau_1} \mathbf{b} \mathbf{u}(t - \tau_1) d\tau_1$, $\mathbf{x}_2(t) = \int_0^t e^{\mathbf{A}\tau_2} [\mathbf{Q} \mathbf{x}_1(t - \tau_2) \otimes \mathbf{x}_1(t - \tau_2) + \mathbf{N} \mathbf{x}_1(t - \tau_2) \mathbf{u}(t - \tau_2)] d\tau_2$, etc. The expression for $\mathbf{x}_1$ is inserted to obtain a representation of $\mathbf{x}_2$ that depends only in $\mathbf{u}$. Expressions for other solution components can be obtained analogously.

Given the expansion for $\mathbf{x}$, the output (6b) (recall $d = 0$) is $\mathbf{y}(t) = \sum_{\ell=1}^{\infty} \mathbf{c}^T \mathbf{x}_{\ell}(t)$. We truncate after $\ell = 2$ and insert the expressions for $\mathbf{x}_1$ and $\mathbf{x}_2$ to obtain the approximate output

$$\begin{aligned} \mathbf{y}(t) = & \int_0^t h_1(\tau_1) \mathbf{u}(t - \tau_1) d\tau_1 + \int_0^t \int_0^{t-\tau_2} h_2(\tau_1, \tau_2) \mathbf{u}(t - \tau_1 - \tau_2) \mathbf{u}(t - \tau_2) d\tau_1 d\tau_2 \\ & + \int_0^t \int_0^{t-\tau_3} \int_0^{t-\tau_3} h_3(\tau_1, \tau_2, \tau_3) \mathbf{u}(t - \tau_1 - \tau_3) \mathbf{u}(t - \tau_2 - \tau_3) d\tau_1 d\tau_2 d\tau_3, \end{aligned}$$

where $h_1(\tau_1) = \mathbf{c}^T e^{\mathbf{A}\tau_1} \mathbf{b}$, $h_2(\tau_1, \tau_2) = \mathbf{c}^T e^{\mathbf{A}\tau_2} \mathbf{N} e^{\mathbf{A}\tau_1} \mathbf{b}$, and $h_3(\tau_1, \tau_2, \tau_3) = \mathbf{c}^T e^{\mathbf{A}\tau_3} \mathbf{Q} (e^{\mathbf{A}\tau_2} \mathbf{b} \otimes e^{\mathbf{A}\tau_1} \mathbf{b})$. Application of the multivariate Laplace transform leads to the generalized rational transfer functions, which correspond to the kernels h_1, h_2, h_3 . The Loewner framework constructs ROMs that approximate these transfer functions at certain frequencies. We sketch the approach and refer to [8] for details.

As in the linear case, the interpolation points are partitioned into two disjoint sets of left and right interpolation points. Since the transfer functions corresponding to (18) depend on multiple frequencies (with our truncation on up to three frequencies), the interpolation points need to be arranged in a suitable way. To simplify the presentation, assume that we have $k = 3\bar{k}$ left and right interpolation points, which are renamed as follows:

$$\mu_1^{(1)}, \mu_2^{(1)}, \mu_3^{(1)}, \dots, \mu_1^{(\bar{k})}, \mu_2^{(\bar{k})}, \mu_3^{(\bar{k})}, \quad \lambda_1^{(1)}, \lambda_2^{(1)}, \lambda_3^{(1)}, \dots, \lambda_1^{(\bar{k})}, \lambda_2^{(\bar{k})}, \lambda_3^{(\bar{k})}. \quad (19)$$

Next, the left and right interpolation points are grouped in multi-tuples

$$\boldsymbol{\mu}^{(j)} = \{(\mu_1^{(j)}), (\mu_1^{(j)}, \mu_2^{(j)}), (\mu_1^{(j)}, \lambda_1^{(j)}, \mu_3^{(j)})\}, \quad j = 1, \dots, \bar{k}, \quad (20a)$$

$$\boldsymbol{\lambda}^{(j)} = \{(\lambda_1^{(j)}), (\lambda_2^{(j)}, \lambda_1^{(j)}), (\lambda_3^{(j)}, \lambda_1^{(j)}, \lambda_1^{(j)})\}, \quad j = 1, \dots, \bar{k}. \quad (20b)$$

The generalized controllability matrix $\mathcal{R} \in \mathbb{C}^{n \times k}$ associated with the right multi-tuples $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}, \dots, \boldsymbol{\lambda}^{(\bar{k})}$ is

$$\mathcal{R} = \left[\mathcal{R}^{(1)}, \mathcal{R}^{(2)}, \dots, \mathcal{R}^{(\bar{k})} \right], \quad (21)$$

where the matrices $\mathcal{R}^{(j)} \in \mathbb{C}^{n \times 3}$, $j = 1, \dots, \bar{k}$, are associated with the j -th multi-tuple $\boldsymbol{\lambda}^{(j)}$ in (20) are given by

$$\mathcal{R}^{(j)} = \left[\boldsymbol{\Phi}(\lambda_1^{(j)}) \mathbf{b}, \boldsymbol{\Phi}(\lambda_2^{(j)}) \mathbf{N} \boldsymbol{\Phi}(\lambda_1^{(j)}) \mathbf{b}, \boldsymbol{\Phi}(\lambda_3^{(j)}) \mathbf{Q}(\boldsymbol{\Phi}(\lambda_1^{(j)}) \mathbf{b} \otimes \boldsymbol{\Phi}(\lambda_1^{(j)}) \mathbf{b}) \right]. \quad (22)$$

Similarly, the generalized observability matrix $\mathcal{O} \in \mathbb{C}^{k \times n}$ associated with the left multi-tuples $\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \dots, \boldsymbol{\mu}^{(\bar{k})}$ is given by

$$\mathcal{O} = \left[(\mathcal{O}^{(1)})^T, (\mathcal{O}^{(2)})^T, \dots, (\mathcal{O}^{(\bar{k})})^T \right]^T \in \mathbb{C}^{k \times n}, \quad (23)$$

where the matrices $\mathcal{O}^{(j)} \in \mathbb{C}^{3 \times n}$, $j = 1, \dots, \bar{k}$, correspond to the j -th multi-tuple $\boldsymbol{\mu}^{(j)}$ in (20) and are given by

$$\mathcal{O}^{(j)} = \begin{bmatrix} \mathbf{c}^T \boldsymbol{\Phi}(\mu_1^{(j)}) \\ \mathbf{c}^T \boldsymbol{\Phi}(\mu_1^{(j)}) \mathbf{N} \boldsymbol{\Phi}(\mu_2^{(j)}) \\ \mathbf{c}^T \boldsymbol{\Phi}(\mu_1^{(j)}) \mathbf{Q}(\boldsymbol{\Phi}(\lambda_1^{(j)}) \mathbf{b} \otimes \boldsymbol{\Phi}(\mu_3^{(j)})) \end{bmatrix}. \quad (24)$$

Now the Loewner matrix $\mathbb{L}$ and the shifted Loewner matrix $\mathbb{L}_s$ are defined using the generalized controllability (21) and observability (23) matrices as

$$\mathbb{L} = -\mathcal{O} \mathbf{E} \mathcal{R}, \quad \mathbb{L}_s = -\mathcal{O} \mathbf{A} \mathcal{R}. \quad (25)$$

The fact that the Loewner matrices are factorized in terms of the pairs of matrices $(\mathbf{E}, \mathbf{A})$ and $(\mathcal{O}, \mathcal{R})$ is an inherent property of the Loewner framework which holds true for both the bilinear and quadratic-bilinear extensions of the method.

As in Sect. 2.1 we focus in the case where a redundant amount of data is provided. We use the SVD to extract the relevant data. Consider the (short) SVDs (10) of the Loewner matrices (25). The matrices $\mathbf{Y}, \mathbf{X} \in \mathbb{C}^{k \times r}$ are obtained by selecting the first r columns of the matrices $\mathbf{Y}_1$ and $\mathbf{X}_2$. We define

$$\mathbf{V} = \mathcal{R} \mathbf{X} \in \mathbb{C}^{n \times r}, \quad \mathbf{W} = \mathcal{O}^* \mathbf{Y} \in \mathbb{C}^{n \times r}. \quad (26)$$

The Loewner ROM is

$$\widehat{\mathbf{E}} = -\mathbf{Y}^* \mathbf{L} \mathbf{X} = \mathbf{W}^* \mathbf{E} \mathbf{V}, \quad \widehat{\mathbf{A}} = -\mathbf{Y}^* \mathbb{L}_s \mathbf{X}^* = \mathbf{W}^* \mathbf{E} \mathbf{V}, \quad (27a)$$

$$\widehat{\mathbf{Q}} = \mathbf{W}^* \mathbf{O}(\mathbf{V} \otimes \mathbf{V}), \quad \widehat{\mathbf{N}} = \mathbf{W}^* \mathbf{N} \mathbf{V}, \quad (27b)$$

$$\widehat{\mathbf{b}} = \mathbf{W}^* \mathbf{b}, \quad \widehat{\mathbf{c}} = \mathbf{V}^* \mathbf{c}. \quad (27c)$$

As in the linear case the projection matrices $\mathbf{V}$, $\mathbf{W}$ and the ROM (17) are complex, but we can obtain real projection matrices $\mathbf{V}$, $\mathbf{W}$ and corresponding ROMs if the sets of left and of right interpolation points contain also the conjugate complex data. See [3, p. 360].

The definition of the Kronecker product and (5b) imply

$$\widehat{\mathbf{Q}}(\widehat{\mathbf{x}} \otimes \widehat{\mathbf{z}}) = \mathbf{W}^* \mathbf{O}(\mathbf{V} \otimes \mathbf{V})(\widehat{\mathbf{x}} \otimes \widehat{\mathbf{z}}) = \mathbf{W}^* \mathbf{O}(\mathbf{V} \widehat{\mathbf{x}} \otimes \mathbf{V} \widehat{\mathbf{z}}) = \mathbf{W}^* \mathbf{G}(\mathbf{V} \widehat{\mathbf{x}}, \mathbf{V} \widehat{\mathbf{z}}).$$

Since the nonlinearity is bilinear, the small $r \times r^2$ matrix $\widehat{\mathbf{Q}}$ can be precomputed for efficient evaluations of the ROM.

We have introduced the Loewner ROM (27) as a Petrov-Galerkin projection. This obviously requires access to the original system (6) matrices $\mathbf{E}$, $\mathbf{A}$, etc. Just as in the linear case, it is possible to generate the same Loewner ROM directly from measurements of the generalized transfer functions. See [8, Sect. 3.4] for details.

3 Implementation of the Loewner Framework

So far we have used the Kronecker product representation (5b) of the quadratic term. As we have mentioned before, the Kronecker product is always possible but not convenient from a practical point of view. In this section we show how computations involving expressions of the form $\mathbf{Q} \mathbf{x} \otimes \mathbf{z}$ can actually be implemented using $\mathbf{G}$. We focus on computations that arise in the Loewner framework, but similar operations also arise in other ROM approaches, see, e.g., [4, 9]. Thus the following discussions are also relevant for those ROM approaches.

The partial Jacobians of the bilinear map $(\mathbf{x}, \mathbf{z}) \mapsto \mathbf{G}(\mathbf{x}, \mathbf{z})$ are $\mathbf{G}_1(\mathbf{z}) = D_{\mathbf{x}} \mathbf{G}(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{n \times n}$ and $\mathbf{G}_2(\mathbf{x}) = D_{\mathbf{z}} \mathbf{G}(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{n \times n}$. Because $\mathbf{G}$ is bilinear

$$\mathbf{G}(\mathbf{x}, \mathbf{z}) = \mathbf{G}_1(\mathbf{z})\mathbf{x} = \mathbf{G}_2(\mathbf{x})\mathbf{z} \quad \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^n. \quad (28)$$

Recall the definition (12) of $\Phi(s) = (s\mathbf{E} - \mathbf{A})^{-1}$ and that $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$. Given scalars $\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3 \in \mathbb{C}$, the Loewner framework requires evaluation of quantities like

$$\Phi(\lambda_1) \mathbf{Q} (\Phi(\lambda_2)\mathbf{b} \otimes \Phi(\lambda_3)\mathbf{b}) \in \mathbb{C}^n, \quad (29a)$$

$$\mathbf{c}^T \Phi(\mu_1) \mathbf{Q} (\Phi(\mu_2)\mathbf{b} \otimes \Phi(\mu_3)\mathbf{b}) \in \mathbb{C}^{1 \times n}, \quad (29b)$$

see (22) and (23). Next we discuss how to express these quantities in terms of $\mathbf{G}$. We consider general scalars $\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3$. In (22) and (23) some of them are related which can be exploited to reduce some of the computations below in a straight forward way.

The vector (29a) is

$$\Phi(\lambda_1) \mathbf{Q} (\Phi(\lambda_2)\mathbf{b} \otimes \Phi(\lambda_3)\mathbf{b}) = \Phi(\lambda_1) \mathbf{G}(\Phi(\lambda_2)\mathbf{b}, \Phi(\lambda_3)\mathbf{b})$$

and is evaluated as follows.

1. Solve $(\lambda_2\mathbf{E} - \mathbf{A})\mathbf{z}_2 = \mathbf{b}$ to get $\mathbf{z}_2 = \Phi(\lambda_2)\mathbf{b} \in \mathbb{C}^n$.
2. Solve $(\lambda_3\mathbf{E} - \mathbf{A})\mathbf{z}_3 = \mathbf{b}$ to get $\mathbf{z}_3 = \Phi(\lambda_3)\mathbf{b} \in \mathbb{C}^n$.
3. Evaluate $\mathbf{y} = \mathbf{G}(\mathbf{z}_2, \mathbf{z}_3) \in \mathbb{C}^n$.
4. Solve $(\lambda_1\mathbf{E} - \mathbf{A})\mathbf{x} = \mathbf{y}$ to get $\mathbf{x} = \Phi(\lambda_1) \mathbf{G}(\Phi(\lambda_2)\mathbf{b}, \Phi(\lambda_3)\mathbf{b})$.

The vector (29b) can be expressed in terms of $\mathbf{G}$ using (28). If $\mathbf{e}_j$ denotes the j -th unit vector, the j -th entry of the vector (29b) is

$$\begin{aligned} \mathbf{c}^T \Phi(\mu_1) \mathbf{Q} (\Phi(\mu_2)\mathbf{b} \otimes \Phi(\mu_3)\mathbf{e}_j) &= \mathbf{c}^T \Phi(\mu_1) \mathbf{Q} (\Phi(\mu_2)\mathbf{b} \otimes \Phi(\mu_3)\mathbf{e}_j) \\ &= \mathbf{c}^T \Phi(\mu_1) \mathbf{G}(\Phi(\mu_2)\mathbf{b}, \Phi(\mu_3)\mathbf{e}_j) = \mathbf{c}^T \Phi(\mu_1) \mathbf{G}_2(\Phi(\mu_2)\mathbf{b}) \Phi(\mu_3)\mathbf{e}_j \end{aligned}$$

so that

$$\mathbf{c}^T \Phi(\mu_1) \mathbf{Q} (\Phi(\mu_2)\mathbf{b} \otimes \Phi(\mu_3)) = \mathbf{c}^T \Phi(\mu_1) \mathbf{G}_2(\Phi(\mu_2)\mathbf{b}) \Phi(\mu_3). \quad (30)$$

Computationally, (30) is evaluated as follows.

1. Solve $(\mu_1\mathbf{E} - \mathbf{A})^T \mathbf{z} = \mathbf{c}$ to get $\mathbf{z}^T = \mathbf{c}^T \Phi(\mu_1) \in \mathbb{C}^{1 \times n}$.
2. Solve $(\mu_2\mathbf{E} - \mathbf{A})\mathbf{w} = \mathbf{b}$ to get $\mathbf{w} = \Phi(\mu_2)\mathbf{b} \in \mathbb{C}^{n \times 1}$.
3. Evaluate $\mathbf{y}^T = \mathbf{z}^T \mathbf{G}_2(\mathbf{w}) \in \mathbb{C}^{1 \times n}$.
4. Solve $(\mu_3\mathbf{E} - \mathbf{A})^T \mathbf{x} = \mathbf{y}$ to get $\mathbf{x}^T = \mathbf{c}^T \Phi(\mu_1) \mathbf{G}_2(\Phi(\mu_2)\mathbf{b}) \Phi(\mu_3) \in \mathbb{C}^{1 \times n}$.

4 Application to Burgers' Equation

4.1 Burgers' Equation

As an example we consider Burgers' equation with Robin boundary conditions. Given viscosity $\nu > 0$ and parameters $\sigma_0 \leq 0, \sigma_1 \geq 0$, consider

$$\frac{\partial}{\partial t} v(x, t) - \nu \frac{\partial^2}{\partial x^2} v(x, t) + v(x, t) \frac{\partial}{\partial x} v(x, t) = 0, \quad x \in (0, 1), t \in (0, T), \quad (31a)$$

$$\nu \frac{\partial}{\partial x} v(x, 0) + \sigma_0 v(0, t) = u_0(t), \quad t \in (0, T), \quad (31b)$$

$$\nu \frac{\partial}{\partial x} v(x, 1) + \sigma_1 v(1, t) = u_1(t), \quad t \in (0, T), \quad (31c)$$

$$v(x, 0) = 0, \quad x \in (0, 1). \quad (31d)$$

The functions u_0, u_1 are the inputs. Since we have restricted our presentation to the case of one input, we will later set $u_1 \equiv 0$ and consider u_0 as the only input. Our output is

$$y(t) = \int_0^1 v(x, t) dx.$$

We seek weak solutions of (31) in the function space

$$W(0, T) = \left\{ \phi \in L^2(0, T; H^1(0, 1)) : \frac{\partial}{\partial t} \phi \in L^2(0, T; (H^1(0, 1))') \right\},$$

where $(H^1(0, 1))'$ is the dual of $H^1(0, 1)$. The function $v \in W(0, T)$ with $v(x, 0) = 0$ a.e. is a weak solution of (31) if

$$\begin{aligned} & \frac{d}{dt} \int_0^1 v(x, t) \varphi(x) dx + \nu \int_0^1 \frac{\partial}{\partial x} v(x, t) \frac{\partial}{\partial x} \varphi(x) dx \\ & - \sigma_0 v(0, t) \varphi(0) + \sigma_1 v(1, t) \varphi(1) + \int_0^1 \frac{\partial}{\partial x} v(x, t) v(x, t) \varphi(x) dx \\ & = u_0(t) \varphi(0) - u_1(t) \varphi(1), \quad \text{for all } \varphi \in H^1(0, 1) \text{ and a.a. } t \in (0, T). \end{aligned} \quad (32)$$

An existence and uniqueness proof for (31) is given, e.g., by Volkwein [17]. In particular, [17, Theorem 2.3] proves that for every $u_0, u_1 \in L^2(0, T)$ there exists a unique solution $v \in W(0, T) \cap L^\infty((0, 1) \times (0, T))$ that obeys

$$\|v\|_{W(0, T)} + \|v\|_{L^\infty} \leq C(1 + \|u_0\|_{L^2(0, T)} + \|u_1\|_{L^2(0, T)}) \quad (33)$$

for a constant $C > 0$ that depends on $T, \nu > 0$, but not on u_0, u_1 .

Since the Loewner framework depends only on transfer function information, it can in theory be applied directly in a function space setting. However, transfer function information is only analytically available for special linear examples. Therefore, we use a fixed finite element semi-discretization to generate transfer function information numerically. Dependence of the Loewner ROM on the mesh size is part of future research.

We discretize (32) in space using linear finite elements on a uniform grid $x_i = ih, i = 0, \dots, n-1, h = 1/(n-1)$. The weak solution of Burgers' equation (32) is approximated by $v_h(x, t) = \sum_{j=0}^{n-1} v_j(t) \varphi_j(x)$, where $\varphi_i \in H^1(0, 1), i = 0, \dots, n-1$, are the usual piecewise linear 'hat' functions. We set $u_1 \equiv 0$ and consider u_0 as the only input to arrive at a system (2) with $\mathbf{N} = \mathbf{0}$, and $\mathbf{D} = \mathbf{0}$. For given

vectors $\mathbf{v} = (v_0, \dots, v_{n-1})^T$, $\mathbf{z} = (z_0, \dots, z_{n-1})^T$, the i -th component of the bilinear map $\mathbf{G}$ is

$$\mathbf{G}_i(\mathbf{v}, \mathbf{z}) = - \int_0^1 \left(\frac{d}{dx} \sum_{j=0}^{n-1} v_j \varphi_j(x) \right) \left(\sum_{l=0}^{n-1} z_l \varphi_l(x) \right) \varphi_i(x) dx.$$

4.2 Numerical Results

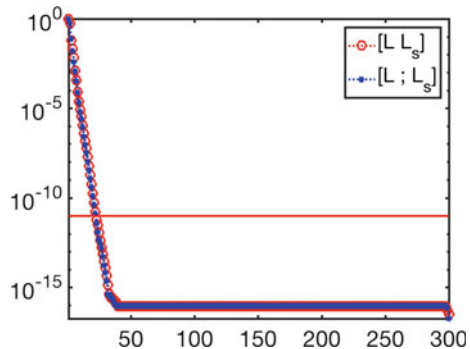
We use the problem data $\nu = 0.01$, $\sigma_0 = 0$, $\sigma_1 = 0.1$. The FOM is the linear finite element semi-discretization with $n = 257$. The semi-discretized system (2) is approximately solved using backward Euler over varying time intervals $[0, T]$ specified below with time step size $\Delta t = 1/128$. In all simulations we use the input $u_0(t) = 0.1 \sin(4\pi t)$ and $u_1 \equiv 0$.

The interpolation points to construct the Loewner ROM are chosen as follows. First we create 300 logarithmically spaced points ξ_j , $j = 1, \dots, 300$, between 1 and 10^3 (in Matlab `logspace(0, 3, 300)`), and then we select the left interpolation points $\mu_{2j-1} = \xi_{2j-1}i$, $\mu_{2j} = -\xi_{2j-1}i$, $j = 1, \dots, 150$, and the right interpolation points $\lambda_{2j-1} = \xi_{2j}i$, $\lambda_{2j} = -\xi_{2j}i$, $j = 1, \dots, 150$. The choice of interpolation points clearly has an impact on the quality of the ROM approximation and how to choose ‘good’ interpolation points is still an open question. Thus, the above choice is somewhat arbitrary.

The singular values of the matrices in (25) are shown in Fig. 1. Let σ_j denote the singular values of $[\mathbb{L} \mathbb{L}_s]$. The size r of the ROM is chosen to the smallest r with $\sigma_r/\sigma_1 > 10^{-11}$. This leads to a Loewner ROM of size $r = 22$.

The outputs of the FOM and of the Loewner ROM are shown in Fig. 2. For approximately $t \leq 1.5$ the agreement between the FOM and the Loewner ROM output is good, but there are larger differences between both outputs for approximately $t > 1.5$. Moreover, the Loewner ROM exhibits instabilities starting around $t = 1.5$,

Fig. 1 The normalized singular values of the Loewner matrices (25). The size $r = 22$ of the ROM is chosen to the smallest r with $\sigma_r/\sigma_1 > 10^{-11}$



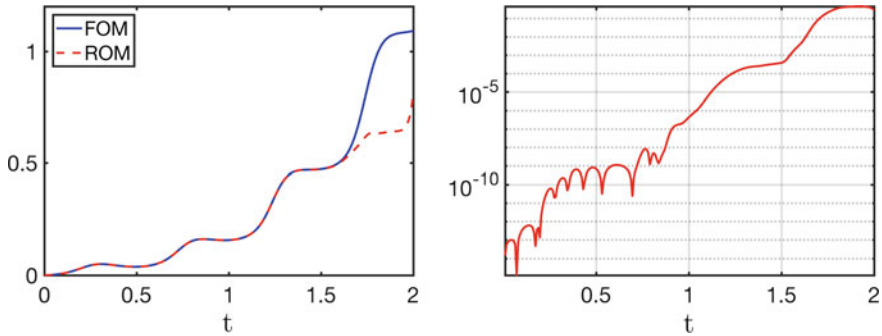


Fig. 2 Left plot: output of the FOM (solid blue line) and of the Loewner ROM with Petrov-Galerkin projection matrices $\mathbf{V} \neq \mathbf{W}$ (dashed red line). Right plot: error between outputs of the FOM and the Loewner ROM

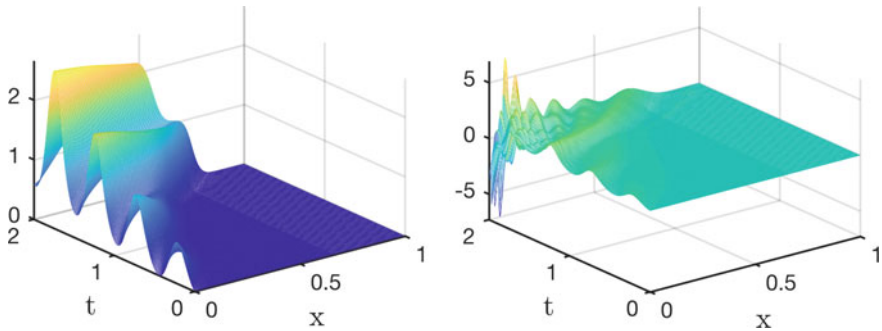


Fig. 3 Solution of the FOM (left) and of the Loewner ROM with Petrov-Galerkin projection matrices $\mathbf{V} \neq \mathbf{W}$ (right). The Loewner ROM exhibits instabilities

as can be seen from the states $\mathbf{x}$ generated by the FOM and the state $\mathbf{V}\hat{\mathbf{x}}$ generated by the Loewner ROM, which are shown in Fig. 3.

Stability results for the Burgers' equation like (33) are based on the weak form (32) and can be carried over to Galerkin approximations of (32), such as the finite element discretization or Galerkin projection based ROMs with $\mathbf{V} = \mathbf{W}$. In the standard Loewner approach the projection matrices $\mathbf{V} \neq \mathbf{W}$, and it is not clear in the general case how to construct a Loewner ROM with $\mathbf{V} = \mathbf{W}$. We merge the Loewner projection matrices $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ into one larger matrix $[\mathbf{V}, \mathbf{W}] \in \mathbb{R}^{n \times 2r}$ (we actually compute an orthonormal basis of the columns of $[\mathbf{V}, \mathbf{W}]$ to ensure that the resulting matrix is full rank), and we use a Galerkin projection with this matrix $[\mathbf{V}, \mathbf{W}] \in \mathbb{R}^{n \times 2r}$. We refer to the resulting ROM as a Loewner Galerkin ROM.

The outputs of the FOM and of the Loewner Galerkin ROM are shown in Fig. 4. We can now simulate the Loewner Galerkin ROM at least until $T = 6$ and there is good agreement between the outputs of the FOM and of the Loewner Galerkin ROM. The states $\mathbf{x}$ and $\mathbf{V}\hat{\mathbf{x}}$ generated by the FOM and the Loewner Galerkin ROM

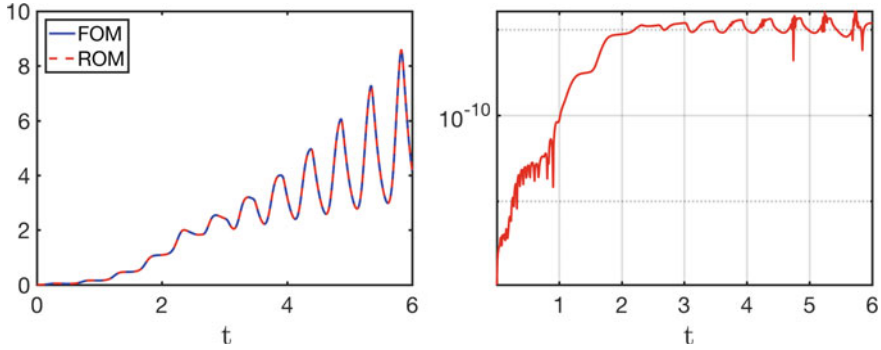


Fig. 4 Left plot: output of the FOM (solid blue line) and of the Loewner ROM with Galerkin projection matrix $[\mathbf{V}, \mathbf{W}]$ (dashed red line). Right plot: error between outputs of the FOM and the Loewner ROM

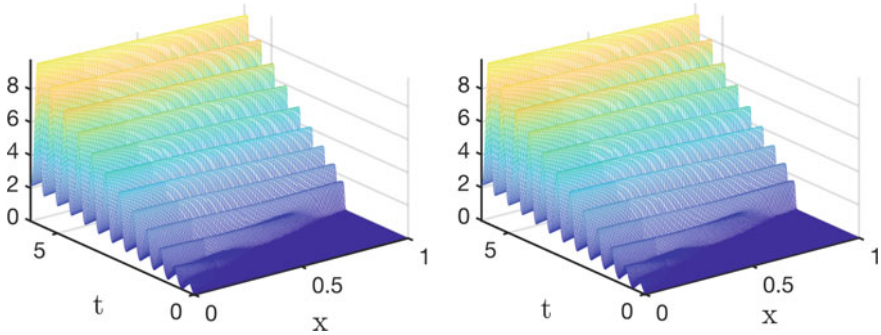


Fig. 5 Solution of the FOM (left) and of the Loewner ROM with Galerkin projection matrix $[\mathbf{V}, \mathbf{W}]$ (right)

are shown in Fig. 5. The FOM solution in Fig. 5 restricted to the time interval $[0, 2]$ is identical to the FOM solution shown in Fig. 3.

Of course, since our Loewner Galerkin ROM is twice the size of the standard Loewner Petrov-Galerkin ROM, it is not entirely clear whether this improvement in results is due to the increased ROM size, or the switch from a Petrov-Galerkin projection to a Galerkin projection. We did change the accuracy and corresponding ROM size r of the standard Loewner Petrov-Galerkin ROM slightly and still observed instabilities in the resulting ROMs. Thus it seems more accurate Loewner Petrov-Galerkin ROMs alone do not restore stability, but this issue is still under investigation.

In our current implementation of the Loewner approach, we generate $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ and compute the ROM explicitly as a Petrov-Galerkin projection ROM (27). As mentioned at the end of Sect. 2.2, the same Loewner ROM can be computed directly from measurements of the generalized transfer functions. In this case our approach to enforce stability via the use of Loewner Galerkin ROM with $[\mathbf{V}, \mathbf{W}] \in \mathbb{R}^{n \times 2r}$

is no longer possible. Extension of the Loewner approach to generate ROM that correspond to a Galerkin projection ROM is an interesting research question.

5 Conclusions and Future Work

We have presented an extension of the Loewner framework to compute ROMs of quadratic-bilinear systems. Specifically, we have used the Kronecker product representation of quadratic-bilinear systems to present the algorithm, but then used the bilinear maps that naturally arise in semi-discretizations of fluid flow problems, such as Burgers' equation or the Navier-Stokes equations to express the actual computations. This makes it possible to apply the Loewner framework to large-scale problems. In this paper we have applied to the viscous Burgers' equation. Application to the Navier-Stokes equations is ongoing work.

The application to Burgers' equation showed the potential of the Loewner framework, but also raises some questions that still need to be addressed. Generally, the selection of interpolation points (the μ_j 's and λ_j 's) is an issue. Current numerical experiments indicate that the more interpolation points can be used, the better given a constant ROM size r . Recall that the data gets assembled in the Loewner and shifted Loewner matrices and then is compressed via the SVD. Thus more data does not necessarily mean larger ROMs. We consider SISO systems. The extension to multiple input and multiple output systems is possible, using so-called tangential interpolation. For linear systems this is described in the tutorial paper [3].

An important issue is stability. Our numerics have shown that the standard Loewner ROM may not be stable. Currently, our Loewner ROM is equivalent to a Petrov-Galerkin projection, $\mathbf{W} \neq \mathbf{V}$. At the same time, stability results like (33) for Burgers' equation are based on the weak form and Galerkin projection. Thus if we can modify the Loewner framework to enforce $\mathbf{W} = \mathbf{V}$, then the resulting ROM inherits the stability properties of the underlying original system. In the linear case stability issues can be treated by postprocessing, see, e.g., Gosea and Antoulas [7]. If we explicitly compute $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times r}$ we can enforce stability via the use of Loewner Galerkin ROM with $[\mathbf{V}, \mathbf{W}] \in \mathbb{R}^{n \times 2r}$ as demonstrated in Sect. 4.2. However, this is not possible if the Loewner ROM is computed directly from data.

Finally, the Loewner framework starts from system representations in frequency domain and is based on measurements in frequency domain. This is inconvenient for many applications where only time domain measurements or simulations are accessible. Initial work towards time-domain Loewner ROMs is presented by Peherstorfer et al. [12].

Acknowledgements The authors gratefully acknowledge support by NSF grants CNS-1701292 and DMS-1522798. We thank the two referees for their comments, which have lead to improvements in the presentation.

References

1. Antoulas, A.C.: Approximation of large-scale dynamical systems. In: *Advances in Design and Control*, vol. 6. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2005). <https://doi.org/10.1137/1.9780898718713>
2. Antoulas, A.C., Gosea, I.V., Ionita, A.C.: Model reduction of bilinear systems in the Loewner framework. *SIAM J. Sci. Comput.* **38**(5), B889–B916 (2016). <https://doi.org/10.1137/15M1041432>
3. Antoulas, A.C., Lefteriu, S., Ionita, A.C.: Chapter 8: A tutorial introduction to the Loewner framework for model reduction. In: P. Benner, A. Cohen, M. Ohlberger, K. Willcox (eds.) *Model Reduction and Approximation: Theory and Algorithms*, pp. 335–376. SIAM, Philadelphia (2017). <https://doi.org/10.1137/1.9781611974829.ch8>
4. Benner, P., Breiten, T.: Two-sided projection methods for nonlinear model order reduction. *SIAM J. Sci. Comput.* **37**(2), B239–B260 (2015). <https://doi.org/10.1137/14097255X>
5. Breiten, T., Damm, T.: Krylov subspace methods for model order reduction of bilinear control systems. *Syst. Control Lett.* **59**(8), 443–450 (2010). <https://doi.org/10.1016/j.sysconle.2010.06.003>
6. Elman, H.C., Silvester, D.J., Wathen, A.J.: *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, 2nd edn. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford (2014). <https://doi.org/10.1093/acprof:oso/9780199678792.001.0001>
7. Gosea, I.V., Antoulas, A.C.: Stability preserving post-processing methods applied in the Loewner framework. In: *IEEE 20th Workshop on Signal and Power Integrity (SPI)*, pp. 1–4 (2016). <https://doi.org/10.1109/SaPIW.2016.7496283>
8. Gosea, I.V., Antoulas, A.C.: Data-driven model order reduction of quadratic-bilinear systems. *Numer. Linear Algebra Appl.* (2018). Under review
9. Gu, C.: QLMOR: a projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **30**(9), 1307–1320 (2011). <https://doi.org/10.1109/TCAD.2011.2142184>
10. Hesthaven, J.S., Rozza, G., Stamm, B.: *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer Briefs in Mathematics. Springer, New York (2015). <https://doi.org/10.1007/978-3-319-22470-1>
11. Layton, W.: Introduction to the numerical analysis of incompressible viscous flows. *Computational Science and Engineering*, vol. 6. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2008). <https://doi.org/10.1137/1.9780898718904>
12. Peherstorfer, B., Gugercin, S., Willcox, K.: Data-driven reduced model construction with time-domain loewner models. *SIAM J. Sci. Comput.* **39**(5), A2152–A2178 (2017). <https://doi.org/10.1137/16M1094750>
13. Quarteroni, A., Manzoni, A., Negri, F.: *Reduced Basis Methods for Partial Differential Equations. An Introduction*. Unitext, vol. 92. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-15431-2>
14. Rowley, C.W., Dawson, S.T.M.: Model reduction for flow analysis and control. *Ann. Rev. Fluid Mech.* **49**(1), 387–417 (2017). <https://doi.org/10.1146/annurev-fluid-010816-060042>
15. Rozza, G., Huynh, D.B.P., Patera, A.T.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Methods Eng.* **15**(3), 229–275 (2008). <https://doi.org/10.1007/s11831-008-9019-9>
16. Rugh, W.J.: *Nonlinear System Theory. The Volterra/Wiener Approach*. Johns Hopkins University Press, Baltimore, Md. (1981). <https://sites.google.com/site/wilsonjrugh>. Accessed 22 Feb 2018
17. Volkwein, S.: Second order conditions for boundary control problems of the Burgers equation. *Control Cybernet* **30**(3), 249–278 (2001). http://www.oxygene.ibspan.waw.pl:3000/contents/export?filename=2001-3-02_volkwein.pdf. Accessed 22 Feb 2018