



Towards Data-Driven Model Reduction of the Navier-Stokes Equations Using the Loewner Framework

Alejandro N. Diaz and Matthias Heinkenschloss^(✉)

Department of Computational and Applied Mathematics, MS-134, Rice University,
6100 Main Street, Houston, TX 77005-1892, USA
`{and5,heinken}@rice.edu`

Abstract. The Loewner framework is extended to compute reduced order models (ROMs) for systems governed by the incompressible Navier-Stokes (NS) equations. For quadratic ordinary differential equations (ODEs) it constructs a ROM directly from measurements of transfer function components derived from an expansion of the system's input-to-output map. Given measurements, no explicit access to the system is required to construct the ROM.

To extend the Loewner framework, the NS equations are transformed into ODEs by projecting onto the subspace defined by the incompressibility condition. This projection is used theoretically, but avoided computationally. This paper presents the overall approach. Currently, transfer function measurements are obtained via computational simulations; obtaining them from experiments is an open issue. Numerical results show the potential of the Loewner framework, but also reveal possible lack of stability of the ROM. A possible approach, which currently requires access to the NS system, to deal with these instabilities is outlined.

Keywords: Model reduction · Loewner framework · Navier-Stokes equations · Data-driven

1 Introduction

This paper extends the data-driven Loewner framework to construct a reduced order model (ROM) for systems governed by the semi-discretized incompressible Navier-Stokes (NS) equations. These computationally inexpensive ROMs are useful in applications that require many queries of the system, such as optimal design, optimal control, or uncertainty quantification, which would be prohibitively expensive with the original, high dimensional, computationally expensive full order model (FOM). The Loewner framework constructs a ROM that is given by a Petrov-Galerkin projection of the FOM. However, unlike traditional Petrov-Galerkin projection ROMs, the Loewner ROM is computed directly from measurements of transfer function components associated with the FOM. In

particular, the Loewner framework is non-intrusive. It only requires these transfer function component measurements, but it does not need explicit access to the FOM or explicit application of the Petrov-Galerkin projections. The transfer function components of the Loewner ROM typically well approximate the corresponding transfer function components of the FOM near the frequencies at which measurements were taken to construct the Loewner ROM. Thus, the overall quality of the Loewner ROM depends on the measurements on which it is built. This is the first application of the Loewner framework to the semi-discretized incompressible NS equations, which in the terminology of ROMs is a quadratic semi-explicit differential algebraic equation (DAE) system.

The development of the Loewner framework started with the paper [16]. The tutorial paper [6] and the book [2] provide the state-of-the-art of the Loewner framework. The papers [5, 12], extend the Loewner framework to so-called bilinear and quadratic ordinary differential equation (ODE) systems, and [3] discusses the application of the Loewner framework to systems governed by Burgers' equation. The Loewner framework is extended in [4] to a class of linear semi-explicit DAE systems, which includes the Oseen equations. This paper builds on [3, 4].

To extend the existing Loewner framework to the semi-discretized NS equations we first project these equations onto the subspace defined by the discrete incompressibility constraints to express the semi-discretized NS equations as a quadratic ODE system. This projection has already been used, e.g., in [1, 4, 7, 9, 14]. Then the quadratic ODE system is expanded into a system of infinitely many linear equations. Such expansions are discussed, e.g., in the book [17] and they have been applied to develop ROMs for bilinear or quadratic bilinear systems in, e.g., [1, 3, 5, 8–10, 12, 13]. The papers [3, 5, 8, 10, 12] consider ODE systems, not DAEs. The papers [1, 9] also consider projection based ROM for the NS equations, and the ROMs are also designed to approximate transfer function components of the FOM. However, the transfer function components used in [1, 9] are different from those used here, and the ROM approaches in [1, 9] require explicit access to components of the NS equations and are intrusive.

This paper describes the extension of the Loewner framework to a class of semi-discretized incompressible NS systems, and numerically explores potential stability issues of the resulting Loewner ROM and possible modifications of this ROM to avoid them.

2 Loewner Framework for the Navier-Stokes Equations

We state the semi-discretized NS system, and its projection onto the subspace defined by the discrete incompressibility conditions. This projection transforms the Navier-Stokes system into a quadratic ODE system. We then review the Loewner framework for this projected ODE system.

2.1 Navier-Stokes System

Let $\mathbf{v} : (0, T) \rightarrow \mathbb{R}^{n_v}$ and $\mathbf{p} : (0, T) \rightarrow \mathbb{R}^{n_p}$ be the semi-discretized velocities and pressures, respectively. Furthermore, let $\mathbf{E}_{11} \in \mathbb{R}^{n_v \times n_v}$ be symmetric positive

definite, $\mathbf{A}_{11} \in \mathbb{R}^{n_v \times n_v}$, let $\mathbf{A}_{12}^T \in \mathbb{R}^{n_p \times n_v}$ be of rank $n_p < n_v$, $\mathbf{b} \in \mathbb{R}^{n_v}$, $\mathbf{Q} \in \mathbb{R}^{n_v \times n_v^2}$, and $\mathbf{c} \in \mathbb{R}^{n_v}$. We consider the following system with input $\mathbf{g} : (0, T) \rightarrow \mathbb{R}$ and output $\mathbf{y} : (0, T) \rightarrow \mathbb{R}$,

$$\mathbf{E}_{11} \frac{d}{dt} \mathbf{v}(t) = \mathbf{A}_{11} \mathbf{v}(t) + \mathbf{Q}(\mathbf{v}(t) \otimes \mathbf{v}(t)) + \mathbf{A}_{12} \mathbf{p}(t) + \mathbf{b} \mathbf{g}(t), \quad t \in (0, T), \quad (1a)$$

$$\mathbf{0} = \mathbf{A}_{12}^T \mathbf{v}(t), \quad t \in (0, T), \quad (1b)$$

with homogeneous initial condition $\mathbf{v}(0) = \mathbf{0}$, and with the output equation

$$\mathbf{y}(t) = \mathbf{c}^T \mathbf{v}(t), \quad t \in (0, T). \quad (1c)$$

Semi-discretization of the Navier-Stokes equations leads to (1), but the Loewner framework presented in this paper can, of course, be applied to any system governed by DAEs (1). Because of page limitations we only consider single-input-single-output (SISO) systems (1). We will elaborate elsewhere on how to generalize the approach to multiple-input-multiple-output (MIMO) systems by interpolating transfer function components along tangential directions.

2.2 Transformation into Quadratic ODE System

Next, we project (1) onto the subspace defined by the discrete incompressibility conditions (1b) to write (1) as an ODE system. As in [14], define the projection

$$\mathbf{\Pi} = \mathbf{I} - \mathbf{A}_{12}(\mathbf{A}_{12}^T \mathbf{E}_{11}^{-1} \mathbf{A}_{12})^{-1} \mathbf{A}_{12}^T \mathbf{E}_{11}^{-1}.$$

It can be verified that $\mathbf{\Pi}^2 = \mathbf{\Pi}$, $\mathbf{\Pi} \mathbf{E}_{11} = \mathbf{E}_{11} \mathbf{\Pi}^T$, $\text{null}(\mathbf{\Pi}) = \text{range}(\mathbf{A}_{12})$, and $\text{range}(\mathbf{\Pi}) = \text{null}(\mathbf{A}_{12}^T \mathbf{E}_{11}^{-1})$, i.e. $\mathbf{\Pi}$ is an $\mathbf{E}_{11}$ -orthogonal projection. The properties of $\mathbf{\Pi}$ imply that

$$\mathbf{A}_{12}^T \mathbf{v}(t) = \mathbf{0} \quad \text{if and only if} \quad \mathbf{\Pi}^T \mathbf{v}(t) = \mathbf{v}(t). \quad (2)$$

Next we express $\mathbf{p}$ in terms of $\mathbf{v}$ and project onto the constraint (1b). Specifically, we premultiply (1a) by $\mathbf{A}_{12}^T \mathbf{E}_{11}^{-1}$, then use (1b) and solve the resulting equation for $\mathbf{p}$ to obtain

$$\mathbf{p}(t) = -(\mathbf{A}_{12}^T \mathbf{E}_{11}^{-1} \mathbf{A}_{12})^{-1} \mathbf{A}_{12}^T \mathbf{E}_{11}^{-1} (\mathbf{A}_{11} \mathbf{v}(t) + \mathbf{Q}(\mathbf{v}(t) \otimes \mathbf{v}(t)) + \mathbf{b} \mathbf{g}(t)). \quad (3)$$

Now insert (3) into (1), apply (2), and use $\mathbf{\Pi} \mathbf{A}_{12}(\mathbf{A}_{12}^T \mathbf{E}_{11}^{-1} \mathbf{A}_{12})^{-1} = \mathbf{0}$ to write (1) as

$$\mathbf{\Pi} \mathbf{E}_{11} \mathbf{\Pi}^T \frac{d}{dt} \mathbf{v}(t) = \mathbf{\Pi} \mathbf{A}_{11} \mathbf{\Pi}^T \mathbf{v}(t) + \mathbf{\Pi} \mathbf{Q}(\mathbf{\Pi}^T \mathbf{v}(t) \otimes \mathbf{\Pi}^T \mathbf{v}(t)) + \mathbf{\Pi} \mathbf{b} \mathbf{g}(t), \quad (4a)$$

$$\mathbf{y}(t) = \mathbf{c}^T \mathbf{\Pi}^T \mathbf{v}(t), \quad (4b)$$

with initial condition $\mathbf{\Pi}^T \mathbf{v}(0) = \mathbf{0}$. This is a dynamical system in the $n_v - n_p$ dimensional subspace $\text{null}(\mathbf{\Pi})$ and (4a,b) has to be solved for $\mathbf{\Pi}^T \mathbf{v} = \mathbf{v}$.

As in [14], this is made explicit by decomposing $\boldsymbol{\Pi} = \boldsymbol{\Theta}_l \boldsymbol{\Theta}_r^T$ with $\boldsymbol{\Theta}_l, \boldsymbol{\Theta}_r \in \mathbb{R}^{n_v \times (n_v - n_p)}$ satisfying $\boldsymbol{\Theta}_l^T \boldsymbol{\Theta}_r = \mathbf{I}$. Substituting this decomposition into (4) and using the Kronecker product property

$$\mathbf{H}\mathbf{X} \otimes \mathbf{K}\mathbf{Z} = (\mathbf{H} \otimes \mathbf{K})(\mathbf{X} \otimes \mathbf{Z}) \quad (5)$$

shows that $\tilde{\mathbf{v}} = \boldsymbol{\Theta}_l^T \mathbf{v} \in \mathbb{R}^{n_v - n_p}$ satisfies

$$\tilde{\mathbf{E}} \frac{d}{dt} \tilde{\mathbf{v}}(t) = \tilde{\mathbf{A}} \tilde{\mathbf{v}}(t) + \tilde{\mathbf{Q}}(\tilde{\mathbf{v}}(t) \otimes \tilde{\mathbf{v}}(t)) + \tilde{\mathbf{b}} \mathbf{g}(t), \quad t \in (0, T), \quad (6a)$$

$$\mathbf{y}(t) = \tilde{\mathbf{c}}^T \tilde{\mathbf{v}}(t), \quad t \in (0, T), \quad (6b)$$

with initial conditions $\tilde{\mathbf{v}}(0) = \mathbf{0}$, where

$$\begin{aligned} \tilde{\mathbf{E}} &:= \boldsymbol{\Theta}_r^T \mathbf{E}_{11} \boldsymbol{\Theta}_r \in \mathbb{R}^{(n_v - n_p) \times (n_v - n_p)}, & \tilde{\mathbf{A}} &:= \boldsymbol{\Theta}_r^T \mathbf{A}_{11} \boldsymbol{\Theta}_r \in \mathbb{R}^{(n_v - n_p) \times (n_v - n_p)}, \\ \tilde{\mathbf{b}} &:= \boldsymbol{\Theta}_r^T \mathbf{b} \in \mathbb{R}^{n_v - n_p}, & \tilde{\mathbf{c}} &:= \boldsymbol{\Theta}_r^T \mathbf{c} \in \mathbb{R}^{n_v - n_p}, \\ \tilde{\mathbf{Q}} &:= \boldsymbol{\Theta}_r^T \mathbf{Q}(\boldsymbol{\Theta}_r \otimes \boldsymbol{\Theta}_r) \in \mathbb{R}^{(n_v - n_p) \times (n_v - n_p)^2}. \end{aligned}$$

Since $\tilde{\mathbf{E}}$ is invertible, (6) is an ODE. The DAE system (1) and the ODE system (6) are equivalent. Specifically, the transfer function components of (1) are identical to the transfer function components of (6). The matrices $\boldsymbol{\Theta}_l, \boldsymbol{\Theta}_r \in \mathbb{R}^{n_v \times (n_v - n_p)}$ are expensive to compute, and (6) is a dense large-scale system. Therefore, (6) is used only theoretically. Ultimately, computations are performed using quantities that arise in the original system (1).

2.3 Expansion of Quadratic ODE System into a Linear System

Now we expand the quadratic ODEs (6) into a system of infinitely many linear ODEs. The approach used is the so-called variational approach in [17, Ch. 3].

Let $\tilde{\mathbf{v}}(\mathbf{g}; \cdot)$ denote the solution of (6a,b) with input $\mathbf{g}$. The variational approach computes an expansion of the solution in terms of its derivatives with respect to $\mathbf{g}$. The solution of (6a,b) with input $\alpha \mathbf{g}$, $\alpha \in \mathbb{R}$, is given by

$$\tilde{\mathbf{v}}(\alpha \mathbf{g}; \cdot) = \tilde{\mathbf{v}}(\mathbf{0} + \alpha \mathbf{g}; \cdot) = \tilde{\mathbf{v}}(\mathbf{0}; \cdot) + \sum_{l=1}^{\infty} \alpha^l \tilde{\mathbf{v}}_l = \sum_{l=1}^{\infty} \alpha^l \tilde{\mathbf{v}}_l, \quad (7)$$

where $\tilde{\mathbf{v}}_l$ is the l -th derivative of the solution map $\mathbf{g} \mapsto \tilde{\mathbf{v}}(\mathbf{g}; \cdot)$ at $\mathbf{0}$ evaluated in the direction $\mathbf{g}$, and we have used that the solution with zero input is zero, $\tilde{\mathbf{v}}(\mathbf{0}; \cdot) = \mathbf{0}$. The proof that the series in the right hand side of (7) converges and is equal to $\tilde{\mathbf{v}}(\alpha \mathbf{g}; \cdot)$ is given, e.g., in [17, Appendix 3.1].

Consider (6) with $\mathbf{g}$ replaced by $\alpha \mathbf{g}$ and insert the expansion (7) to arrive at

$$\sum_{l=1}^{\infty} \alpha^l \tilde{\mathbf{E}} \frac{d}{dt} \tilde{\mathbf{v}}_l(t) = \sum_{l=1}^{\infty} \alpha^l \tilde{\mathbf{A}} \tilde{\mathbf{v}}_l(t) + \alpha \tilde{\mathbf{b}} \mathbf{g}(t) + \sum_{m,l=1}^{\infty} \alpha^{l+m} \tilde{\mathbf{Q}}(\tilde{\mathbf{v}}_l(t) \otimes \tilde{\mathbf{v}}_m(t)), \quad (8a)$$

$$\mathbf{y}(t) = \sum_{l=1}^{\infty} \alpha^l \tilde{\mathbf{c}}^T \tilde{\mathbf{v}}_l(t), \quad (8b)$$

and $\sum_{l=1}^{\infty} \alpha^l \tilde{\mathbf{v}}_l(0) = \mathbf{0}$. Equating powers of α gives the infinite system of ODEs

$$\tilde{\mathbf{E}} \frac{d}{dt} \tilde{\mathbf{v}}_1(t) = \tilde{\mathbf{A}} \tilde{\mathbf{v}}_1(t) + \tilde{\mathbf{b}} \mathbf{g}(t), \quad \tilde{\mathbf{v}}_1(0) = \mathbf{0}, \quad (9a)$$

$$\tilde{\mathbf{E}} \frac{d}{dt} \tilde{\mathbf{v}}_l(t) = \tilde{\mathbf{A}} \tilde{\mathbf{v}}_l(t) + \sum_{j=1}^{l-1} \tilde{\mathbf{Q}}(\tilde{\mathbf{v}}_j(t) \otimes \tilde{\mathbf{v}}_{l-j}(t)), \quad \tilde{\mathbf{v}}_l(0) = \mathbf{0}, \quad l \geq 2, \quad (9b)$$

with output equations

$$\mathbf{y}_1(t) = \tilde{\mathbf{c}}^T \tilde{\mathbf{v}}_1(t), \quad \mathbf{y}_l(t) = \tilde{\mathbf{c}}^T \tilde{\mathbf{v}}_l(t), \quad l \geq 2. \quad (9c)$$

Recall that the solution of the ODE

$$\tilde{\mathbf{E}} \frac{d}{dt} \tilde{\mathbf{v}}(t) = \tilde{\mathbf{A}} \tilde{\mathbf{v}}(t) + \mathbf{f}(t), \quad t \in (0, T), \quad \tilde{\mathbf{v}}(0) = \mathbf{0}$$

is given by $\tilde{\mathbf{v}}(t) = \int_0^t e^{\tilde{\mathbf{E}}^{-1} \tilde{\mathbf{A}}(t-\tau)} \tilde{\mathbf{E}}^{-1} \mathbf{f}(\tau) d\tau = \int_0^t e^{\tilde{\mathbf{E}}^{-1} \tilde{\mathbf{A}} \tau} \tilde{\mathbf{E}}^{-1} \mathbf{f}(t-\tau) d\tau = \int_0^t \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} \tau} \mathbf{f}(t-\tau) d\tau$. Applying this to (9a,b) yields

$$\tilde{\mathbf{v}}_1(t) = \int_0^t \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_1} \tilde{\mathbf{b}} \mathbf{g}(t-t_1) dt_1, \quad (10a)$$

$$\tilde{\mathbf{v}}_l(t) = \int_0^t \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_l} \sum_{j=1}^{l-1} \tilde{\mathbf{Q}}(\tilde{\mathbf{v}}_j(t-t_l) \otimes \tilde{\mathbf{v}}_{l-j}(t-t_l)) dt_l, \quad l \geq 2. \quad (10b)$$

Next we use (9c) and (10) to write $\mathbf{y}$ in terms of $\mathbf{g}$. We focus on the first two terms in the series, as the resulting expressions become increasingly complex and tedious to compute. Using (10) and the Kronecker product property (5) gives

$$\begin{aligned} \tilde{\mathbf{v}}_2(t) &= \int_0^t \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_2} \tilde{\mathbf{Q}}(\tilde{\mathbf{v}}_1(t-t_2) \otimes \tilde{\mathbf{v}}_1(t-t_2)) dt_2 \\ &= \int_0^t \int_0^{t-t_2} \int_0^{t-t_2} \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_2} \tilde{\mathbf{Q}}\left(\tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_1} \tilde{\mathbf{b}} \otimes \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_3} \tilde{\mathbf{b}}\right) \\ &\quad \times \left(\mathbf{g}(t-t_1-t_2) \otimes \mathbf{g}(t-t_2-t_3)\right) dt_1 dt_3 dt_2. \end{aligned} \quad (11)$$

Substituting (10a) and (11) into the output equation (8c) yields

$$\mathbf{y}_1(t) = \int_0^t \mathbf{h}_1(t_1) \mathbf{g}(t-t_1) dt_1,$$

$$\mathbf{y}_2(t) = \int_0^t \int_0^{t-t_3} \int_0^{t-t_3} \mathbf{h}_2(t_1, t_2, t_3) (\mathbf{g}(t-t_1-t_2) \otimes \mathbf{g}(t-t_2-t_3)) dt_1 dt_2 dt_3,$$

etc., where

$$\mathbf{h}_1(t_1) = \tilde{\mathbf{c}}^T \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_1} \tilde{\mathbf{b}}, \quad (12a)$$

$$\mathbf{h}_2(t_1, t_2, t_3) = \tilde{\mathbf{c}}^T \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_3} \tilde{\mathbf{Q}}(\tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_1} \tilde{\mathbf{b}} \otimes \tilde{\mathbf{E}}^{-1} e^{\tilde{\mathbf{A}} \tilde{\mathbf{E}}^{-1} t_2} \tilde{\mathbf{b}}), \quad (12b)$$

etc. Lastly we take the (multidimensional) Laplace transforms of the kernel function components (12) to compute the transfer function components

$$\mathbf{H}_1(s_1) = \tilde{\mathbf{c}}^T \tilde{\boldsymbol{\Phi}}(s_1) \tilde{\mathbf{b}}, \quad \mathbf{H}_2(s_1, s_2, s_3) = \tilde{\mathbf{c}}^T \tilde{\boldsymbol{\Phi}}(s_1) \tilde{\mathbf{Q}} \left(\tilde{\boldsymbol{\Phi}}(s_2) \tilde{\mathbf{b}} \otimes \tilde{\boldsymbol{\Phi}}(s_3) \tilde{\mathbf{b}} \right), \quad (13)$$

where $\tilde{\boldsymbol{\Phi}}(s) := (s\tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1}$.

2.4 Transfer Function Interpolation for Quadratic ODE Systems

The Loewner framework is related to interpolation based ROM approaches that construct a ROM whose transfer function components interpolate the FOM transfer function components at desired points. See, e.g., [1, 2, 8, 10, 12, 13]. A review of these interpolation based ROM approaches is useful to understand the Loewner framework. Different transfer function components are interpolated in [1, 8, 10, 12, 13]. We follow [12].

We truncate the expansion (8) after $l = 2$ terms, and we seek a ROM so that the FOM transfer function components (13) that correspond to this truncation are interpolated at selected points by the corresponding ROM transfer function components. These interpolation conditions are derived for linear systems, e.g., in [2, Sect. 3.3], and for quadratic bilinear systems in [12]. They require a particular grouping of the frequencies at which the transfer functions are interpolated. Some benefits of this grouping are only reaped in the MIMO case (not considered in this paper) in which the transfer functions (13) are matrix valued, and interpolation conditions hold for the matrices multiplied from the left or from the right by so-called tangential directions, but not for the entire matrix valued transfer functions. To be consistent with the literature, e.g., [2, 12], we apply these groupings even though we only consider the SISO case.

We assume that the total number of frequencies is a multiple of four, $4\bar{k}$, and we split the interpolation points into two disjoint sets $\mathcal{S}_\mu \subset \mathbb{C}$ and $\mathcal{S}_\lambda \subset \mathbb{C}$, each containing $k = 2\bar{k}$ points. This split could be avoided in the SISO case, but is needed in the MIMO case where the frequencies in $\mathcal{S}_\mu$ and $\mathcal{S}_\lambda$ are associated with the left and the right tangential directions respectively. Within $\mathcal{S}_\mu$ and $\mathcal{S}_\lambda$ we arrange the interpolation points as

$$\mathcal{S}_\mu = \cup_{1 \leq j \leq \bar{k}} \{\mu_1^{(j)}, \mu_2^{(j)}\}, \quad \mathcal{S}_\lambda = \cup_{1 \leq j \leq \bar{k}} \{\lambda_1^{(j)}, \lambda_2^{(j)}\}. \quad (14)$$

For each j , we will obtain interpolation conditions at combinations of $\mu_1^{(j)}, \mu_2^{(j)}, \lambda_1^{(j)}, \lambda_2^{(j)}$. See (22) below.

The ROM is constructed using a Petrov-Galerkin projection with projection matrices given by the so-called generalized controllability and generalized observability matrices. The generalized controllability matrix is

$$\tilde{\mathcal{R}} = [\tilde{\mathcal{R}}^{(1)}, \tilde{\mathcal{R}}^{(2)}, \dots, \tilde{\mathcal{R}}^{(\bar{k})}] \in \mathbb{C}^{(n_v - n_p) \times k}, \quad (15)$$

where

$$\tilde{\mathcal{R}}^{(j)} = [\tilde{\boldsymbol{\Phi}}(\lambda_1^{(j)}) \tilde{\mathbf{b}}, \tilde{\boldsymbol{\Phi}}(\lambda_2^{(j)}) \tilde{\mathbf{Q}} (\tilde{\boldsymbol{\Phi}}(\lambda_1^{(j)}) \tilde{\mathbf{b}} \otimes \tilde{\boldsymbol{\Phi}}(\lambda_1^{(j)}) \tilde{\mathbf{b}})] \in \mathbb{C}^{(n_v - n_p) \times 2}, \quad (16)$$

$j = 1, \dots, \bar{k}$. The generalized observability matrix is

$$\tilde{\mathcal{O}} = \left[(\tilde{\mathcal{O}}^{(1)})^T, (\tilde{\mathcal{O}}^{(2)})^T, \dots, (\tilde{\mathcal{O}}^{(\bar{k})})^T \right]^T \in \mathbb{C}^{k \times (n_v - n_p)}, \quad (17)$$

where, for $j = 1, \dots, \bar{k}$,

$$\tilde{\mathcal{O}}^{(j)} = \begin{bmatrix} \tilde{\mathbf{c}}^T \tilde{\boldsymbol{\Phi}}(\mu_1^{(j)}) \\ \tilde{\mathbf{c}}^T \tilde{\boldsymbol{\Phi}}(\mu_1^{(j)}) \tilde{\mathbf{Q}}(\tilde{\boldsymbol{\Phi}}(\lambda_1^{(j)}) \tilde{\mathbf{b}} \otimes \tilde{\boldsymbol{\Phi}}(\mu_2^{(j)})) \end{bmatrix} \in \mathbb{C}^{2 \times (n_v - n_p)}. \quad (18)$$

Projecting (6) from the left and the right by the generalized observability matrix (17) and the generalized controllability matrix (15), respectively, gives a ROM whose transfer function components have the desired interpolation properties. Consider the ROM

$$\hat{\tilde{\mathbf{E}}} \frac{d}{dt} \hat{\tilde{\mathbf{v}}}(t) = \hat{\tilde{\mathbf{A}}} \hat{\tilde{\mathbf{v}}}(t) + \hat{\tilde{\mathbf{Q}}}(\hat{\tilde{\mathbf{v}}}(t) \otimes \hat{\tilde{\mathbf{v}}}(t)) + \hat{\tilde{\mathbf{b}}} \mathbf{g}(t), \quad (19a)$$

$$\hat{\tilde{\mathbf{y}}}(t) = \hat{\tilde{\mathbf{c}}}^T \hat{\tilde{\mathbf{v}}}(t) \quad (19b)$$

of state dimension k , where

$$\hat{\tilde{\mathbf{E}}} = \tilde{\mathcal{O}} \tilde{\mathbf{E}} \tilde{\mathcal{R}}, \quad \hat{\tilde{\mathbf{A}}} = \tilde{\mathcal{O}} \tilde{\mathbf{E}} \tilde{\mathcal{R}}, \quad \hat{\tilde{\mathbf{Q}}} = \tilde{\mathcal{O}} \tilde{\mathbf{Q}}(\tilde{\mathcal{R}} \otimes \tilde{\mathcal{R}}), \quad \hat{\tilde{\mathbf{b}}} = \tilde{\mathcal{O}} \tilde{\mathbf{b}}, \quad \hat{\tilde{\mathbf{c}}} = \tilde{\mathcal{R}}^T \tilde{\mathbf{c}}. \quad (20)$$

Analogous to (13), the first two transfer function components of (19) are

$$\hat{\mathbf{H}}_1(s_1) = \hat{\tilde{\mathbf{c}}}^T \hat{\tilde{\boldsymbol{\Phi}}}(s_1) \hat{\tilde{\mathbf{b}}}, \quad \hat{\mathbf{H}}_2(s_1, s_2, s_3) = \hat{\tilde{\mathbf{c}}}^T \hat{\tilde{\boldsymbol{\Phi}}}(s_1) \hat{\tilde{\mathbf{Q}}}(\hat{\tilde{\boldsymbol{\Phi}}}(s_2) \hat{\tilde{\mathbf{b}}} \otimes \hat{\tilde{\boldsymbol{\Phi}}}(s_3) \hat{\tilde{\mathbf{b}}}), \quad (21)$$

where $\hat{\tilde{\boldsymbol{\Phi}}}(s) = (s \hat{\tilde{\mathbf{E}}} - \hat{\tilde{\mathbf{A}}})^{-1}$. If $s \hat{\tilde{\mathbf{E}}} - \hat{\tilde{\mathbf{A}}}$ is invertible for all $s \in \mathcal{S}_\mu \cup \mathcal{S}_\lambda$, then the interpolation conditions

$$\mathbf{H}_1(\lambda_1^{(j)}) = \hat{\mathbf{H}}_1(\lambda_1^{(j)}), \quad \mathbf{H}_1(\mu_1^{(j)}) = \hat{\mathbf{H}}_1(\mu_1^{(j)}), \quad (22a)$$

$$\mathbf{H}_2(\lambda_2^{(j)}, \lambda_1^{(j)}, \lambda_1^{(j)}) = \hat{\mathbf{H}}_2(\lambda_2^{(j)}, \lambda_1^{(j)}, \lambda_1^{(j)}), \quad (22b)$$

$$\mathbf{H}_2(\mu_1^{(j)}, \lambda_1^{(j)}, \mu_2^{(j)}) = \hat{\mathbf{H}}_2(\mu_1^{(j)}, \lambda_1^{(j)}, \mu_2^{(j)}), \quad (22c)$$

for all $j = 1, \dots, \bar{k}$, and additional interpolation conditions are satisfied. This fact follows from [12, Lemma 3.1].

2.5 The Loewner Framework for Quadratic ODE Systems

As written, the ROM (19) requires the explicit projection by $\tilde{\mathcal{O}}$ and $\tilde{\mathcal{R}}$. The crucial observation that underlies the Loewner framework is that the matrices in (20) can be computed directly from measurements of the transfer function components, but without explicit projection.

The Loewner matrix $\mathbb{L}$ and the shifted Loewner matrix $\mathbb{L}_s$ are defined as

$$\mathbb{L} = -\tilde{\mathcal{O}} \tilde{\mathbf{E}} \tilde{\mathcal{R}} \in \mathbb{C}^{k \times k}, \quad \mathbb{L}_s = -\tilde{\mathcal{O}} \tilde{\mathbf{A}} \tilde{\mathcal{R}} \in \mathbb{C}^{k \times k} \quad (23)$$

and can be expressed directly in terms of measurements. Unfortunately, the expression of all terms in $\mathbb{L}$ and $\mathbb{L}_s$ becomes involved very quickly. Therefore we only consider one case to illustrate the idea and refer to [12] for a detailed discussion. Recall (15)–(18). The Loewner matrix $\mathbb{L}$ in (23) contains entries

$$\begin{aligned} & \tilde{\mathbf{c}}^T (\mu_1^{(j)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{E}} (\lambda_1^{(l)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{b}} \\ &= \frac{1}{\mu_1^{(j)} - \lambda_1^{(l)}} \left(\tilde{\mathbf{c}}^T (\mu_1^{(j)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1} (\mu_1^{(j)} - \lambda_1^{(l)}) \tilde{\mathbf{E}} (\lambda_1^{(l)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{b}} \right) \\ &= \frac{1}{\mu_1^{(j)} - \lambda_1^{(l)}} \left(\tilde{\mathbf{c}}^T (\mu_1^{(j)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1} \left((\mu_1^{(j)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}}) - (\lambda_1^{(l)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}}) \right) (\lambda_1^{(l)} \tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1} \tilde{\mathbf{b}} \right) \\ &= \frac{1}{\mu_1^{(j)} - \lambda_1^{(l)}} \left(\mathbf{H}_1(\lambda_1^{(l)}) - \mathbf{H}_1(\mu_1^{(j)}) \right), \end{aligned}$$

which are written in terms of measurements of the transfer function component $\mathbf{H}_1$. As mentioned before, the representation of other entries of $\mathbb{L}$ and entries of $\mathbb{L}_s$ follows the same ideas, but is more involved. See [12] for details.

So far we assumed that the ‘right’ amount of data is available so that $\tilde{\mathcal{O}}$ and $\tilde{\mathcal{R}}$ have full row and column rank, respectively, and the ROM (19) is well-posed. Another advantage of the Loewner framework is that larger amounts of data can be used. Since the ROMs can be directly computed from data, using all the measurements available intuitively leads to a better ROM. Specifically, the Loewner and shifted Loewner matrices $\mathbb{L}, \mathbb{L}_s \in \mathbb{C}^{k \times k}$ are computed from data and the SVD is used to extract the relevant data. Consider the (short) SVDs

$$[\mathbb{L} \quad \mathbb{L}_s] = \mathbf{Y}_1 \Sigma_1 \mathbf{X}_1^*, \quad \begin{bmatrix} \mathbb{L} \\ \mathbb{L}_s \end{bmatrix} = \mathbf{Y}_2 \Sigma_2 \mathbf{X}_2^*, \quad (24)$$

where $\Sigma_1 \in \mathbb{R}^{k \times 2k}$, $\Sigma_2 \in \mathbb{R}^{2k \times k}$, $\mathbf{Y}_1, \mathbf{X}_2 \in \mathbb{C}^{k \times k}$. 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

$$\tilde{\mathbf{V}} = \tilde{\mathcal{R}} \mathbf{X} \in \mathbb{C}^{(n_v - n_p) \times r}, \quad \tilde{\mathbf{W}} = \tilde{\mathcal{O}}^* \mathbf{Y} \in \mathbb{C}^{(n_v - n_p) \times r}, \quad (25)$$

The reduced order model that matches the desired interpolation/approximation property of the transfer function components is

$$\hat{\tilde{\mathbf{E}}} = -\mathbf{Y}^* \mathbb{L} \mathbf{X} = \tilde{\mathbf{W}}^* \tilde{\mathbf{E}} \tilde{\mathbf{V}}, \quad \hat{\tilde{\mathbf{A}}} = -\mathbf{Y}^* \mathbb{L}_s \mathbf{X}^* = \tilde{\mathbf{W}}^* \tilde{\mathbf{A}} \tilde{\mathbf{V}}, \quad (26a)$$

$$\hat{\tilde{\mathbf{Q}}} = \tilde{\mathbf{W}}^* \tilde{\mathbf{Q}} (\tilde{\mathbf{V}} \otimes \tilde{\mathbf{V}}), \quad \hat{\tilde{\mathbf{b}}} = \tilde{\mathbf{W}}^* \tilde{\mathbf{b}}, \quad \hat{\tilde{\mathbf{c}}}^T = \tilde{\mathbf{c}}^T \tilde{\mathbf{V}}. \quad (26b)$$

We have written (26) in terms of projections with $\tilde{\mathbf{V}}, \tilde{\mathbf{W}}$, but these quantities can be computed from measurements and $\mathbf{Y}, \mathbf{X}$. See [12].

2.6 Computational Details

The projection matrices $\tilde{\mathbf{V}}, \tilde{\mathbf{W}}$ and the ROM (26) are complex, but we can obtain real projection matrices $\tilde{\mathbf{V}}, \tilde{\mathbf{W}}$ and corresponding ROMs if the sets of interpolation points $\mathcal{S}_\mu \subset \mathbb{C}$ and $\mathcal{S}_\lambda \subset \mathbb{C}$ contain also the conjugate complex data. See [2, Appendix A.1] or [6, p. 360]. This is what we do in our implementation.

As we have mentioned earlier, the projection of the Navier-Stokes system (1) to obtain the ODE system (6) is a theoretical tool, but this projection is not carried out explicitly. The approaches in [3, 4, 14] can be extended to our case, and this extension is applied in the following computations. Because of space limitations, we will expand on these computational details elsewhere.

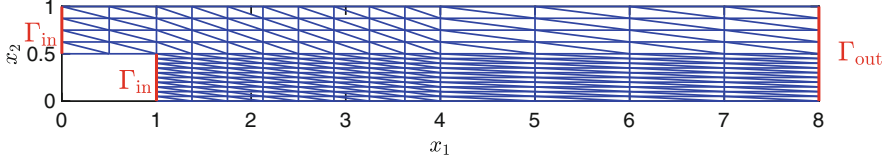


Fig. 1. Channel with backward facing step and coarse grid

3 Numerical Example

We extend an example modeled after [4, 14], where model reduction of the Oseen equation is considered. Let $\Omega \subset \mathbb{R}^2$ be the backward facing step geometry shown in Fig. 1. The boundary is decomposed into segments $\Gamma_{\text{out}}, \Gamma_d, \Gamma_{\text{in}}$, where $\Gamma_{\text{out}} = \{8\} \times (0, 1)$ is the outflow boundary, inputs are applied on $\Gamma_{\text{in}} = \{0\} \times (1/2, 1) \cup \{1\} \times (0, 1/2)$, and the velocities are set to zero on $\Gamma_D = \partial\Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}})$. Consider the Navier-Stokes equations

$$\partial_t v(x, t) - \nu \Delta v(x, t) + v(x, t) \cdot \nabla v(x, t) + \nabla p(x, t) = 0, \text{ in } \Omega \times (0, T), \quad (27a)$$

$$\nabla \cdot v(x, t) = 0, \text{ in } \Omega \times (0, T), \quad (27b)$$

$$(\nabla v(x, t) - p(x, t)I)n(x) + \frac{1}{\delta}v(x, t) = \frac{1}{\delta}g_{\text{in}}(x, t), \text{ on } \Gamma_{\text{in}} \times (0, T), \quad (27c)$$

$$v(x, t) = 0, \text{ on } \Gamma_D \times (0, T), \quad (27d)$$

$$(\nabla v(x, t) - p(x, t)I)n(x) = 0, \text{ on } \Gamma_{\text{out}} \times (0, T), \quad (27e)$$

where $\nu > 0$ is the viscosity and $\delta > 0$. The Robin boundary condition (27c) can be viewed as penalized version of a Dirichlet boundary condition as $\delta \rightarrow 0$.

We assume that the boundary input in (27c) is parameterized as

$$\mathbf{g}_{\text{in}}(x, t) = \mathbf{g}_1(t) \begin{pmatrix} \sin(2\pi(x_2 - 1/2)) \\ 0 \end{pmatrix} \text{ on } \{0\} \times (1/2, 1) \quad (28)$$

and $\mathbf{g}_{\text{in}}(x, t) = 0$ on $\{1\} \times (0, 1/2)$.

Our output is the integral of the curl of the velocity, often used in the context of control of vorticity,

$$y(t) = \int_{\Omega_{\text{obs}}} -\partial_{x_2} v_1(x, t) + \partial_{x_1} v_2(x, t) dx$$

over the subdomain $\Omega_{\text{obs}} = (1, 3) \times (0, 1/2)$ behind the backward facing step.

We use a $P_1 - P_2$ Taylor-Hood discretization (see, e.g., [11]) to arrive at the semi-discrete equations (1). The grid used in our computations is obtained from the coarse grid shown in Fig. 1 by uniform refinement. The used grid leads to $n_v = 11,489$, $n_p = 2,929$ degrees of freedom for the velocities and pressures, respectively. We use a viscosity of $\nu = 1/50$.

To generate the Loewner ROMs we use sets $\mathcal{S}_\mu \subset \mathbb{C}$ and $\mathcal{S}_\lambda \subset \mathbb{C}$ of left and right interpolation points with either $k = 50$ or $k = 100$ samples in each set. The frequencies are computed as follows: We first generate k real numbers $\omega_1, \dots, \omega_k \in \mathbb{R}$, logarithmically spaced between 10 and 10^3 , and then set $\mathcal{S}_\mu = \{\pm\omega_{2j-1}i : j = 1, \dots, k/2\}$ and $\mathcal{S}_\lambda = \{\pm\omega_{2j}i : j = 1, \dots, k/2\}$. The pairings of left interpolation points shown in (14) are as follows: $\mu_1^{(1)}, \mu_2^{(1)} = \omega_1 i, \omega_3 i$, $\mu_1^{(2)}, \mu_2^{(2)} = -\omega_1 i, -\omega_3 i$, $\mu_1^{(3)}, \mu_2^{(3)} = \omega_5 i, \omega_7 i$, $\mu_1^{(4)}, \mu_2^{(4)} = -\omega_5 i, -\omega_7 i$, etc. The right interpolation points are paired analogously.

The transfer function measurements are obtained via computational simulations; obtaining them from experiments is an open issue. Actually, in our computations we generate the generalized controllability and observability matrices. However, we do not generate them from (15), (17) but instead extend the approaches in [14] to compute them in terms of the original system (1).

Figure 2 shows the normalized singular values $\sigma_j^{(1)}/\sigma_1^{(1)}$ and $\sigma_j^{(2)}/\sigma_1^{(2)}$ of the matrices in (24) generated with $k = 50$ and with $k = 100$ frequency samples. For Loewner matrices generated with fixed number k of frequency samples the normalized singular values $\sigma_j^{(1)}/\sigma_1^{(1)}$ and $\sigma_j^{(2)}/\sigma_1^{(2)}$ are very similar, which has also been observed in other applications. Also, the decay of the normalized singular values for the matrices (24) generated with $k = 50$ and with $k = 100$ frequency samples is similar.

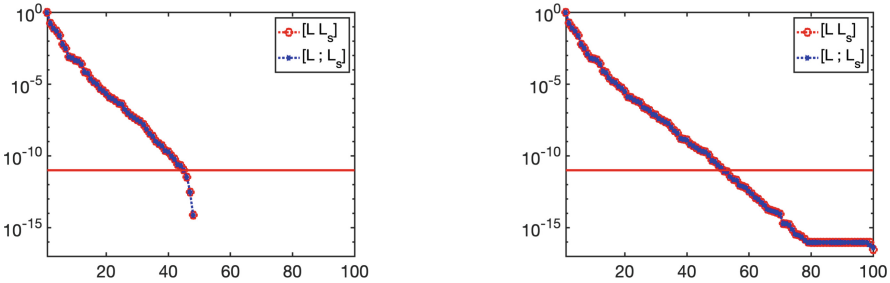


Fig. 2. Normalized singular values of the matrices (24) generated with $k = 50$ (left plot) and $k = 100$ (right plot) frequency samples each in $\mathcal{S}_\mu$ and in $\mathcal{S}_\lambda$. Solid line indicates $\text{tol} = 10^{-11}$ used to determine Loewner ROMs.

The size of the Loewner ROM is computed as the smallest r such that

$$\sigma_{r+1}^{(1)}/\sigma_{r+1}^{(1)} \leq \text{tol}.$$

We use $\text{tol} = 10^{-11}$ which gives Loewner ROM sizes $r = 45$ when $k = 50$ samples are used and $r = 51$ when $k = 100$ samples are used to generate the data.

To compare the FOM (1) and with the ROM (26) we perform time domain simulations over $[0, T] = [0, 2\pi]$ with input $\mathbf{g}_1(t) = \sin(t)$. The FOM and the ROM are solved numerically using the backward Euler method in time with time step size $T/100$.

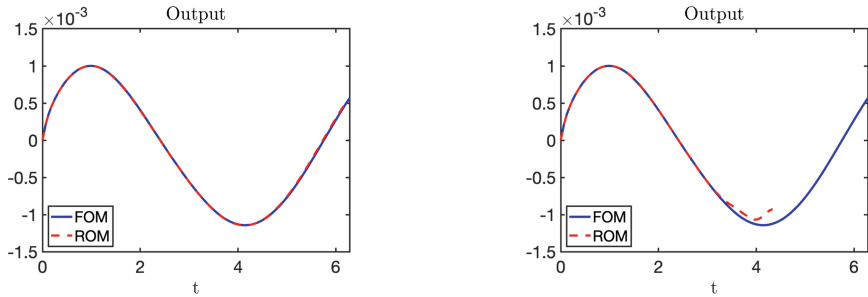


Fig. 3. System outputs computed with the standard Loewner Petrov-Galerkin ROMs generated from $k = 50$ (left plot) and $k = 100$ (right plot) frequency samples.

Figure 3 compares the FOM output against the outputs of the Loewner ROMs generated with $k = 50$ and with $k = 100$ frequency samples, respectively. The output of the Loewner ROM generated with $k = 50$ frequency samples is in excellent agreement with the FOM output (left plot). However, interestingly, when more data, i.e., $k = 100$ frequency samples are used to generate the Loewner ROM, then the ROM and FOM outputs begin to differ around $t = 3.5$. In fact, at time $t_{70} \approx 4.34$ Newton's method used to solve the equation in the backward Euler time stepping for the ROM fails to converge, and the ROM simulation is terminated (right plot in Fig. 3). It is illustrative to look at the velocities generated by the FOM and the Loewner ROMs. The magnitude of the velocities generated by the FOM and the Loewner ROMs are shown in Fig. 4. These plots indicate that the Loewner ROM generated with $k = 100$ frequency samples becomes unstable (right column in Fig. 4). There are also noticeable differences between the velocities generated by the FOM and by the Loewner ROM generated with $k = 50$ frequency samples (middle column in Fig. 4). However, since the Loewner ROM aims to approximate the input-to-output map $\mathbf{g} \mapsto \mathbf{y}$ of the system, only the state information (here the velocities) needed in the input-to-output map is well approximated, but the entire system states may not be well approximated.

The stability properties of the Loewner ROM, and the source of the instability of the Loewner ROM in this case are still under investigation. However, instability in the Loewner ROM when applied to Burgers' equation was also observed in [3]. The standard Loewner approach is based on a Petrov-Galerkin projection with projection matrices $\tilde{\mathbf{V}} \neq \tilde{\mathbf{W}}$. In [3] it was demonstrated that the stability

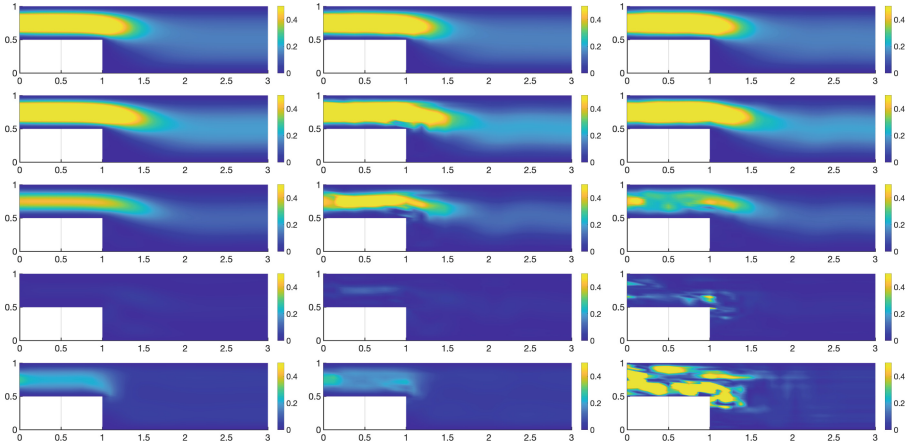


Fig. 4. Magnitudes of the velocity computed with the FOM (left column) and with the standard Loewner Petrov-Galerkin ROMs generated from $k = 50$ (middle column) and $k = 100$ (right column) frequency samples. Magnitudes of the velocities at times $t = 1.19$, $t = 1.82$, $t = 2.45$, $t = 3.08$, $t = 3.71$ are shown top to bottom.

properties can be maintained if the projection matrices $\tilde{\mathbf{V}}, \tilde{\mathbf{W}} \in \mathbb{R}^{(n_v - n_p) \times r}$ are merged into one larger matrix $[\tilde{\mathbf{V}}, \tilde{\mathbf{W}}] \in \mathbb{R}^{(n_v - n_p) \times 2r}$ (more precisely, an orthonormal basis of the columns of $[\tilde{\mathbf{V}}, \tilde{\mathbf{W}}]$ is computed to ensure that the resulting matrix is full rank), and this matrix is used to construct a Galerkin ROM. This resulting ROM is referred to as a Loewner Galerkin ROM. Actually, in the context of the Navier-Stokes equations the use of Galerkin ROMs may allow the extension of stability estimates for semidiscrete finite element approximations (see, e.g., [15, Sect. 9.2]) to the Galerkin ROM.

We merge the matrices $\tilde{\mathbf{V}} \neq \tilde{\mathbf{W}}$ computed using the standard Loewner approach with $k = 100$ frequency samples into a projection matrix $[\tilde{\mathbf{V}}, \tilde{\mathbf{W}}] \in \mathbb{R}^{(n_v - n_p) \times 2r}$ and construct a Loewner Galerkin ROM of size $2r = 102$. Figure 5 shows that the output of the Loewner Galerkin ROM is in excellent agreement with the output of the FOM.

The magnitude of the velocities generated by the FOM and the Loewner ROMs are shown in Fig. 4. In this case the velocities of the Loewner Galerkin ROM approximate the FOM velocities well at all simulation times.

While these results indicate that the Loewner Galerkin ROM performs well when the standard Loewner Petrov-Galerkin ROM suffers from instabilities, the computation of the Loewner Galerkin ROM is intrusive and requires projection of the system with $[\tilde{\mathbf{V}}, \tilde{\mathbf{W}}]$. Analysis of the source of instability in the standard Loewner Petrov-Galerkin ROM and possible remedies, which preserve the non-intrusive data-driven nature of this approach, are part of ongoing research.

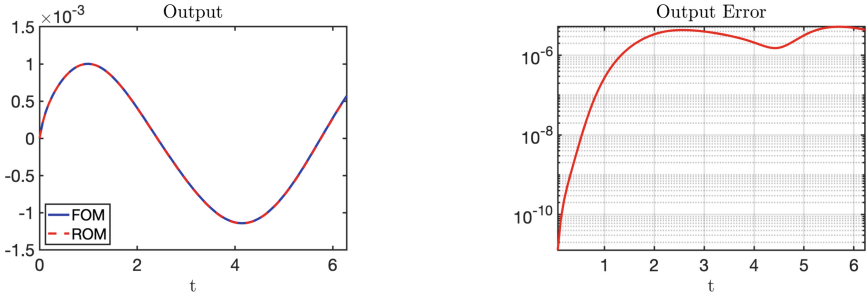


Fig. 5. System output computed with the Loewner Galerkin ROM generated from $k = 100$ frequency samples is in excellent agreement with the FOM output.

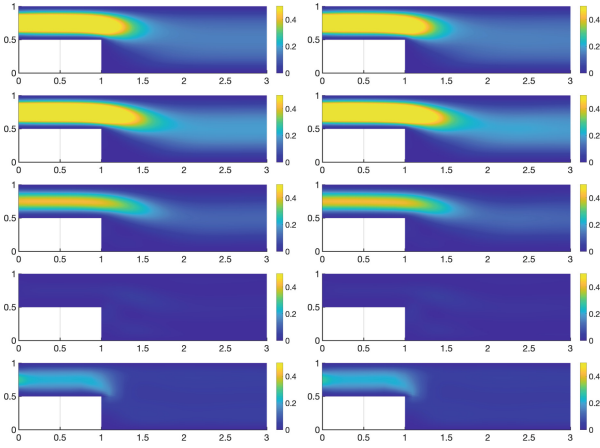


Fig. 6. Magnitudes of the velocity computed with the FOM (left column) and with the Loewner Galerkin ROMs generated from $k = 100$ (right column) frequency samples. Magnitudes of the velocities at times $t = 1.19$, $t = 1.82$, $t = 2.45$, $t = 3.08$, $t = 3.71$ are shown top to bottom.

4 Conclusions and Future Work

We have presented an extension of the Loewner framework to compute ROMs of quadratic-bilinear systems arising from semi-discretized incompressible Navier-Stokes equations.

The application to the Navier-Stokes equations showed the potential of the Loewner framework, but also raises a number of important questions that still need to be addressed. One question is data generation. Currently, transfer function component measurements are obtained via computational simulations; obtaining them from experiments is an open issue. The most important issue is stability. Our numerics have shown that the standard Loewner framework, which generates a Petrov-Galerkin reduced order model, $\tilde{\mathbf{W}} \neq \tilde{\mathbf{V}}$, may not be stable. We note that [8, page B255] also report instability of their ROM based

on Petrov-Galerkin and interpolation when applied to Burgers' equation with smaller viscosity. In our experiments, combining the projection matrices $\tilde{\mathbf{W}}, \tilde{\mathbf{V}}$ generated by the standard Loewner framework and applying a Galerkin projection with the larger projection matrix $[\tilde{\mathbf{W}}, \tilde{\mathbf{V}}]$ gave good results. However, using this Galerkin projection destroys the purely data driven aspect, since this Galerkin projection ROM is computed by explicitly projecting the FOM, while standard Loewner Petrov-Galerkin ROM can be computed from data alone.

The specific incompressible Navier-Stokes system (1) is somewhat limiting. First, the output (1c) does not depend on pressure. Extending the output to $\mathbf{y}(t) = \mathbf{C}_1 \mathbf{v}(t) + \mathbf{C}_2 \mathbf{p}(t)$ leads to an output of the type $\mathbf{y}(t) = \mathbf{C}_3 \boldsymbol{\Theta}_r \tilde{\mathbf{v}}(t) + \mathbf{C}_4 \mathbf{Q}(\boldsymbol{\Theta}_r \tilde{\mathbf{v}}(t) \otimes \boldsymbol{\Theta}_r \tilde{\mathbf{v}}(t))$ in the resulting projected system corresponding to (6). Thus in addition to the quadratic term in (6a) another quadratic term appears in the output. Incorporation of quadratic terms in the output equation is under investigation. Moreover, Dirichlet boundary condition inputs lead to inputs given by $\mathbf{g}$ as well as its derivative $\frac{d}{dt} \mathbf{g}$. These derivative terms leads to additional terms in projected equations and in the transfer function. They also impact the behavior of transfer function components at infinity, which has to be addressed by extending the approach in [4] for systems governed by the Oseen equation.

Acknowledgements. The authors gratefully acknowledge beneficial discussions with Drs. Antoulas and Gosea, constructive comments of two referees, as well as support by NSF grants CCF-1816219 and DMS-1819144.

References

1. Ahmad, M.I., Benner, P., Goyal, P., Heiland, J.: Moment-matching based model reduction for Navier-Stokes type quadratic-bilinear descriptor systems. *ZAMM Z. Angew. Math. Mech.* **97**(10), 1252–1267 (2017)
2. Antoulas, A.C., Beattie, C.A., Gugercin, S.: *Interpolatory model reduction*, Computational Science & Engineering, vol. 21. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2020). <https://doi.org/10.1137/1.9781611976083>, <https://doi.org/10.1137/1.9781611976083>
3. Antoulas, A.C., Gosea, I.V., Heinkenschloss, M.: On the Loewner framework for model reduction of Burgers' equation. In: King, R. (ed.) *Active Flow and Combustion Control 2018*, pp. 255–270. Springer-Verlag, Berlin, Heidelberg, New York (2018). https://doi.org/10.1007/978-3-319-98177-2_16, https://doi.org/10.1007/978-3-319-98177-2_16
4. Antoulas, A.C., Gosea, I.V., Heinkenschloss, M.: Data-driven model reduction for a class of semi-explicit DAEs using the Loewner framework. In: Reis, T., Grundel, S., Schöps, S. (eds.) *Progress in Differential-Algebraic Equations II*, pp. 185–210. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-53905-4_7, https://doi.org/10.1007/978-3-030-53905-4_7
5. 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>

6. Antoulas, A.C., Lefteriu, S., Ionita, A.C.: Chapter 8: a tutorial introduction to the Loewner framework for model reduction. In: Benner, P., Cohen, A., Ohlberger, M., Willcox, K. (eds.) *Model Reduction and Approximation: Theory and Algorithms*, pp. 335–376. SIAM, Philadelphia (2017). <https://doi.org/10.1137/1.9781611974829.ch8>, <https://doi.org/10.1137/1.9781611974829.ch8>
7. Bänsch, E., Benner, P., Saak, J., Weichelt, H.K.: Riccati-based boundary feedback stabilization of incompressible Navier-Stokes flows. *SIAM J. Sci. Comput.* **37**(2), A832–A858 (2015)
8. Benner, P., Breiten, T.: Two-sided projection methods for nonlinear model order reduction. *SIAM J. Sci. Comput.* **37**(2), B239–B260 (2015)
9. Benner, P., Goyal, P.: An iterative model reduction scheme for quadratic-bilinear descriptor systems with an application to Navier-Stokes equations. In: Keiper, W., Milde, A., Volkwein, S. (eds.) *Reduced-Order Modeling (ROM) for Simulation and Optimization*, pp. 1–19. Springer Verlag (2018). DOI: DOIurl10.1007/978-3-319-75319-5_1 https://doi.org/10.1007/978-3-319-75319-5_1
10. Benner, P., Goyal, P., Gugercin, S.: $\mathcal{H}_2$ -Quasi-optimal model order reduction for quadratic-bilinear control systems. *SIAM J. Matrix Anal. Appl.* **39**(2), 983–1032 (2018)
11. 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>, <http://dx.doi.org/10.1093/acprof:oso/9780199678792.001.0001>
12. Gosea, I.V., Antoulas, A.C.: Data-driven model order reduction of quadratic-bilinear systems. *Numer. Linear Algebra Appl.* **25**(6), e2200 (2018)
13. Gu, C.: QLMOR: a projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems. *Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Trans.* **30**(9), 1307–1320 (2011)
14. Heinkenschloss, M., Sorensen, D.C., Sun, K.: Balanced truncation model reduction for a class of descriptor systems with application to the Oseen equations. *SIAM J. Sci. Comput.* **30**(2), 1038–1063 (2008)
15. Layton, W.: Introduction to the numerical analysis of incompressible viscous flows. In: *Computational Science & Engineering*, vol. 6. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2008). <https://doi.org/10.1137/1.9780898718904>, <https://doi.org/10.1137/1.9780898718904>
16. Mayo, A.J., Antoulas, A.C.: A framework for the solution of the generalized realization problem. *Linear Algebra Appl.* **425**(2-3), 634–662 (2007). <https://doi.org/10.1016/j.laa.2007.03.008>, <https://doi.org/10.1016/j.laa.2007.03.008>
17. Rugh, W.J.: *Nonlinear System Theory. The Volterra/Wiener approach*. Johns Hopkins Series in Information Sciences and Systems. Johns Hopkins University Press, Baltimore (1981). PDF version with corrections (2002). <https://sites.google.com/site/wilsonjrugh>. Accessed 13 Feb 2018