A computational method for model reduction in index-2 dynamical systems for Stokes equations

A. Chkifa, M.A. Hamadi, Khalide Jbilou, A. Ratnani

Version of Record: https://www.sciencedirect.com/science/article/pii/S0898122121002935
Manuscript_997613376187928e73ac0a3a5deba23f

A computational method for model reduction in index-2 dynamical systems for Stokes equations

A. Chkifa[a], M.A. Amine[a,b], K. Jbilou[a,b], A. Ratnani[a]

[a] *Lab. MSDA, Mohammed VI Polytechnic University , Lot 660, Hay Moulay Rachid, Ben Guerir, 43150 Maroc*
[b] *Lab. LMPA, University of Littoral Côte d'Opale, 50 Rue F. Buisson, BP 699–62228 Calais cedex, France*

Abstract

Our aim through this paper is to describe a Krylov based projection method in order to reduce high-order dynamical systems. We focus on differential algebraic equations (DAEs) of index-2 that arise from spatial discretization of Stokes equations. An efficient algorithm based on a projection technique onto an extended block Krylov subspace that appropriately allows us to construct a reduced order system is described. Numerical results are provided to confirm the performance of the derived method compared with other known ones.

Keywords: Index 2 systems, Model order reduction, Projection methods, Stokes equations, Transfer function.

1. Introduction

Complex dynamical systems arising from descritization in space of a PDEs that modelize some physic or engineering problems, have generally high dimensionality and this leads to a computational problem to treat those dynamical systems for simulation or control. Therefore, the need for an appropriate method to reduce the large dimension is a very important task and this is usually called Model Order Reduction. Many works have been realized in the recent years on model reduction [1, 2, 3, 4, 5, 6]. The main idea is to approach the given large-scale dynamical system of differential equations by a new low-dimensional system that has almost the same response characteristics. Consider the Multiple Input Multiple Output (MIMO in short) linear time-invariant (LTI) system, described by the following state-space dynamical system

$$(\Sigma)\begin{cases} M\,\dot{x}(t) &= A\,x(t) + B\,u(t), \qquad x(0) = 0, \\ y(t) &= C\,x(t) + D\,u(t), \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $u(t) \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^s$ are the input and the output vectors of the system (1), respectively. The matrices A, $M \in \mathbb{R}^{n\times n}$ are large and sparse, $B \in \mathbb{R}^{n\times p}$

Email addresses: chkifa.abdellah@um6p.ma (A. Chkifa), amine.hamadi@um6p.ma (M.A. Amine), jbilou@univ-littoral.fr (K. Jbilou), ahmed.ratnani@um6p.ma (A. Ratnani)

and $C^T \in \mathbb{R}^{n\times s}$ are supposed to have small number of columns i.e., $p, s \ll n$. For simplicity we assume that $p = s$. If M is singular then the system (1) is called a "descriptor system", otherwise it is called a generalized system. The main idea of model reduction is to consider a reduced model that approximates the original one in some sense and having the form

$$(\Sigma_m) \begin{cases} M_m\, \dot{x}_m(t) &= A_m\, x_m(t) + B_m\, u(t), \\ y_m(t) &= C_m\, x_m(t) + D_m\, u(t), \end{cases} \tag{2}$$

where M_m, $A_m \in \mathbb{R}^{m\times m}$, B_m, $C_m^T \in \mathbb{R}^{m\times p}$, $D_m = D \in \mathbb{R}^{p\times p}$ and $m \ll n$. Different computational methods for deriving the low order dynamical system (2) have been proposed the last years. The first class contains the well known balanced truncation method which is based on a singular value decomposition. The second class uses Krylov-based methods and contains methods such as moment matching techniques. The main advantage of SVD methods is that they preserve important properties such as stability, and also provide an error-norm bound. However, they are not suited for large scale dynamical systems and consequently are impractical for large systems since they require a computational cost of order $\mathcal{O}(n^3)$ and storage of order $\mathcal{O}(n^2)$. On the other hand, Krylov-based methods use projections onto some well chosen Krylov-type subspaces and require lower cost and storage as compared to SVD-based methods since only matrix-vector multiplications are required. The major challenge with balanced truncation based methods is that they require solving two large Lyapunov equations [7, 8]. A mixing of these two methods has been elaborated in [9], where the authors solve the two required large Lyapunov matrix equations in balanced truncation by using projection onto some block Krylov subspaces.

Applying the Laplace transform

$$L(f)(z) := \int_0^\infty e^{-zt} f(t) dt,$$

to the system(1), gives the following system in the frequency domain

$$\begin{cases} z M\, X(z) &= A\, X(z) + B\, U(z), \\ Y(z) &= C\, X(z) + D\, U(z). \end{cases}$$

By eliminating $X(z)$ from the two equations, we obtain

$$Y(z) = F(z)\, U(z),$$

where

$$F(z) = C\, (z\, M - A)^{-1}\, B + D,$$

is the transfer function associated the original system (1). In a similar way, we can obtain a transfer function $F_m(z)$ associated to the reduced system (2) as

$$F_m(z) = C_m\, (z\, M_m - A_m)^{-1}\, B_m + D_m.$$

Therefore,

$$\|Y(z) - Y_m(z)\| \leq \|F(z) - F_m(z)\|\|U(z)\|.$$

In order to measure the accuracy of the resulting reduced system, we have to compute the error $\|F - F_m\|$ with respect to a specific norm which tells us how the response of the reduced system is close to that of the original one.

The determination of a reduced order dynamical system via projections using bases from Krylov-type subspaces can be described as follows. We construct the block matrix $\mathbb{V}_m = [V_1, \ldots, V_m]$ whose columns form an orthonormal basis of some Krylov-type subspace via an Arnoldi-based or Lanczos-based processes. Then, we approximate the full order state $x(t)$ by $\mathbb{V}_m x_m(t)$, and by enforcing the Petrove-Galerkin condition, we obtain the projected system

$$\begin{cases} \mathbb{V}_m^T(M\mathbb{V}_m\,\dot{x}_m(t) & -A\,\mathbb{V}_m x_m(t) - B\,u(t)) = 0, \\ y_m(t) & = C\mathbb{V}_m\,x_m(t) + D\,u(t). \end{cases} \tag{3}$$

Therefore, the obtained reduced order system can be expressed as

$$\begin{cases} M_m\,\dot{x}_m(t) &= A_m\,x_m(t) + B_m\,u(t), \\ y_m(t) &= C_m\,x_m(t) + D\,u(t), \end{cases} \tag{4}$$

where $M_m = \mathbb{V}_m^T M \mathbb{V}_m$, $A_m = \mathbb{V}_m^T A \mathbb{V}_m \in \mathbb{R}^{m\times m}$, $B_m = \mathbb{V}_m^T B \in \mathbb{R}^{m\times p}$, $C_m = C\,\mathbb{V}_m \in \mathbb{R}^{p\times m}$ and $D \in \mathbb{R}^{p\times p}$ with $m \ll n$.

In this work, we deal with a specific large system that is depicted from a spatial discretization of Stokes equations; here D is the zero matrix. It is known as the index-2 systems and has the following structure

$$\begin{cases} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} &= \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t), \\ y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \end{cases} \tag{5}$$

Through this paper, we describe a model reduction technique via Krylov-based subspace method in order to construct an efficient reduced system of 5 that has nearly the same response characteristics. Our methods work well for a class of descriptor dynamical systems represented by a set of ordinary differential equations(ODEs). Unfortunately, this is not our case since the dynamical system 5 is represented by a set of differential algebraic equations(DAEs) and therefore these methods are not directly applicable to 5. To overcome this problem, we present in this paper a technique that allows us to transform the DAE system to an ODE one using a Leray projection. Moreover, we give a simplification on how to avoid this dense projection matrix while we perform

our process to get a reduced system.

Such a system (5) appears also after a spatial discretization of the linearised Navier-Stokes equations around a steady state. The difference is that with Stokes equations, the matrix A is symmetric definite negative, while for Navier-Stokes equations, the matrix A is neither symmetric nor negative or positive definite, see [10]. Numerous model reduction methods have been explored for Navier-Stokes equations using balanced truncation and proper orthogonal decomposition [11, 12]. A balanced truncation model reduction method for the Ossen equations has been investigated in [13].

The remainder of this paper is organised as follows. Section 2 describes the incompressible Stokes equations, and its differential-algebraic system of differential index-2 that arise after a mixed finite element discretization. An extended block Krylov subspace method used to construct a simplified system from the complex one 5 is explained in Section 3. In Section 3 we give some properties and describe an algorithm that allows us to construct an orthonormal matrix in an efficient way by avoiding the dense projection matrix that appears after the transformation to an ODEs. Section 4 is devoted to some numerical experiments to show the effectiveness of the proposed approach.

2. The incompressible Stokes equations

We consider the incompressible Stokes equations

$$\frac{\partial v}{\partial t} - \nu \Delta v + \nabla p = f, \tag{6a}$$

$$\nabla \cdot v = 0, \tag{6b}$$

where the vector $v(t,x) = [v_1(t,x), v_2(t,x)] \in \mathbb{R}^2$ refers to the velocity, $p(t,x) \in \mathbb{R}$ is the pressure filed, f is known as the forcing term and $\nu \in \mathbb{R}^+$ is the viscosity. The operators Δ, ∇ and $\nabla\cdot$ are defined as the Laplacien, Gradient and divergence operators, respectively. The choice of an appropriate discretization procedure depends on the specific governing equations used, for example (compressible or incompressible flow(*our case*)), mesh type (structured or unstructured), etc. The classical discretisation techniques are finite difference, finite element and finite volume. One of the known methods used to discretize instationary problems is the method of lines which is based on the replacement of the spatial derivatives in the PDEs with algebraic approximations leading to a system of ODEs that approximates the original PDE problem. In this paper we use a mixed finite element method to discretize the Stokes equations in space, see [14]. After the discretization, we get a system of differential-algebraic equations of the form

$$M\,\frac{d}{dt}\mathbf{v}(t) = \frac{1}{\mathrm{Re}}A\mathbf{v}(t) + G\mathbf{p}(t) + \mathbf{f}(t), \tag{7a}$$

$$0 = G^T\mathbf{v}(t), \tag{7b}$$

where $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ is the nodal vector of the discretized velocity, $\mathbf{p}(t) \in \mathbb{R}^{n_p}$ is the discretized pressure and $\mathrm{Re} = 1/\nu$ is the Reynolds number.

In what follows, we refer to A as $\dfrac{1}{\mathrm{Re}}A$ and assume that the forcing term $f(t)$ is given by

$$\mathbf{f}(t) = Bu(t),$$

where $u(t)$ is the input vector and $B \in \mathbb{R}^{n_v \times n_b}$ is a full rank matrix. Moreover, the mass matrix $M \in \mathbb{R}^{n_v \times n_v}$ is symmetric and postive-definite, the discrete Laplacian matrix $A \in \mathbb{R}^{n_v \times n_v}$ is symmetric and negative definite, and $G \in \mathbb{R}^{n_v \times n_p}$ represents the discrete gradient matrix. We add to the system (7) an output function given by

$$y(t) = C\mathbf{v}(t),$$

where $y(t)$ is the output vector and $C^T \in \mathbb{R}^{n_v \times n_c}$ is a full rank matrix. The dynamical system (7) can be rewritten in a new form as follows

$$\begin{cases} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}(t) \\ \dot{\mathbf{p}}(t) \end{bmatrix} = \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{p}(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t), \\ y(t) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{p}(t) \end{bmatrix}, \end{cases} \tag{8}$$

which uses the following matrix-pencil

$$\left(\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \right). \tag{9}$$

This symmetric matrix pencil has $n_v - n_p$ finite eigenvalues $\lambda_i \in \mathbb{R}^-$ and $2n_p$ infinite eigenvalues $\lambda_\infty = \infty$, see [[15], Theorem 2.1]. The dynamical system (8) is known as an index-2 descriptor dynamical system, see [16] for more details. To guarantee a well processing of our proposed method, we need to establish a transformation of the system (7) into an ordinary differential equations (ODEs). To this end, we need to introduce the discrete Leray projector denoted by Π. After using this transformation, an adapted extended block Krylov subspace projection method is described. This projection method allows us to construct a reduced system with a reasonable computational costs. In the next section, we give a brief description of the extended block Krylov Arnoldi method.

3. Applying the extended block Krylov subspace method

3.1. The extended block Arnoldi process

For $A \in \mathbb{R}^{n_v \times n_v}$ and $B \in \mathbb{R}^{n_v \times n_b}$, the extended block Krylov subspace is defined as follows

$$\mathbb{K}_m^{ext}(A,B) = Range([A^{-m}B, \cdots, A^{-1}B, B, AB, \ldots, A^{m-1}B]).$$

Druskin and Knizhnerman introduced in [17] the extended *Lanczos* process to approximate the action of a matrix function $f(A)$ on a vector v where A is a symmetric matrix. The extended block *Arnoldi* process to solve *Lyapunov* equations was defined in [18, 19]. Moreover, a model reduction method for large scale dynamical systems based on a projection technique onto $\mathbb{K}_m^{ext}(A,B)$ has been established in [1]. We can generate an orthonormal basis formed by the columns of $\{V_1, \ldots, V_m\}$ of the subspace $\mathbb{K}_m^{ext}(A,B)$ using the following extented block *Arnoldi* algorithm [18, 19].

Algorithm 1 The extended-block Arnoldi algorithm

- Input : $A \in \mathbb{R}^{n_v \times n_v}$, $B \in \mathbb{R}^{n_v \times n_b}$ and a fixed integer m.
- Compute $V_1 = \mathtt{qr}([B, A^{-1}B], 0)$(skinny $\mathtt{qr}$), $\mathbb{V}_1 = [V_1]$.
- for $j = 1 \cdots m-1$
  1. Set $V_j^{(1)}$: first n_b columns of V_j; $V_j^{(2)}$: second n_b columns of V_j.
  2. $\tilde{V}_{j+1} = [A\,V_j^{(1)}, A^{-1}V_j^{(2)}]$.
  3. Orthogonalization step:

     for $i = 1, 2 \cdots j$

     $H_{ij} = V_i^T\,\tilde{V}_{j+1}$.

     $\tilde{V}_{j+1} = \tilde{V}_{j+1} - V_i\,H_{ij}$.

     end for
- $\mathtt{qr}(\tilde{V}_{j+1}) = V_{j+1}\,H_{j+1,j}$.
- $\mathbb{V}_{j+1} = [\mathbb{V}_j, V_{j+1}]$.
- End.

After m steps, Algorithm 1 builds a matrix $\mathbb{V}_m = [V_1, \ldots, V_m] \in \mathbb{R}^{n_v \times 2mn_b}$ corresponding to the orthonormal basis of the extended block Krylov subspace and a block upper Hessenberg matrix $\mathbb{H}_m \in \mathbb{R}^{2mn_b \times 2mn_b}$ whose nonzeros blocks are the $H_{i,j}$. Note that each sub-matrix $H_{i,j}$ $(1 \leq i \leq j \leq m)$ is of order $2n_b \times 2n_b$. Let $\mathbb{T}_m = \mathbb{V}_m^T A \mathbb{V}_m \in \mathbb{R}^{2mn_b \times 2mn_b}$ denote the restriction of the matrix A to the extended block Krylov subspace. It was shown in [19] that $\mathbb{T}_m$ is also a block

upper Hessenberg matrix. Assume that m steps of Algorithm 1 have been run, then we get the following classical algebraic relations

$$A\,\mathbb{V}_m = \mathbb{V}_{m+1}\,\overline{\mathbb{T}}_m \tag{10}$$

$$= \mathbb{V}_m\,\mathbb{T}_m + V_{m+1}\,T_{m+1,m}E_m^T, \tag{11}$$

where $\overline{\mathbb{T}}_m = \mathbb{V}_{m+1}^T A\mathbb{V}_m$ and E_m is the last $2n_b$ columns of the identity I_{2mn_b}.

3.2. Deriving the ODE system

We first eliminate the discrete pressure $\mathbf{p}$ from (7a) using the following projection operator

$$\Pi = I_n - G\,(G^T M^{-1} G)^{-1}\,G^T M^{-1} \in \mathbb{R}^{n_v \times n_v}.$$

It is easy to check that

$$(\Pi^T)^2 = \Pi^T, \quad \Pi^2 = \Pi, \quad \Pi\,G = 0, \quad \Pi\,M = M\,\Pi^T \text{ and } M^{-1}\,\Pi = \Pi^T M^{-1}.$$

The projection Π^T is an M-orthogonal projection where for x, $y \in \mathbb{R}^{n_v}$ and $M \in \mathbb{R}^{n_v \times n_v}$, the M-inner product is defined by

$$< x, y >_M = (x, My) = y^T Mx \quad (M \text{ is a symmetric and positive-definite}).$$

Notice that

$$\text{null}(\Pi^T) = \text{range}(M^{-1}G) \quad \textit{and} \quad \text{range}(\Pi^T) = \text{null}(G).$$

By using all these properties we can show that

$$0 = G^T\mathbf{v}(t) \qquad \text{if and only if} \qquad \mathbf{v}(t) = \Pi^T\mathbf{v}(t). \tag{12}$$

Multiplying (7a) by $G^T\,M^{-1}$ and using (7b), the term $\mathbf{p}$ can be expressed as follows

$$\mathbf{p}(t) = -(G^T\,M^{-1}\,G)^{-1}G^T\,M^{-1}\,A\mathbf{v}(t) - (G^T\,M^{-1}\,G)^{-1}G^T\,M^{-1}\,B\mathbf{u}(t).$$

Replacing $\mathbf{p}$ in (7a) and multiplying by Π yields to the following projected system

$$\mathcal{M}\,\frac{d}{dt}\mathbf{v}(t) = \mathcal{A}\mathbf{v}(t) + \mathcal{B}\mathbf{u}(t), \tag{13a}$$

$$\mathbf{y}(t) = \mathcal{C}\mathbf{v}(t). \tag{13b}$$

where $\mathcal{A} = \Pi\,A\,\Pi^T$, $\mathcal{M} = \Pi\,M\,\Pi^T$, $\mathcal{B} = \Pi\,B$ and $\mathcal{C} = C\,\Pi^T$. Since the matrix-pencil (9) has $n_v - n_p$ finite eigenvalues [15], a decomposition of Π can be obtained by using the thin singular value decomposition which leads to the following decomposition

$$\Pi = \Theta_l\Theta_r^T,$$

where Θ_l, $\Theta_r \in \mathbb{R}^{n_v \times (n_v - n_p)}$, are full rank matrices satisfying

$$\Theta_l^T \Theta_r = \mathrm{I}_{n_v - n_p}.$$

Using this decomposition into (13) and defining a new variable $\tilde{\mathbf{v}}(t) = \Theta_l^T \mathbf{v}(t)$ with $\Theta_r \tilde{\mathbf{v}}(t) = \Theta_r \Theta_l^T \mathbf{v}(t) = \Pi^T \mathbf{v}(t) = \mathbf{v}(t)$, we get the following ODE system

$$M_\Theta \frac{d}{dt} \tilde{\mathbf{v}}(t) = A_\Theta \tilde{\mathbf{v}}(t) + B_\Theta \mathbf{u}(t), \tag{14a}$$

$$y(t) = C_\Theta \tilde{\mathbf{v}}(t), \tag{14b}$$

where $M_\theta = \Theta_r^T M \Theta_r$, $A_\theta = \Theta_r^T A \Theta_r \in \mathbb{R}^{(n_v - n_p) \times (n_v - n_p)}$, $B_\theta = \Theta_r^T B \in \mathbb{R}^{(n_v - n_p) \times n_b}$ and $C_\theta = C\Theta_r \in \mathbb{R}^{n_c \times (n_v - n_p)}$. The matrix M_θ is non-singular due to the fact that M is symmetric and defini-positive. Notice that the three systems (7), (13) and (14) are equivalent in the sense that their finite spectrum is the same and they realize the same transfer function, see [20]. We denote by $F_\Theta(s) = C_\Theta(sM_\Theta - A_\Theta)^{-1} B_\Theta = C_\Theta X$ the transfer function associated to the system (14) where X is defined by $X = \Theta_r (sM_\Theta - A_\Theta)^{-1} B_\Theta$ with $B_\Theta = (sM_\Theta - A_\Theta)\Theta_l^T X$ and $\Pi B = \Pi(sM - A)\Pi^T X$. Using the relation (12) and the fact that $null(\Pi) = range(G)$, we can verify that X solves the following saddle point problem

$$\begin{bmatrix} sM - A & -G \\ -G^T & 0 \end{bmatrix} \begin{bmatrix} X \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix},$$

where $\star$ stands for a block vector of order $n_p \times n_b$ that we are not interested in. Therefore, we get

$$F_\Theta(s) = C\,X = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} sM - A & -G \\ -G^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} B \\ 0 \end{bmatrix} = F(s), \tag{15}$$

where $F(s)$ is the transfer function associated to the original dynamical system (7). The technique used here allows us to solve a saddle point problem instead of solving a linear system depending on the dense matrix Π and its Θ-decomposition as was done in [13].

Remark 1. *We notice that instead of reducing the original system (8), we can reduce the ODE system (14) since they have the same transfer functions as it is stated in (15).*

The matrices involved in (14) are dense due to the projector Π and its Θ-decomposition, and that is why we need a strategy to avoid using directly computations with these matrices. An approximation of the original system (14) by a reduced one, without requiring any explicit computation of the dense matrices $(M_\Theta, A_\Theta, B_\Theta, C_\Theta)$ will be described in next subsection. Only the structure of the original system (5) will be used. Our calculations involve using the system 14 implicitly and this leads to solve some saddle point problems. Details will be given later.

3.3. The extended block Arnoldi algorithm for the Θ system

Multiplying from the left by the inverse of M_θ the first equation of the system (14), which gives the following dynamical system that we called the Θ-system

$$\frac{d}{dt}\tilde{\mathbf{v}}(t) = M_\Theta^{-1} A_\Theta \tilde{\mathbf{v}}(t) + M_\Theta^{-1} B_\Theta \mathbf{u}(t), \tag{16a}$$

$$y(t) = C_\Theta \tilde{\mathbf{v}}(t). \tag{16b}$$

Our goal is to find a reduced order system to 16 since it realizes the same transfer function of system 8 as we mentioned before. This new system can be constructed using a projection technique onto the extended block Krylov subspace $\mathbb{K}_m^{ext}(M_\Theta^{-1}A_\Theta, M_\Theta^{-1}B_\Theta)$. We apply the extended block Arnoldi Algorithm 1 to the pair $(M_\Theta^{-1}A_\Theta, M_\Theta^{-1}B_\Theta)$ which allows to construct an orthonormal basis formed by the columns of $\{\mathcal{V}_1^b, \ldots, \mathcal{V}_m^b\}$, where $\mathcal{V}_j^b$ for $(j = 1, \ldots, m)$ in an $(n_v - n_p) \times 2n_b$ matrix.

Proposition 1. *Let $\mathcal{V}_m = [\mathcal{V}_1^b, \ldots, \mathcal{V}_m^b] \in \mathbb{R}^{2mn_b \times 2mn_b}$ be the matrix generated by the extended block Arnoldi Algorithm 1 for the pair $(M_\Theta^{-1}A_\Theta, M_\Theta^{-1}B_\Theta)$. Then we have the following relations*

$$M_\Theta^{-1} A_\Theta \, \mathcal{V}_m = \mathcal{V}_{m+1} \, \overline{\mathbb{T}}_m \tag{17}$$

$$= \mathcal{V}_m \, \mathbb{T}_m + \mathcal{V}_{m+1}^b \, T_{m+1,m} E_m^T, \tag{18}$$

where T_{m+1} is the last $2n_b \times 2n_b$ block of $\overline{\mathbb{T}}_m \in \mathbb{R}^{2(m+1)n_b \times 2mn_b}$ and E_m^T is the last $2n_b$ columns of the identity matrix I_{2mn_b}.

Proof 1. *Using the fact that $M_\Theta^{-1}A_\Theta \mathbb{K}_m^{ext}(M_\Theta^{-1}A_\Theta, M_\Theta^{-1}B_\Theta) \subset \mathbb{K}_{m+1}^{ext}(M_\Theta^{-1}A_\Theta, M_\Theta^{-1}B_\Theta)$ and the orthogonality of $\mathcal{V}_m$, there exists a matrix L such that*

$$M_\Theta^{-1} A_\Theta \, \mathcal{V}_m = \mathcal{V}_{m+1} \, L. \tag{19}$$

Since $\mathcal{V}_{m+1} = [\mathcal{V}_m, \mathcal{V}_{m+1}^b]$, we have

$$\begin{aligned}
\mathbb{T}_{m+1} &= \mathcal{V}_{m+1}^T \, M_\Theta^{-1} A_\Theta \, \mathcal{V}_{m+1} \\
&= \begin{bmatrix} \mathcal{V}_m^T \, M_\Theta^{-1} A_\Theta \, \mathcal{V}_m & \mathcal{V}_m^T M_\Theta^{-1} A_\Theta \, \mathcal{V}_{m+1}^b \\ (\mathcal{V}_{m+1}^b)^T \, M_\Theta^{-1} A_\Theta \, \mathcal{V}_m & (\mathcal{V}_{m+1}^b)^T M_\Theta^{-1} A_\Theta \mathcal{V}_{m+1}^b \end{bmatrix} \\
&= \begin{bmatrix} \mathbb{T}_m & \mathcal{V}_m^T M_\Theta^{-1} A_\Theta \, \mathcal{V}_{m+1}^b \\ (\mathcal{V}_{m+1}^b)^T \, M_\Theta^{-1} A_\Theta \, \mathcal{V}_m & (\mathcal{V}_{m+1}^b)^T M_\Theta^{-1} A_\Theta \mathcal{V}_{m+1}^b \end{bmatrix}.
\end{aligned}$$

As $\mathbb{T}_{m+1}$ is an upper block Hessenberg matrix, it follows that

$$T_{m+1} \, E_m^T = (\mathcal{V}_{m+1}^b)^T \, M_\Theta^{-1} A_\Theta \, \mathcal{V}_m,$$

and

$$\overline{\mathbb{T}}_m = \mathcal{V}_{m+1}^T M_\Theta^{-1} A_\Theta \mathcal{V}_m = \begin{bmatrix} \mathbb{T}_m \\ T_{m+1,m} E_m^T \end{bmatrix} \in \mathbb{R}^{2(m+1)n_b \times 2mn_b}.$$

Multiplication from the left the equation 19 by $\mathcal{V}_{m+1}^T$*, we obtain* $\overline{\mathbb{T}}_m = L$*. Then*

$$M_\Theta^{-1} A_\Theta \, \mathcal{V}_m = \mathcal{V}_{m+1} \, \overline{\mathbb{T}}_m = [\mathcal{V}_m, \mathcal{V}_{m+1}^b] \begin{bmatrix} \mathbb{T}_m \\ T_{m+1,m} E_m^T \end{bmatrix} \tag{20}$$

$$= \mathcal{V}_m \, \mathbb{T}_m + \mathcal{V}_{m+1}^b \, T_{m+1,m} E_m^T, \tag{21}$$

which ends the proof.

In what follows, we describe an appropriate process to get a reduced system to 16 by avoiding an explicit computation of $\mathcal{V}_m$, since the j-th bloc $\mathcal{V}_j^b$ of $\mathcal{V}_m$ relies on Θ_r, and this calculation is not allowed in our approach, due to the density that will make our computations infeasible.
The main computational issue when we apply Algorithm 1 to the pair $(M_\Theta^{-1} A_\Theta, M_\Theta^{-1} B_\Theta)$ is to compute blocks of the form

$$M_\Theta^{-1} B_\Theta, (M_\Theta^{-1} A_\Theta)^{-1} \, M_\Theta^{-1} B_\Theta, (M_\Theta^{-1} A_\Theta) \, \mathcal{V}_j^{(1)} \text{ and } (M_\Theta^{-1} A_\Theta)^{-1} \, \mathcal{V}_j^{(2)},$$

for $j = 1, \ldots, m$. Where $\mathcal{V}_j^{(1)}$ and $\mathcal{V}_j^{(2)}$ are the first and second n_b columns of $\mathcal{V}_j^b$, respectively. Our strategy consists in reformulating those blocks onto new ones without an explicit calculation of Θ_r.
We set $\mathbb{V}_m = \Theta_r \mathcal{V}_m \in \mathbb{R}^{n_v \times 2mn_b}$ satisfying

$$\Pi^T \mathbb{V}_m = \Theta_r \Theta_l^T \mathbb{V}_m = \Theta_r \mathcal{V}_m = \mathbb{V}_m, \tag{22}$$

and denote by $V_j = \Theta_r \mathcal{V}_j^b$ the new j-th block of $\mathbb{V}_m$ that we aim to compute. In what follows, we show how to compute the block $V_j \in \mathbb{R}^{n_v \times 2n_b}$ in an appropriate way without using directly the matrix Θ_r nor the blocks $\mathcal{V}_j^b$.
The result 22 confirms that $G^T \mathbb{V}_m = 0$ as it is shown in 12, and consequently we obtain the following results

$$\begin{aligned} M_\Theta^{-1} B_\Theta &= \mathcal{V}_1^{(1)}, \\ M_\Theta \mathcal{V}_1^{(1)} &= B_\Theta, \\ \Theta_r^T M \Theta_r \mathcal{V}_1^{(1)} &= \Theta_r B, \\ \Pi M \Pi^T V_1^{(1)} &= \Pi B, \\ \Pi (M V_1^{(1)} - B) &= 0, \\ (M V_1^{(1)} - B) &\in null(\Pi) = range(G). \end{aligned}$$

Therefore, the first n_b columns $V_1^{(1)}$ of V_1 can be computed by solving the following saddle point problem

$$\begin{bmatrix} M & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} V_1^{(1)} \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

The same process can be done to get $V_1^{(2)}$ using the matrix equation

$$\mathcal{V}_1^{(2)} (M_\Theta^{-1} A_\Theta)^{-1} M_\Theta^{-1} B_\Theta.$$

To get the first n_b column $V_{j+1}^{(1)}$ of V_{j+1}, we follow the following process

$$\begin{aligned} (M_\Theta^{-1} A_\Theta)\mathcal{V}_j^{(1)} &= \mathcal{V}_{j+1}^{(1)}, \\ M_\Theta \mathcal{V}_{j+1}^{(1)} &= A_\Theta \mathcal{V}_j^{(1)}, \\ \Theta_r^T M \Theta_r \mathcal{V}_{j+1}^{(1)} &= \Theta_r^T A \Theta_r \mathcal{V}_j^{(1)}, \\ \Pi M \Pi^T V_{j+1}^{(1)} &= \Pi A V_j^{(1)}, \\ \Pi(M V_{j+1}^{(1)} - A V_j^{(1)}) &= 0, \\ (M V_{j+1}^{(1)} - A V_j^{(1)}) &\in null(\Pi) = range(G). \end{aligned}$$

Then we have to solve the following saddle point problem

$$\begin{bmatrix} M & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} V_{j+1}^{(1)} \\ \star \end{bmatrix} = \begin{bmatrix} A V_j^{(1)} \\ 0 \end{bmatrix}.$$

In a same manner, we can compute the last n_b column $V_{j+1}^{(2)}$ of V_{j+1}.

After showing how to compute the blocks of the matrix $\mathbb{V}_m$ without computing neither the matrix $\mathcal{V}_m$ corresponding to the orthonormal basis of $\mathbb{K}_m^{ext}(M_\Theta^{-1} A_\Theta, M_\Theta^{-1} B_\Theta)$ nor Θ_r, we can now present the new derived extended block Arnoldi algorithm based only on the sparse system matrices of the index-2 system. Here, we have to mention that this algorithm is based on a Gram-Shmidt orthogonalization process, which reconstructs the blocks $\{V_1, \ldots, V_m\}$, such that their columns form an orthonormal matrix $\mathbb{V}_m$. This matrix is the key to obtain an efficient reduced system to the index-2 original one 7. Details are given in the next sections.We summarize all these results in the following algorithm.

Algorithm 2 The extended block Arnoldi algorithm associated to the index-2 system

Inputs: $M \in \mathbb{R}^{n_v \times n_v}$, $A \in \mathbb{R}^{n_v \times n_v}$, $G \in \mathbb{R}^{n_v \times n_p}$, $B \in \mathbb{R}^{n_v \times n_b}$ and m.

1. Solve the first sadlle point problem

$$\begin{bmatrix} M & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} V_1^{(1)} \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} V_1^{(2)} \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix}.$$

2. Compute $[V_1, \Lambda] = QR\left([V_1^{(1)}, V_1^{(2)}]\right)$ and set $\mathbb{V}_1 = [V_1]$.

3. For $j = 1, \ldots, m$

   a. Set $V_j^{(1)}$: the first n_b columns of V_j and $V_j^{(2)}$ the second n_b columns of V_j.

   b. $\widetilde{V}_{j+1} = \left(\begin{bmatrix} M & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} V_{j+1}^{(1)} \\ \star \end{bmatrix} = \begin{bmatrix} A\, V_j^{(1)} \\ 0 \end{bmatrix}, \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} V_{j+1}^{(2)} \\ \star \end{bmatrix} = \begin{bmatrix} M\, V_j^{(2)} \\ 0 \end{bmatrix} \right).$

   c. Orthogonalize $\widetilde{V}_{j+1}$ with respect to $\mathbb{V}_1, \ldots, \mathbb{V}_j$ to get $\mathbb{V}_{j+1}$, i.e.,

      for $i = 1, 2, \ldots, j$

      $H_{i,j} = (V_i)^T\, \widetilde{V}_{j+1}$.

      $\widetilde{V}_{j+1} = \widetilde{V}_{j+1} - V_i\, H_{i,j}$.

      end for

   d. $[V_{j+1},\ H_{j+1,j}] = QR(\widetilde{V}_{j+1})$.

   e. $\mathbb{V}_{j+1} = [\mathbb{V}_j,\ V_{j+1}]$.

   End For.

As we noticed earlier, the main step in Algorithm 2 is the solution of saddle-point problems of dimension $n_v + n_p$. To solve these problems in our numerical tests, we used LU factorisation from MATLAB. A second way for solving those sparse saddle-point could be done by applying an iterative solver with a preconditioner. Similar algebraic relations to those given by 17 could be stated using only the sparse matrices (M, A) and the matrix $\mathbb{V}_m$ generated by Algorithm 1. We present this result in the following proposition

Proposition 2. *Let $\mathbb{V}_m \in \mathbb{R}^{n_v \times 2mn_b}$ be an orthonormal matrix generated by Algorithm1 and*

$\overline{\mathbb{T}}_m \in \mathbb{R}^{2(m+1)n_b \times 2mn_b}$ *an upper block Hessenberg matrix. Then we have the next result*

$$\begin{aligned} M^{-1}\Pi\, A\mathbb{V}_m &= \mathbb{V}_{m+1}\,\overline{\mathbb{T}}_m \\ &= \mathbb{V}_m \mathbb{T}_m + V_{m+1} T_{m+1} E_m^T. \end{aligned}$$

Proof 2. *Multiplying from the left 17 by* Θ_r*, and using the fact that* $\mathbb{V}_m = \Theta_r \mathcal{V}_m$*, we get*

$$\Theta_r M_\Theta^{-1} \Theta_r^T A \Theta_r\, \mathcal{V}_m = \Theta_r \mathcal{V}_{m+1}\, \overline{\mathbb{T}}_m, \tag{23}$$

$$\Theta_r M_\Theta^{-1} \Theta_r^T A \mathbb{V}_m = \mathbb{V}_{m+1}\, \overline{\mathbb{T}}_m. \tag{24}$$

On the other hand, $\Theta_r M_\Theta^{-1} \Theta_r^T = M^{-1}\Pi$*, in fact*
firstly, it is easy to prove that $\Pi M \Theta_r = M \Theta_r$ *by using the* Θ*-decomposition and also the fact that* $\Pi M = M \Pi^T,\ \Theta_l^T \Theta_r = I_{n_v - n_p}$.
Then,

$$\begin{aligned} \Pi M \Theta_r &= M \Theta_r, \\ \Theta_l \Theta_r^T M \Theta_r &= M \Theta_r, \\ \Theta_l M_\Theta &= M \Theta_r, \\ M^{-1} \Theta_l &= \Theta_r M_\Theta^{-1}, \\ \Theta_r M_\Theta^{-1} \Theta_r^T &= M^{-1}\Pi, \end{aligned}$$

we replace this final result in the formula 24, then we get the desired result

$$\begin{aligned} M^{-1}\Pi\, A\mathbb{V}_m &= \mathbb{V}_{m+1}\,\overline{\mathbb{T}}_m \\ &= \mathbb{V}_m \mathbb{T}_m + V_{m+1} T_{m+1} E_m^T. \end{aligned}$$

The upper block Hessenberg matrix $\overline{\mathbb{T}}_m$ *is defines as*

$$\overline{\mathbb{T}}_m = \begin{bmatrix} \mathbb{T}_m \\ T_{m+1} E_m^T \end{bmatrix} \in \mathbb{R}^{2(m+1)n_b \times 2mn_b},$$

where $\mathbb{T}_m = \mathbb{V}_m^T M^{-1} \Pi A \mathbb{V}_m \in \mathbb{R}^{2mn_b \times 2mn_b}$.

Which ends the proof.

A result has been proven by Simoncini to calculate $\mathbb{T}_m$ directly from the columns of the upper block Hessenberg matrix $\mathbb{H}_m$ generated by Algorithm 2, without computing neither the inverse of M nor the projection matrix Π, for more details see [19].
Notice that from step 1 of Algorithm 2 we have

$$[V_1, \Lambda] = \mathtt{qr}\left([v, w]\right), \tag{25}$$

where $\Lambda \in \mathbb{R}^{2n_b \times 2n_b}$ is an upper triangular matrix defined by

$$\Lambda = \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} \\ 0 & \Lambda^{(2,2)} \end{bmatrix},$$

and v, w are the solutions of the following saddle point problems

$$\begin{bmatrix} M & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} v \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad \begin{bmatrix} A & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} w \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix},$$

we know that

$$\begin{bmatrix} M & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} v \\ \star \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \Leftrightarrow \Pi M \Pi^T v = \Pi B,$$

and from 25 we find that

$$[v, w] = [V_1^{(1)}, V_1^{(2)}] \begin{bmatrix} \Lambda^{(1,1)} & \Lambda^{(1,2)} \\ 0 & \Lambda^{(2,2)} \end{bmatrix},$$

thus

$$v = V_1^{(1)} \Lambda^{(1,1)},$$

and then we have the following result

$$\mathbb{V}_m^T M^{-1} \Pi B = \mathbb{V}_m^T V_1^{(1)} \Lambda^{(1,1)} = \begin{bmatrix} I_{n_b} \\ 0_{n_b} \\ \vdots \\ 0_{n_b} \end{bmatrix} \Lambda^{(1,1)}. \tag{26}$$

We have mentioned before that in order to reduce the original system (7), we can construct a reduced system from the one that we call it Θ system (14) since they realize the same transfer function as it is shown in 15. At the iteration m, we approach $\tilde{\mathbf{v}}(t)$ by $\mathcal{V}_m \hat{v}(t)$ where $\mathcal{V}_m$ is a matrix corresponding to the orthonormal basis of $\mathbb{K}_m^{ext}(M_\Theta^{-1} A_\Theta, M_\Theta^{-1} B_\Theta)$. By injecting the approximation of $\tilde{\mathbf{v}}(t)$ in the system (14) and enforcing the Petrov-Galerkin condition, we get the following reduced system

$$\begin{cases} \dot{\hat{v}}(t) &= \mathcal{V}_m^T M_\Theta^{-1} A_\Theta \, \mathcal{V}_m \, \hat{v}(t) + \mathcal{V}_m^T M_\Theta^{-1} B_\Theta \, u(t), \\ y_m(t) &= C_\Theta \mathcal{V}_m \, \hat{v}(t). \end{cases} \tag{27}$$

We know that $\mathbb{T}_m = \mathcal{V}_m^T M_\Theta^{-1} A_\Theta \, \mathcal{V}_m$ which can be computed only from the upper block Hessenberg matrix $\mathbb{H}_m$ generated by Algorithm 2 as we mentioned before, also $C_\Theta \mathcal{V}_m = C \Theta_r \mathcal{V}_m = C \mathbb{V}_m$, and by using the fact that $M_\Theta^{-1} B_\Theta \in \mathbb{K}_m^{ext}(M_\Theta^{-1} A_\Theta, M_\Theta^{-1} B_\Theta)$ which confirms that $\mathcal{V}_m \mathcal{V}_m^T M_\Theta^{-1} B_\Theta = M_\Theta^{-1} B_\Theta$, then we can prove

$$\mathcal{V}_m^T M_\Theta^{-1} B_\Theta = \mathbb{V}_m^T M^{-1} \Pi B,$$

finally, we get the reduced system described as follows

$$\begin{cases} \dot{\hat{v}}(t) & = & \mathbb{T}_m\,\hat{v}(t) + \mathbb{B}_m\,u(t), \\ y_m(t) & = & \mathbb{C}_m\,\hat{v}(t), \end{cases} \tag{28}$$

where $\mathbb{B}_m = \left[I_{n_b},\, 0_{n_b}, \ldots, 0_{n_b}\right]^T \Lambda^{(1,1)} \in \mathbb{R}^{2mn_b \times n_b}$ as it is mentioned in 26, and $\mathbb{C}_m = C\mathbb{V}_m \in \mathbb{R}^{n_c \times 2mn_b}$.

The reduced transfer function is given by

$$F_m(s) = \mathbb{C}_m(sI_{2mn_b} - \mathbb{T}_m)^{-1}\mathbb{B}_m.$$

Another way to construct a reduced system is by considering the Θ-system 14 without inverting the matrix M_Θ. We again approximate $\tilde{\mathbf{v}}(t)$ by $\mathcal{V}_m\hat{v}(t)$ where $\mathcal{V}_m$ is a matrix described in the previous sections, then we get the following system

$$\begin{cases} \mathcal{V}_m^T\,M_\Theta\mathcal{V}_m\dot{\hat{v}}(t) & = & \mathcal{V}_m^T\,A_\Theta\,\mathcal{V}_m\,\hat{v}(t) + \mathcal{V}_m^T B_\Theta\,u(t), \\ y_m(t) & = & C_\Theta\mathcal{V}_m\,\hat{v}(t), \end{cases} \tag{29}$$

using the fact that $\mathbb{V}_m = \Theta_r\mathcal{V}_m$, we get the following reduced system

$$\begin{cases} \mathbb{M}_m\dot{\hat{v}}(t) & = & \mathbb{A}_m\,\hat{v}(t) + \mathbb{B}_m\,u(t), \\ y_m(t) & = & \mathbb{C}_m\,\hat{v}(t), \end{cases} \tag{30}$$

with the associated reduced transfer function

$$F_m(s) = \mathbb{C}_m(s\mathbb{M}_m - \mathbb{A}_m)^{-1}\mathbb{B}_m,$$

where $\mathbb{M}_m = \mathbb{V}_m^T M\mathbb{V}_m$, $\mathbb{A}_m = \mathbb{V}_m^T A\mathbb{V}_m$, $\mathbb{B}_m = \mathbb{V}_m^T B$ and $\mathbb{C}_m = C\mathbb{V}_m$. In Algorithm 2 we gave a description of the process to get the matrix $\mathbb{V}_m$ without any explicit computation of $\mathcal{V}_m$ or the matrix Θ_r.

Remark 2. *The obtained reduced systems (28) and (30) are considered as an efficient reduced systems to the original one represented by the index-2 system (8). Numerically, the first reduced system (28) is more required since its system matrices ($\mathbb{T}_m\,\mathbb{B}_m$, $\mathbb{C}_m$) could be computed in a appropriate manner and without requiring matrix-vector products with A and M which is the case for the second reduced system (30) represented by the system matrices ($\mathbb{M}_m, \mathbb{A}_m, \mathbb{B}_m$, $\mathbb{C}_m$).*

4. Numerical examples

In this section, we present some numerical tests showcasing the performance of our method compared to others known methods. We show the effectiveness of the proposed model reduction method based on the extended block Krylov subspace for the DAE system (7), that arises after a

spatial discretization of Stokes equations. A comparison of the proposed method with other known methods such as the classical block Krylov subspace and interpolatory projection methods is also depicted in this section. All the examples were carried out using MATLAB on a computer with Intel ® core i7 at 2.3GHz and 8Gb of RAM. All the data was provided from [14]. In Table 1, we present the different dimensions of n_v which is the dimension of the discretized velocity field, and n_p is the dimension of the discretized pressure field.

Table 1: The Matrix dimensions

Level of dicritization	n_v	n_p	full model $(n_v + n_p)$
1	4796	672	5468
2	12292	1650	13942
3	28914	3784	32698

For different levels, we use matrices $B \in \mathbb{R}^{n_v \times 2}$ and $C \in \mathbb{R}^{6 \times n_v}$. Note that the norm used here is the $\mathcal{H}_\infty$ norm defined by

$$\|F - F_m\|_\infty = \sup_{\omega \in \mathbb{R}} \|F(j\omega) - F_m(j\omega)\|_2. \tag{31}$$

To compute this norm we use the following functions from the library `lyapack` [21]

1. `lp_lgfrq` : Generates a set of logarithmically distributed frequency sampling points $\omega \in [10^{-5}, 10^5]$.
2. `lp_gnorm` : Computes a vector which contains the 2-norm $\|F - F_m\| = \sigma_{max}(F(i\omega) - F_m(i\omega))$ for each sampling points $\omega \in [10^{-5}, 10^5]$, $i = \sqrt{-1}$ and σ_{max} denotes the maximum singular value.

Remark 3. *Turbulent flow occurs at high Reynolds number and it is usually modelled by the Navier-Stokes equations. In our numerical examples, we used fourth different low Reynolds numbers (100, 200, 300 and 500) since we are interested in the Stokes equations that describes a laminar flow regime.*

Example 1. In the first example, we present the frequency response of the original system (Level 3) and the frequency response of the reduced system with $m = 80$ and two different Reynolds numbers Re=100 in Figure 1 and Re=500 in Figure 2. In the left sides of Figures 1 and 2 we plotted the norms $\|F(j\omega)\|_2$ and its approximation $\|F_m(j\omega)\|_2$ for different values of the frequency

$\omega \in [10^{-5}, 10^{5}]$. As can be seen from Figures 1 and 2, we have obtained a perfect match between the original transfer function and its approximation. In the right sides of Figure 1 and Figure 2, we plotted the obtained error-norms for different values of the frequency ω with the Reynold numbers Re=100 and Re=500. Let us remark here that one of the problems in using projection Krylov-based subspace methods is that we do not have a good way of selecting the best dimension of the projected subspace. In our examples, we used an heuristic selection for the value of m.

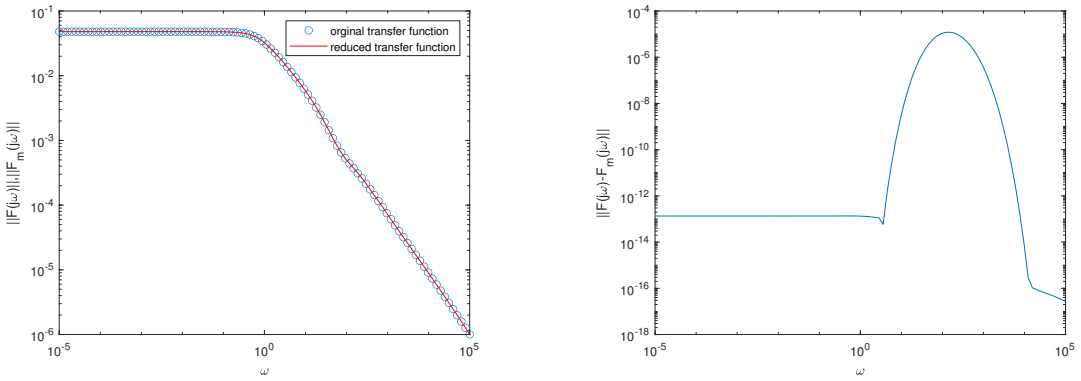

Figure 1: Bode plot (left) and the error-norms versus frequencies (right) with Re=100.

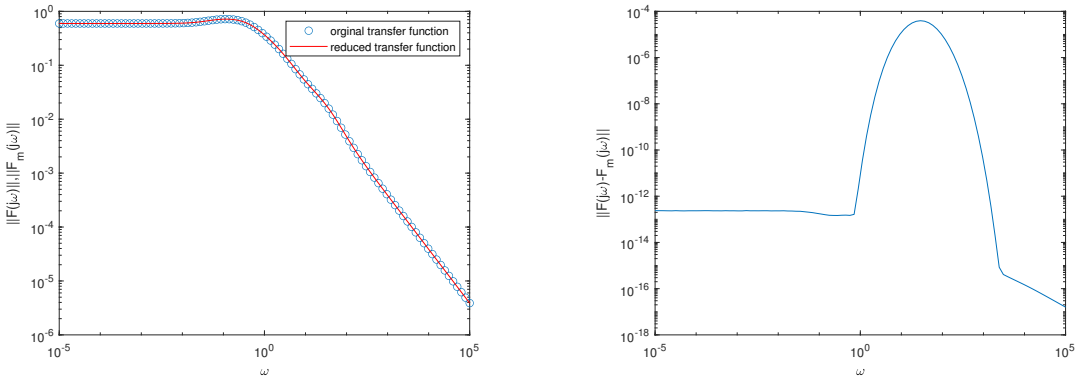

Figure 2: Bode plot (Left) and the error-norms versus frequencies (Right) with Re=500.

In Figure 3, we show a comparison between our method and the so-called classical block Krylov subspace method (CBKSM) described in [22]. We set $m = 30$, then the dimension of the extended block Krylov subspace is $2m$ and to establish a fair comparison, we choose $m = 60$ as a dimension of the classical block Krylov subspace. The errors for our method and the CBKSM computed by formula (31) are, respectively, $\mathbf{6.17}\times\mathbf{1e}^{-06}$ and $\mathbf{6.16}\times\mathbf{1e}^{-02}$.

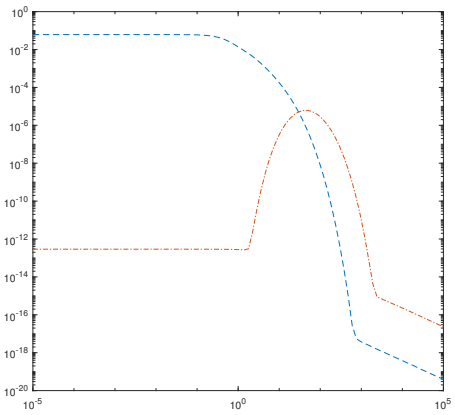

Figure 3: The obtained error-norms $\|F(j\omega) - F_m(j\omega)\|_2$ for different values of the frequency $\omega \in [10^{-5}, 10^5]$ of our method (dash-dotted line) and the CBKSM (dashed line) with Re=100.

We have to mentioned here that we did not use a rational Krylov projection methods because of the difficulty in finding the appropriate poles $\mathbf{s_i}$ that characterize the rational Krylov subspace.

Example 2. In the second example, we compared our proposed method with the one based on an interpolatory projection method; see [23]. The authors established a bi-tangential Hermite interpolation for index-2 descriptor systems via Iterative Rational Krylov Algorithm (IRKA in short). For both methods, we used $m = 30$. The quality of the interpolatory method via IRKA depends on the number of cycles, which means the number of times the interpolation points are updated. We considered here a number of cycles equal to 10. Notice that when we tried a number of cycles more than 10, IRKA becomes expensive. In this example we used the matrices from Level 1 of discretization. In table 2, we reported the $\mathcal{H}_\infty$ error norm and the required cpu-time for our proposed method and IRKA . As can be seen from this table, the extended Krylov-based method returns the best results in times of accuracy and cpu-time.

Table 2: The calculation time and the Err-$\mathcal{H}_\infty$

Reynolds numbers	Extended Krylov-based method		IRKA	
	cpu-time	Err-$\mathcal{H}_\infty$	cpu-time	Err-$\mathcal{H}_\infty$
Re=100	**3.69sec**	6.17×10^{-06}	**37.77 sec**	3.77×10^{-04}
Re=200	**4.16sec**	5.09×10^{-06}	**38.12 sec**	2.2×10^{-05}
Re=300	**3.72sec**	5.86×10^{-06}	**38.11sec**	2.34×10^{-05}

In Figure 4, we showed the error-norm $\|F - F_m\|_\infty$ between the original frequency response and its approximation of both methods with a Reynold number Re=300. In Table 3, we reported the

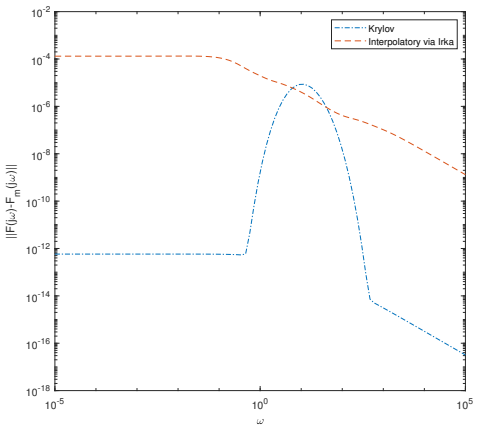

Figure 4: Error-norms versus frequencies

results comparing our method and IRKA after the 1st cycle, also in Figures 5, 6 and 7, we present the error-norms of both methods with three different Reynolds number. All the figures 5, 6 and 7, clearly show that the proposed method has good accuracy compared to the IRKA method.

Table 3: The calculation time and the $\mathcal{H}_\infty$-error

Reynolds numbers	Extended Krylov-based method		IRKA	
	cpu-time	Err-$\mathcal{H}_\infty$	cpu-time	Err-$\mathcal{H}_\infty$
Re=100	**3.69sec**	6.17×10^{-06}	**6.71sec**	1.38×10^{-05}
Re=200	**4.16sec**	5.09×10^{-06}	**6.9sec**	1.45×10^{-05}
Re=300	**3.72sec**	5.86×10^{-06}	**7.13sec**	2.41×10^{-05}

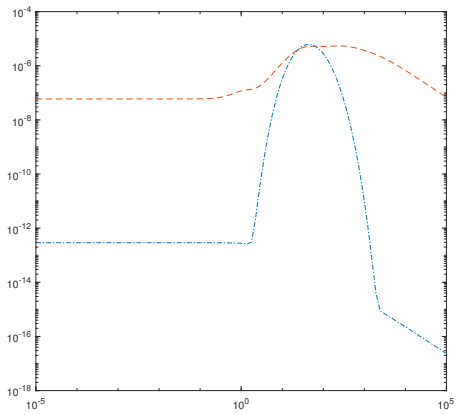

Figure 5: The error-norms of both methods: Krylov(blue dash-dot line) and IRKA(red dash line) with Re=100.

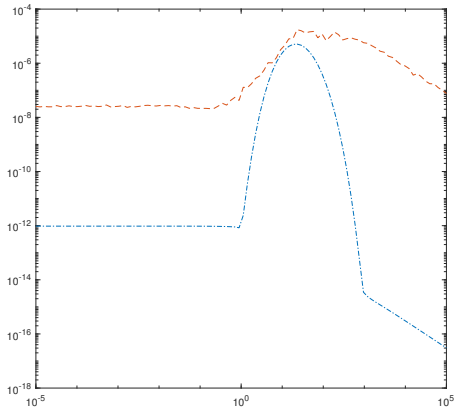

Figure 6: The error-norms of both methods: Krylov(blue dash-dot line) and IRKA(red dash line) with Re=200.

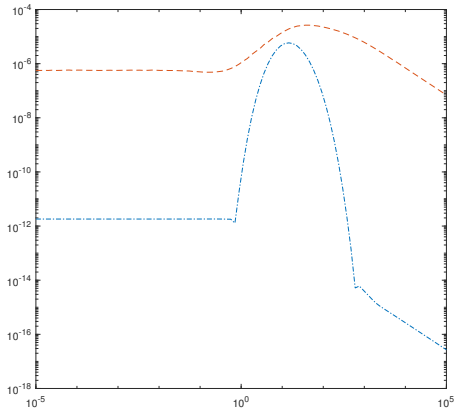

Figure 7: The error-norms of both methods: Krylov(blue dash-dot line) and IRKA(red dash line) with Re=300.

Example 3. In this example, we display a time domain simulation resulting from fourth different input selections. We use the matrices of Level 2 of dicretization and we set $m = 40$. In the left sides of Figures 8–11, we plotted the outputs for the following input selections $\mathbf{u}_i(t) = 0.5\exp(-t)(2 + sin(2i\pi t))$ in Figure 8, $\mathbf{u}_i(t) = sin(6it)$ in Figure 9, $\mathbf{u}_i(t) = \sin(it)$ in Figure 10 and $\mathbf{u}_i(t) = \exp(-t)(sin(\pi it))$ in Figure 11 for $(i = 1, 2)$ since we have two inputs ($B \in \mathbb{R}^{n_v \times \mathbf{2}}$). As you notice all the figures 8–11 illustrate a good accuracy of the reduced output compared to the original one. Error in the outputs for the same input selections given above are presented on the right sides of Figures 8–11.

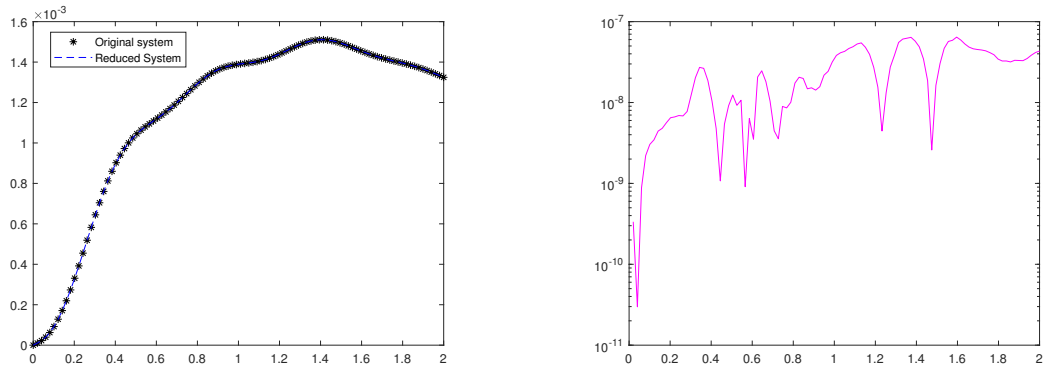

Figure 8: Left: time domain response (output 1 for input 1 with $u_1(t) = 0.5\exp(-t)(2 + sin(2\pi t))$). Right: the error-norms $\|y - y_m\|_2$.

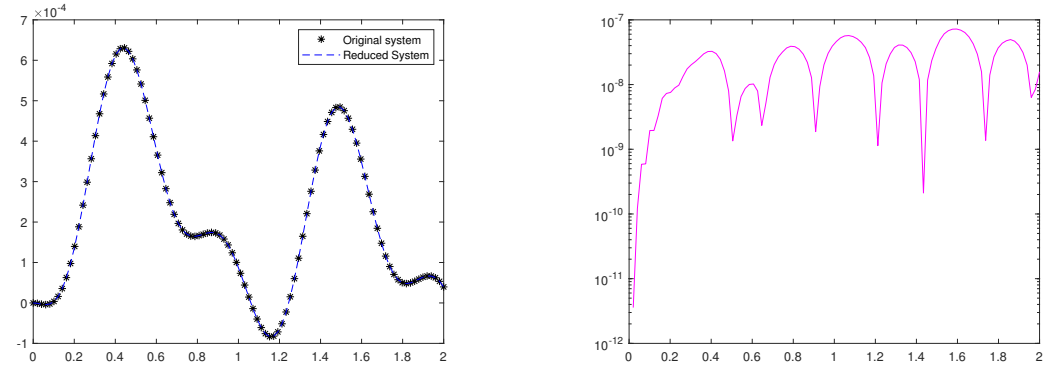

Figure 9: Left: time domain response (output 2 for input 2 with $u_2(t) = \sin(12t)$). Right: the error-norms $\|y - y_m\|_2$.

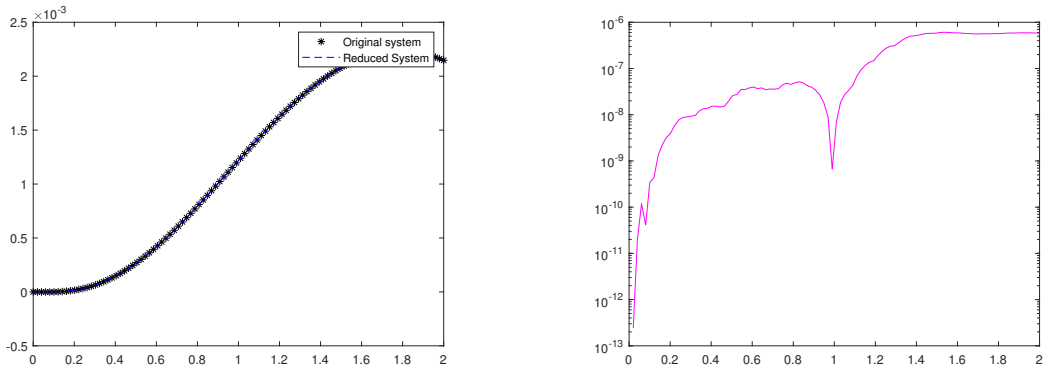

Figure 10: Left: time domain response (output 1 for input 1 with $u_1(t) = \sin(t)$). Right: the error-norms $\|y - y_m\|_2$.

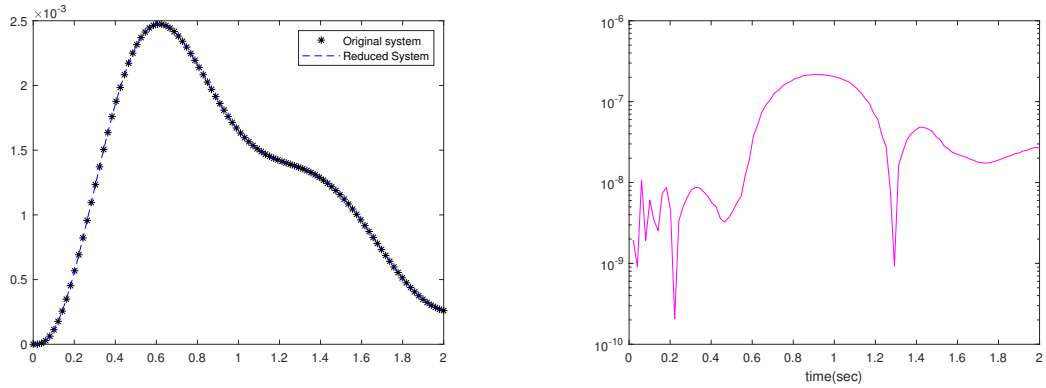

Figure 11: Left: time domain response (output 2 for input 2 with $u_2(t) = e^{-t}sin(2\pi t)$). Right: the error-norms $\|y - y_m\|_2$.

Example 4. For the last example, we showed a visualisation test using two known benchmark examples, namely *driven cavity* and *cylinder wake*. See [24] for more details about these benchmark examples. As a first test, we considered a *driven cavity* on the unit square $\Omega = (0, 1)^2$ while for the second test, we considered the *cylinder wake* on the domain as illustrated in Figure 12. All the examples are modelled by the following Stokes equations as described in Section 2.

$$\frac{\partial v}{\partial t} - \frac{1}{\mathrm{Re}}\Delta v + \nabla p = 0, \tag{32a}$$

$$\nabla \cdot v = 0, \tag{32b}$$

$$v|_\Gamma = g, \tag{32c}$$

$$v|_{t=0} = v_0, \tag{32d}$$

where Re is the Reynolds number and v_0 is an initial condition. The boundary conditions that

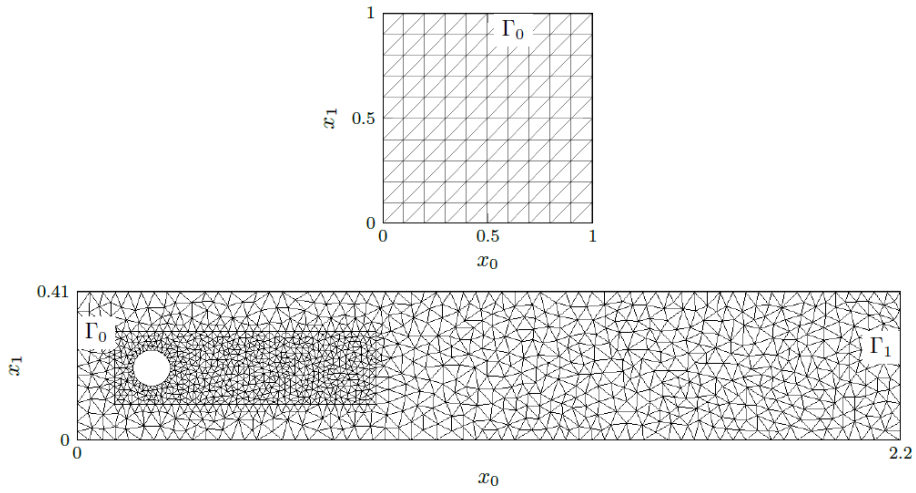

Figure 12: An illustration of the computational domains. Top: driven cavity flow, Bottom: flow around a cylinder.

model the *driven cavity* example is described as follows

$$g : \begin{cases} v &= [1,0]^T & \text{on } \Gamma_0, \\ v &= [0,0]^T & \text{elsewhere.} \end{cases}$$

For the *cylinder wake* example, we prescribe at the inflow Γ_0 a parabolic velocity profile through the function

$$h(s) = 4\left(1 - \frac{s}{0.41}\right)\frac{s}{0.41}.$$

At the Γ_1 outflow a *do-nothing* condition is imposed, as well as *no-slip* condition i.e., zero Dirchlet conditions at the upper and lower wall of the domain and at the cylinder periphery.

$$g : \begin{cases} v &= [h(x_1),0]^T & \text{on } \Gamma_0, \\ pn - \frac{1}{\mathrm{Re}}\frac{\partial v}{\partial n} &= [0,0]^T & \text{on } \Gamma_1, \\ v &= [0,0]^T & \text{elsewhere.} \end{cases}$$

For both examples, we considered Re=50 and aim to determine a reduced system via our proposed method describe by Algorithm 2. The dimension of the reduced systems is $m = 30$ and the dimension of the state space of the first test is $n_v + n_p = 4796 + 672 = 5468$ and for the second test $n_v + n_p = 3042 + 441 = 3483$. The data of the second test was provided from [11] with $B \in \mathbb{R}^{n_v \times 6}$ and $C \in \mathbb{R}^{8 \times n_v}$. A visual comparison of the velocity computed via the full and the reduced system is performed. One can notice that for both tests, the reduced system captures the dynamic of the full system with a good precision.

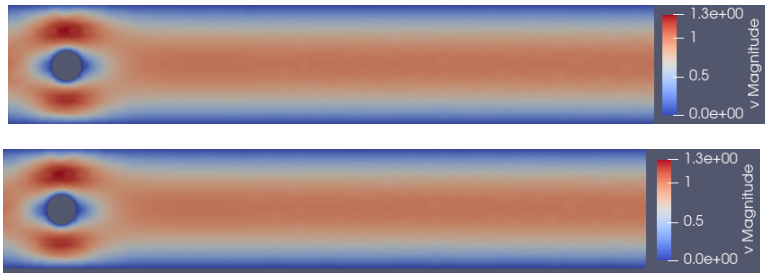

Figure 13: Cylinder wake : The $|v|$ magnitude obtained from the full system (top) and $|v_m|$ obtained from the reduced system (bottom). The error $\|v - v_m\| = 5.57 \times 1e^{-05}$.

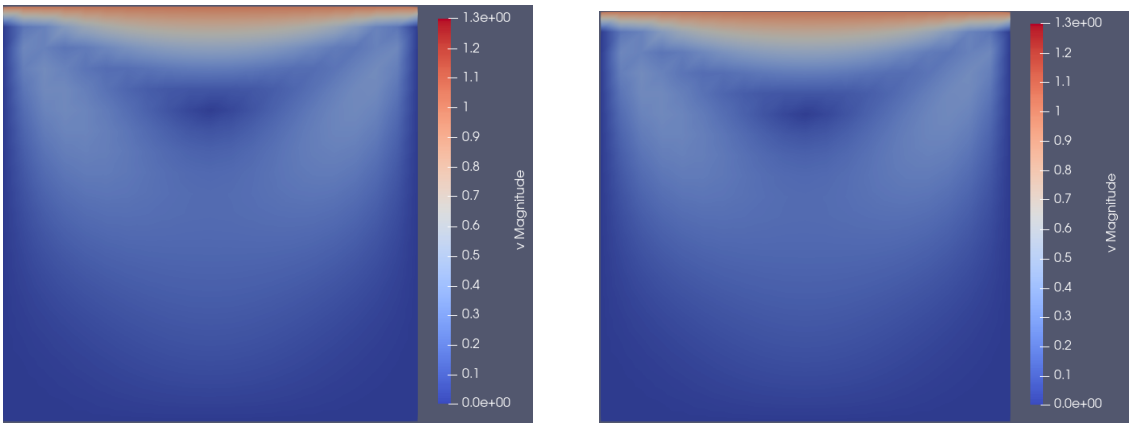

Figure 14: Lid driven cavity : The $|v|$ magnitude obtained from the full system (left) and $|v_m|$ obtained from the reduced system (right). The error $\|v - v_m\| = 3.62 \times 1e^{-06}$.

Conclusion

In this paper, we proposed some projection methods based on extended Krylov subspaces to get reduced order multiple input and multiple output large-scale dynamical systems obtained by a discretization in space of Stokes equations. After a spacial discretization, we obtained differential algebraic equations (DAEs) of index-2. Before projecting, we used a technique to transform the initial DAE system (7) to an ODE system (14). We gave a strategy to avoid dense computations, while performing the model reduction through an extended block Krylov algorithm. The given numerical results show the performance of the proposed approach for Stokes equations.

References

[1] H. Barkouki, A.-H. Bentbib, K. Jbilou, An extended nonsymmetric block lanczos method for model reduction in large scale dynamical systems, Calcolo., 55(1) (2018) 1–23.

[2] A.-H. Bentbib, K. Jbilou, Y. Kaouane, A computational global tangential krylov subspace method for model reduction of large-scale mimo dynamical systems, J. Sci. Comput., 59 (2017) 1–19.

[3] M.-A. Hamadi, K. Jbilou, A. Ratnani, Model reduction method in large scale dynamical systems using an extended-rational block arnoldi method, J. Appl. Math. Comput., (2021) 1–23.

[4] M. Frangos, I.-M. Jaimoukha, Adaptive rational krylov algorithms for model reduction, In Proceedings of the European Control Conference, (2007) 4179–4186.

[5] K. Gallivan, E. Grimme, P. Van-Dooren, Adaptive rational krylov algorithms for model reduction, Numer. Algo., 12 (1996) 33–63.

[6] E. Grimme, D. Sorensen, P. Van-Dooren, Model reduction of state space systems via an implicitly restarted lanczos method, Numer. Algo., 12 (1996) 1–32.

[7] S. Gugercin, A.-C. Antoulas, A survey of model reduction by balanced truncation and some new results, Internat J. Contr., 77(8) (2003) 748–766.

[8] B.-C. Moore, Principal component analysis in linear systems: controllability, observability and model reduction, IEEE Trans. Auto. Contr., AC-26 (1981) 17–32.

[9] O. Abidi, K. Jbilou, Balanced truncation-rational krylov methods for model reduction in large scale dynamical systems, Comput. Appl. Math., 37(1) (2018) 525–540.

[10] E. Bansch, P. Benner, J. Saak, H.-K. Weichelt, Riccati-based boundary feedback stabilization of incompressible navier-stokes flows, SIAM J. Sci. Comput., 37(2) (2015) A832–A858.

[11] P. Benner, P. Goyal, J. Heiland, I.-P. Duff, Operator inference and physics-informed learning of low-dimensional models for incompressible flows, arXiv:2010.06701v1, (2020).

[12] P. Benner, J. Saak, M.-M. Uddin, Balancing based model reduction for structured index-2 unstable descriptor systems with application to flow control, Numer. Alge. Contr. Opti., 6(1) (2016) 1–20.

[13] M. Heinkenschloss, D.-C. Sorensen, K. Sun, Balanced truncation model reduction for a class of descriptor systems with application to the oseen equations, SIAM J. Sci. Comput., 30(2) (2008) 1038–1063.

[14] P. Benner, J. Saak, M. Stoll, H.-K. Weichelt, Efficient solution of large-scale saddle point systems arising in riccati-based boundary feedback stabilization of incompressible stokes flow, SIAM J. Sci. Comput., 35(5) (2013) S150–S170.

[15] K.-A. Cliffe, T.-J. Garratt, A. Spence, Eigenvalues of block matrices arising from problems in fluid mechanics, SIAM J. Matrix Anal. Appl., 15 (1994) 1310–1318.

[16] U.-M. Ascher, L.-R. Petzold, Computer methods for ordinary differential equations and differential-algebraic equations, SIAM., Philadelphia, PA, (1998) 1–314.

[17] V. Druskin, L. Knizhnerman, Extended krylov subspaces: Approximation of the matrix square root and related functions, SIAM J. Matrix Anal. Appl., 19 (1998) 755–771.

[18] M. Heyouni, K. Jbilou, An extended block arnoldi algorithm for large-scale solutions of the continuous-time algebraic riccati equation, Elect. Trans. Numer. Anal., 33 (2009) 53–62.

[19] V. Simoncini, A new iterative method for solving large-scale lyapunov matrix equations, SIAM J. Sci. Comput., 29(3) (2007) 1268–1288.

[20] E. Eich-Soellner, C. Fuhrer, Numerical methods in multibody dynamics, European Consortium for Mathematics in Industry, B. G. Teubner GmbH., Stuttgart, (1998).

[21] V. Mehrmann, T. Penzl, Benchmark collections in slicot, Technical Report SLWN1998- 5, SLICOT Working Note, ESAT, KU Leuven, K. Mercierlaan 94, Leuven-Heverlee 3100, Belgium, (1998).

[22] M. Heyouni, K. Jbilou, A. Messaoudi, T. Tabaa, Model reduction in large scale mimo dynamical systems via the block lanczos method, Comput. Appl. Math., 27(2) (2008) 211–236.

[23] S. Gugercin, T. Stykel, S. Wyatt, Model reduction of descriptor systems by interpolatory projection methods, SIAM J. Sci. Comput., 35(5) (2013) B1010–B1033.

[24] M. Behr, P. Benner, J. Heiland, Example setups of navier-stokes equations with control and observation: Spatial discretization and representation via linear-quadratic matrix coefficients, arXiv. abs/1707.08711, (2017).