Model Order Reduction

by

Olcay Ciftci

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 13, 2014

Keywords: Model Order Reduction, Krylov Subspaces, POD

Approved by

Yanzhao Cao, Professor of Mathematics and Statistics
Narendra Kumar Govil, Alumni Professor of Mathematics and Statistics
Bertram Zinner, Associate Professor of Mathematics and Statistics

Abstract

Model Order Reduction (MOR) is a wide area, and it has many techniques. In this thesis, we focus on Krylov subspaces method and Proper Orthogonal Decomposition (POD). We show some details about these two methods. A MATLAB code is written for Nonsymmetric Band Lanczos algorithm (Bai Z. (2003)). Also by using triangle elements, a MATLAB code for a numerical example about Surface area is demonstrated.

Acknowledgments

Firstly, I want to thank Professor Yanzhao Cao, the supervisor of this thesis, for the patient guidence, encouragement and advice. Also, I would like to thank Professors Narendra Kumar Govil and Bertram Zinner for becoming my master thesis committee members. I got many ideas from their and my supervisor's classes.

Without financial and academic support it would be really difficult to be a master student in the USA, so I would like to thank the government of Republic of Turkey and Mathematics and Statistics Department of Auburn University for their generous financial support.

Special thanks to my friend Osman Yardimci and his family. As a family friend, they always encouraged and helped me during the years I worked on my master degree.

Last but not the least, I would like to thank my wife and my son who gave me moral support.

Thanks again to all who helped me.

Table of Contents

List of Figures

Chapter 1

Introduction

Model order reduction (MOR) has several definitions which depend on the context. Shortly, the reduced-order modeling problem is to find a mathematical model of a system which has much lower dimension than the original model. By simulating this reduced small system, we can see the main characteristics of the original system, so in numerous areas MOR is an indispensable tool. Modeling, simulation, analysis of integrated circuit components, control theory and image processing are some of the areas which MOR is used frequently. We can see the meaning of model order reduction in the following example. Suppose a system of equations is given

$$\frac{d}{dt}x(t) = Ax(t) + bu(t) \tag{1.1}$$

$$y(t) = c^T x(t) \tag{1.2}$$

This system is a linear single input-single output system with specified input $u(t)$ and output $y(t)$, but we can generalize it to the vector input-vector output system. Here $x(t)$ is an N-dimensional vector, A is an N by N matrix and b and c are N-dimensional vectors. Generally, if it is not denoted specifically, the initial value of $x(t)$ is zero, and the dimension of N is between 10^5 and 10^9. The idea behind MOR is to find another system

$$\frac{d}{dt}\tilde{x}(t) = \tilde{A}\tilde{x}(t) + \tilde{b}u(t) \tag{1.3}$$

$$\tilde{y}(t) = \tilde{c}^T \tilde{x}(t) \tag{1.4}$$

where $\tilde{x}(t)$ has the dimension n, $\tilde{A}$ is n by n matrix $\tilde{b}$ and $\tilde{c}$ are n-dimensional vector and $n << N$. This example is a linear system, but in engineering people usually have to deal with nonlinear systems. Firstly, by linearizing the system and then performing model order reduction we can find reduced models for nonlinear systems.

Now there are many reduction techniques which are used in simulation, modeling and integrated circuit components' analysis. In the eighties and nineties of the last century, some fundamental methods were published. The most known MOR methods are Krylov subspaces, P$\grave{a}$de-via-Lanczos method, Proper Orthogonal Decomposition (POD) and PRIMA. In this thesis, we focus on Krylov subspaces method and Proper Orthogonal Decomposition.

Chapter 2

Dynamical Systems

Differential equations are used in many mathematical models of scientific systems which use time as the independent variable. Economics, biology, physics and engineering are some of the fields in which the systems arise. Generally, we can use a system of differential equations

$$\frac{dx}{dt} = f(x) \tag{2.1}$$

to generate a dynamical system. In this chapter, we show time-invariant linear dynamical systems, Krylov Subspaces and then Pàde and Pàde-type approximants as a technique for reduced-order model. When we use model order reduction, we must consider and keep some important characteristic features of the system. Since passivity and stability are the most important concepts for the systems and process, we also discuss them.

2.1 First-Order Time-Invariant Linear Dynamical System and Transfer Functions

Firstly, we can define state space as the m-dimentional space whose coordinate axes are $x_1, x_2, ..., x_m$. Now, let consider m-input q-output first order time-invariant linear dynamical system. We can give the system

$$E_N \frac{d}{dt} x(t) = A_N x(t) + B_N u(t) \tag{2.2}$$

$$y(t) = C_N^T x(t) + D_N u(t) \tag{2.3}$$

with initial conditions $x(0) = x^{(0)}$ as a form with state-space description. These matrices $E_N, A_N \in \mathbb{R}^{N\times N}$, $B_N \in \mathbb{R}^{N\times m}$, $C_N \in \mathbb{R}^{N\times q}$ and $D_N \in \mathbb{R}^{q\times m}$ are given. Sequently, these vectors $x(t) \in \mathbb{R}^N$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^q$ denotes vector of state variables, vector of input and vector of output where N is the state-space dimension.Also, m and q are the inputs and outputs numbers. If the matrix E_N is nonsingular we say the system is *regular*. If not it is a descriptor system or singular. If the system is regular it can be re-arranged and then it can be written as

$$\frac{d}{dt}x(t) = (E_N^{-1}A_N)x(t) + (E_N^{-1}B_N)u(t) \tag{2.4}$$

$$y(t) = C_N^T x(t) + D_N u(t) \tag{2.5}$$

Generally, descriptor systems arise in the modeling, simulation and analysis of integrated circuit components, especially in circuit interconnect and packaging as linear dynamical system. For this reason, in this thesis we usually use $E_N \in \mathbb{R}^{N\times N}$ as a singular matrix. Now, let assume that $A_N, E_N \in \mathbb{R}^{N\times N}$ be matrices such that the matrix pencil $A_N - sE_N$ is regular, i.e., it is singular only for finitely many values of $s \in \mathbb{C}$. If we apply the Laplace transform to the system (2.2) and (2.3) we find these algebraic equations:

$$sE_N X(s) - E_N x(0) = A_N X(s) + B_N U(s)$$

$$Y(s) = C_N^T X(s) + D_N U(s)$$

where $s \in \mathbb{C}$. Let assume that $x(0) = 0$. Then we obtain

$$sE_N X(s) = A_N X(s) + B_N U(s)$$

$$Y(s) = C_N^T X(s) + D_N U(s)$$

By applying the Laplace transform instead of time domain variables $x(t)$, $u(t)$, and $y(t)$ we find the frequency-domain variables $X(s)$, $U(s)$, and $Y(s)$. Then, if we do the following calculations

$$sE_N X(s) - A_N X(s) = B_N U(s)$$

$$(sE_N - A_N)X(s) = B_N U(s)$$

$$X(s) = (sE_N - A_N)^{-1} B_N U(s)$$

$$Y(s) = C_N^T (sE_N - A_N)^{-1} B_N U(s) + D_N U(s)$$

$$Y(s) = (C_N^T (sE_N - A_N)^{-1} B_N + D_N) U(s)$$

we obtain the function $G(s) := C_N^T (sE_N - A_N)^{-1} B_N + D_N$ where $G : \mathbb{C} \mapsto (\mathbb{C} \cup \infty)^{qxm}$. The function is called transfer function of the system (2.2) and (2.3).

2.2 Model Order Reduction

The main task at model order reduction is to reduce the dimension of the state space vector by considering some important characters such as passivity and stability. Let the following system be a reduced-order model of the system (2.2) and (2.3).

$$\widetilde{E}_n \frac{d}{dt} \widetilde{x}(t) = \widetilde{A}_n \widetilde{x}(t) + \widetilde{B}_n u(t) \tag{2.6}$$

$$y(t) = \widetilde{C}_n^T \widetilde{x}(t) + \widetilde{D}_n u(t) \tag{2.7}$$

where $\widetilde{A}_n, \widetilde{E}_n \in \mathbb{R}^{nxn}$, $\widetilde{B}_n \in \mathbb{R}^{nxm}$, $\widetilde{C}_n \in \mathbb{R}^{nxq}$, $\widetilde{D}_n \in \mathbb{R}^{qxm}$, and $n << N$. Our goal is to find $\widetilde{x}(t)$ which has much smaller dimension than N. To obtain a good approximation of the original input-output system, the following conditions should be satisfied:

- Preservation stability, passivity and the other characteristics of original system
- Obtaining a small approximation error
- Computationally efficient reduction procedure.

2.3 Pàde and Pàde-type Approximation

Pàde and Pàde-type approximation have an important role to define reduced-order models by using the transfer functions.

Let $s_0 \in \mathbb{C}$ be any point which is not a pole of the transfer function $G(s)$. Since s_0 is not a pole of G, the Taylor expansion can be used to redefine the transfer function $G(s)$ as

$$G(s) = \sum_{j=0}^{\infty} H_j (s - s_0)^j \tag{2.8}$$

Here $G(s)$ is redefined about s_0, and the coefficients $H_j \in \mathbb{C}^{q \times m}, j = 0, 1, ...$ are called the moments of G.

If the Taylor expansions of the transfer functions $G(s)$ and $\widetilde{G}(s)$ have many common leading terms then the reduced-order model (2.6) and (2.7) is called n-th Pàde model at the expansion point s_0, i.e.

$$G(s) = \widetilde{G}(s) + \mathcal{O}((s - s_0))^{p(n)} \tag{2.9}$$

where $p(n)$ is as large as possible.

By generating $p(n)$ moments $H_0, H_1, ..., H_{p(n)-1}$ we can compute $\widetilde{G}(s)$, and then we can obtain Pàde models, but obtaining them directly from the moments is ill-conditioned. For this reason, Krylov-subspace techniques are preferred to obtain Pàde models. To use these techniques, instead of the matrices A_N and E_N we use a single matrix K_N, and we rearrange the transfer function $G(s) := C_N^T (sE_N - A_N)^{-1} B_N + D_N$. Now let

$A_N - s_0 E_N = M_1 M_2$, where $M_1 M_2 \in \mathbb{C}^{N \times N}$

Here, although $A_N - s_0 E_N$ can be large it is sparse, so the LU factorization of $A_N - s_0 E_N$ is feasible. For a detailed discussion we refer the reader to BAI, DEVILDE AND FREUND [2003].

Now we can rewrite the transfer function as follows:

$$G(s) := C_N^T (sE_N - A_N)^{-1} B_N + D_N$$

$$= C_N^T(sE_N - s_0E_N + s_0E_N - A_N)^{-1}B_N + D_N$$

$$= C_N^T((s - s_0)E_N + s_0E_N - A_N)^{-1}B_N + D_N$$

$$= -C_N^T(A_N - s_0E_N - (s - s_0)E_N)^{-1}B_N + D_N$$

$$= -C_N^T(M_1M_2 - (s - s_0)E_N)^{-1}B_N + D_N$$

$$= -C_N^T(M_1(I - (s - s_0)M_1^{-1}E_NM_2^{-1})M_2)^{-1}B_N + D_N$$

$$= -C_N^TM_2^{-1}(I - (s - s_0)M_1^{-1}E_NM_2^{-1})^{-1}M_1^{-1}B_N + D_N$$

$$= -L_N^T(I - (s - s_0)K_N)^{-1}R_N + D_N \tag{2.10}$$

where $K_N := M_1^{-1}E_NM_2^{-1} \in \mathbb{C}^{N \times N}$, $R_N := M_1^{-1}B_N \in \mathbb{C}^{N \times m}$ and $L_N := M_2^{-T}C_N \in \mathbb{C}^{N \times q}$. By (2.8) we can apply Krylov-subspace techniques to the single matrix K_N, and also by using R_N and L_N we can generate blocks of right and left starting vectors.

To obtain good approximations in frequency domain, Pàde modes are efficient, but usually they are insufficient to preserve the important properties of linear dynamical systems such as passivity and stability.

2.4 Krylov Subspaces

2.4.1 Block Krylov Subspaces

As we mentioned in the previous section, to use the Krylov subspace techniques, we need to rewrite the original system (2.2) and (2.3) by replacing the matrices A_N and E_N by a single matrix K_N. To obtain new system, let $s_0 \in \mathbb{C}$ be any point such that s_0 is not a pole of the transfer function $G(s)$ and the matrix $A_N - s_0E_N$ is nonsingular. Then the original system can be written as follows:

$$K_N\frac{dx}{dt} = (I + s_0K_N)x(t) + R_Nu(t) \tag{2.11}$$

$$y(t) = L_N^T x(t) + D_N u(t) \tag{2.12}$$

and the transfer function $G(s)$ can be written in the following form by expanding it about s_0.

$$G(s) = D_N - \sum_{j=0}^{\infty} L_N^T K_N^j R_N (s - s_0)^j \tag{2.13}$$

Since we can define the Pàde and Pàde-type reduced order models by using the leading coefficients of $G(s)$ about s_0, we can obtain the following expansion:

$-L_N^T K_N^j R_N = -L_N^T K_N^j K_N^{-i} K_N^i R_N = -((K_N^{j-i})^T L_N)^T (K_N^i R_N), i = 0, 1, ...$ *and* $j = 0, 1, ...$

Here we obtain the block of the left and right block Krylov matrices:

$$[L_N \quad K_N^T L_N \quad (K_N^T)^2 L_N \quad ... \quad (K_N^T)^i L_N \quad ...] \tag{2.14}$$

$$[R_N \quad K_N R_N \quad K_N^2 R_N \quad ... \quad K_N^i R_N \quad ...] \tag{2.15}$$

Although (2.14) and (2.15) contain the all required information, to obtain Pàde and Pàde-type reduced order models, it is not a good approximation to just compute the blocks $K_N^i R_N$ and $(K_N^T)^i L_N$ and generating (2.13). For this reason, we need to apply some other Krylov-subspace methods to obtain numerically better basis vectors.

Let's look at (2.14) and (2.15) to generate feasible Krylov-subspaces. Each block $(K_N^T)^i L_N$ has q column vectors of length N. Now if we scan the column vectors from left to the right, and if we delete the columns which are linearly independent on its left columns we generate a new deflated left Krylov subspace:

$$[L_{N_1} \quad K_N^T L_{N_2} \quad (K_N^T)^2 L_{N_3} \quad ... \quad (K_N^T)^{j_{max}-1} L_{N_{k_{max}}} \quad ...] \tag{2.16}$$

Let $n_{max}^{(l_N)}$ be the number of columns of (2.16). Here $\mathcal{K}_n(K_N^T, L_N)$ is defined as the n-th left Krylov subspace where $1 \leq n \leq n_{max}^{(l_N)}$.

Applying same steps we obtain a deflated right Krylov subspace:

$$[R_{N_1} \quad K_N R_{N_2} \quad K_N^2 R_{N_3} \quad ... \quad K_N^{j_{max}-1} R_{N_{max}} \quad ...] \tag{2.17}$$

Let $n_{max}^{(r_N)}$ be the number of columns of (2.17), and $\mathcal{K}_n(K_N, R_N)$ be defined as the n-th right Krylov subspace where $1 \leq n \leq n_{max}^{(r_N)}$.

For more detailed discussion we refer the reader to BAI, DEWILDE, FREUND [2003], FREUND [2000b] AND ALIAGA ET AL [2000].

2.5 MOR with Lanczos and Lanczos-type Methods

In this section, our goal is to construct appropriate basis vectors for the right and left Krylov subspaces $\mathcal{K}_n(K_N, R_N)$ and $\mathcal{K}_n(K_N^T, L_N)$ to obtain reduced models. We use Lanczos and Lanczos-type methods to generate these basis vectors.

By Pàde via Lanczos (PVL) algorithm in FELDMANN AND FREUND [1994,1995], we can obtain bi-orthogonal basis vectors for the left and right Krylov subspaces which induce single vectors. This is a special case because these left and right Krylov subspaces are obtained from single-input single output linear dynamical systems.

The underlying block Krylov subspace called the nonsymmetric band lanczos algorithm (FREUND [2000a], BAI, DEWILDE, FREUND [2003]). For general m-input q-output time-invariant linear dynamical systems, we can construct two sets of bi-orthogonal right and left Lanczos vectors

$$v_1, v_2, ..., v_n \quad and \quad w_1, w_2, ..., w_n \tag{2.18}$$

These vectors span the n-th right and left Krylov subspaces and satisfy the following property

$$w_i^T v_j = \begin{cases} 0, & \text{if } i = j \\ \delta_i, & \text{if } i \neq j \end{cases} \quad for\ all \quad i, j = 1, 2, ..., n \tag{2.19}$$

The nonsymmetric band Lanczos algorithm produces the matrices $T_n^{(pr)}$, $\rho_n^{(pr)}$, $\eta_n^{(pr)}$ and Δ_n as output. Here $\Delta_n = diag(\delta_1, \delta_2, ..., \delta_n)$ and δ_i's are from (2.19).

The matrices $T_n^{(pr)}$, $\rho_n^{(pr)}$, $\eta_n^{(pr)}$ and Δ_n are used to generate a reduced-order model of the original linear dynamical system (2.2) and (2.3) as follows

$$T_n^{(pr)} \frac{d}{dt} \widetilde{x}(t) = (s_0 T_n^{(pr)} - I)\widetilde{x}(t) + \rho_n^{(pr)} u(t) \tag{2.20}$$

$$y(t) = (\eta_n^{(pr)})^T \Delta_n \widetilde{x}(t) + D_n u(t) \tag{2.21}$$

2.6 Nonsymmetric band Lanczos Algorithm

INPUT: A matrix $K_N \in \mathbb{C}^{N \times N}$;

A block of m right starting vectors $R_N = [r_1 r_2 ... r_m] \in \mathbb{C}^{N \times m}$;

A block of q left starting vectors $L_N = [l_1 l_2 ... l_q] \in \mathbb{C}^{N \times q}$;

OUTPUT: The nxn Lanczos matrix $T_n^{(pr)}$, and the matrices $\rho_n^{(pr)}$, $\eta_n^{(pr)}$, and Δ_n.

0) For k=1,2,...,m, set $\tilde{v}_k = r_k$.

For k=1,2,...,q, set $\tilde{w}_k = l_k$.

Set $m_c = m$, $q_c = q$, and $\mathcal{I}_v = \mathcal{I}_w = \phi$

For $n = 1, 2, ...,$ until converge or $m_c = 0$ or $q_c = 0$ or $\delta_n = 0$ do:

1) (If necessary, deflate $\tilde{v}_n$.)

Compute $||\tilde{v}_n||_2$.

Decide if $\tilde{v}_n$ should be deflated. If yes, do the following :

a) Set $\tilde{v}_{n-m_c}^{defl} = \tilde{v}_n$ and store this vector $\mathcal{I}_v = \mathcal{I}_v \cup \{n - m_c\}$.

b) Set $m_c = m_c - 1$. If $m_c = 0$, set $n = n - 1$ and stop.

c) For $k = n, n+1, ..., n + m_c - 1$, $\tilde{v}_k = \tilde{v}_{k+1}$.

d) Repeat all of Step 1).

2) (If necessary, deflate $\tilde{w}_n$.)

Compute $||\tilde{w}_n||_2$.

Decide if $\tilde{w}_n$ should be deflated. If yes, do the following :

a) Set $\tilde{w}_{n-q_c}^{defl} = \tilde{w}_n$ and store this vector $\mathcal{I}_w = \mathcal{I}_w \cup \{n - q_c\}$.

b) Set $q_c = q_c - 1$. If $q_c = 0$, set $n = n - 1$ and stop.

c) For $k = n, n+1, ..., n + q_c - 1$, $\tilde{w}_k = \tilde{w}_{k+1}$.

d) Repeat all of Step 2).

3) (Normalize $\tilde{v}_n$ and $\tilde{w}_n$ to obtain $v_n$band w_n.)

Set

$$t_{n,n-m_c} = ||\tilde{v}_n||_2, \quad \tilde{t}_{n,n-q_c} = ||\tilde{w}_n||_2,$$

$$v_n = \frac{\tilde{v}_n}{t_{n,n-m_c}}, \quad and \quad w_n = \frac{\tilde{w}_n}{\tilde{t}_{n,n-q_c}}$$

4) (Compute δ_n and check for possible breakdown.)

Set $\delta_n = w_n^T v_n$. If $\delta_n = 0$, set $n = n - 1$ and stop.

5) (Orthogonalize the right candidate vectors against w_n.)

For $k = n + 1, n + 2, ..., n + m_c - 1$, set

$$t_{n,k-m_c} = \frac{w_n^T \tilde{v}_k}{\delta_n} \quad and \quad \tilde{v}_k = \tilde{v}_k - v_n t_{n,k-m_c}.$$

6) (Orthogonalize the left candidate vectors against v_n.)

For $k = n + 1, n + 2, ..., n + q_c - 1$, set

$$\tilde{t}_{n,k-q_c} = \frac{\tilde{w}_k^T v_n}{\delta_n} \quad and \quad \tilde{w}_k = \tilde{w}_k - w_n \tilde{t}_{n,k-q_c}.$$

7) (Advance the right block Krylov subspace to get $\tilde{v}_{n+m_c}$.)

a) Set $\tilde{v}_{n+m_c} = K_N v_n$.

b) For $k \in \mathcal{I}_w$ (in ascending order), set

$$\tilde{\sigma} = (\tilde{w}_k^{defl})^T v_n, \quad \tilde{t}_{n,k} = \frac{\tilde{\sigma}}{\delta_n},$$

and, if $k > 0$, set

$$t_{k,n} = \frac{\tilde{\sigma}}{\delta_k} \quad and \quad \tilde{v}_{n+m_c} = \tilde{v}_{n+m_c} - v_k t_{k,n}.$$

c) Set $k_v = max\{1, n - q_c\}$.

d) For $k = k_v, k_v + 1, ..., n - 1$, set

$$t_{k,n} = \tilde{t}_{n,k} \frac{\delta_n}{\delta_k} \quad and \quad \tilde{v}_{n+m_c} = \tilde{v}_{n+m_c} - v_k t_{k,n}.$$

e) Set

$$t_{n,n} = \frac{w_n^T \tilde{v}_{n+m_c}}{\delta_n} \quad and \quad \tilde{v}_{n+m_c} = \tilde{v}_{n+m_c} - v_n t_{n,n}.$$

8) (Advance the left block Krylov subspace to get $\tilde{w}_{n+q_c}$.)

a) Set $\tilde{w}_{n+q_c} = K_N^T w_n$.

b) For $k \in \mathcal{I}_v$ (in ascending order), set

$$\sigma = w_n^T \tilde{v}_k^{defl}, \quad t_{n,k} = \frac{\sigma}{\delta_n},$$

and, if $k > 0$, set

$$\tilde{t}_{k,n} = \frac{\sigma}{\delta_k} \quad and \quad \tilde{w}_{n+q_c} = \tilde{w}_{n+q_c} - w_k \tilde{t}_{k,n}.$$

c) Set $k_w = max\{1, n - m_c\}$.

d) For $k = k_w, k_w + 1, ..., n - 1$, set

$$\tilde{t}_{k,n} = t_{n,k} \frac{\delta_n}{\delta_k} \quad and \quad \tilde{w}_{n+q_c} = \tilde{w}_{n+q_c} - w_k \tilde{t}_{k,n}.$$

e) Set

$$\tilde{t}_{n,n} = t_{n,n} \quad and \quad \tilde{w}_{n+q_c} = \tilde{w}_{n+q_c} - w_n \tilde{t}_{n,n}.$$

9) Set

$$T_n^{(pr)} = [t_{i,k}]_{i,k=1,2,...,n},$$

$$\rho_n^{(pr)} = [t_{i,k-m}]_{i=1,2,...,n;k=1,2,...,k_\rho} \quad where \quad k_\rho = m + min\{0, n - m_c\},$$

$$\eta_n^{(pr)} = [\tilde{t}_{i,k-q}]_{i=1,2,...,n;k=1,2,...,k_\eta} \quad where \qquad k_\eta = q + min\{0, n - q_c\},$$

$$\Delta_n = diag(\delta_1, \delta_2, ..., \delta_n).$$

10) Check if n is large enough. If yes, stop.

2.7 Nonsymmetric Band Lanczos Method- MATLAB Code

```
function [Tn, Rn, Pn, Dn] = Lanczos(K, R, L, con)
[N,m]=size(R);
[N,q]=size(L);
dtol=10^(−5);
Iw=[ ];
Iv=[ ];
T=zeros(con,con+m);
Td=zeros(con,con+q);
vs=zeros(N,con+m);
ws=zeros(N,con+q);
vsd=zeros(N,con+m);
wsd=zeros(N,con+q);
v=zeros(N,con);
w=zeros(N,con);
del=zeros(1,con);
for k=1:m
```

```
    vs(:,k)=R(:,k);
end
for k=1:q
    ws(:,k)=L(:,k);
end
mc=m;
qc=q;
for n=1:con
  while norm(vs(:,n))<=dtol
      vsd(:,n)=vs(:,n);
          Iv=union(Iv,n-mc);
      mc=mc-1;
      if mc==0
          n=n-1;
          break
      end
      for k=n:n+mc-1
         vs(:,k)=vs(:,k+1);
      end
  end
  while norm(ws(:,n))<=dtol
      wsd(:,n)=ws(:,n);
      Iw=union(Iw,n-qc);
  qc=qc-1;
  if qc==0
      n=n-1;
      break
```

```
      end
      for k=n:n+qc-1
ws(:,k)=ws(:,k+1);
      end
   end
      T(n,n)=norm(vs(:,n));
     Td(n,n)=norm(ws(:,n));
v(:,n)=vs(:,n)/norm(vs(:,n));
w(:,n)=ws(:,n)/norm(ws(:,n));
del(1,n)=dot(v(:,n),w(:,n));
if del(1,n)==0
      n=n-1;
      break
end
for k=n+1:n+mc-1
      T(n,k)=(dot(w(:,n),vs(:,k)))/del(1,n);
      vs(:,k)=vs(:,k)-v(:,n)*(dot(w(:,n),vs(:,k)))/del(1,n);
end
for k=n+1:n+qc-1
      Td(n,k)=(dot(ws(:,k),v(:,n)))/del(1,n);
      ws(:,k)=ws(:,k)-w(:,n)*(dot(ws(:,k),v(:,n)))/del(1,n);
end
vs(:,n+mc)=K*v(:,n);
  [m1,m2]=size(Iw);
for i=1:m2
      Td(n,Iw(i)+q)=(dot(v(:,n),wsd(:,Iw(i)+q)))/del(1,n);
      if Iw(i)>0
```

```
        T(Iw(i)+m,n)=(dot(v(:,n),wsd(:,Iw(i)+q)))/del(1,Iw(i));
    vs(:,n+mc)=vs(:,n+mc)-v(:,Iw(i))*T(Iw(i)+m,n);
    end
end
kv=max(1,n-qc);
if n>1
    for k=kv:n-1
    T(k,n+m)=Td(n,k+q)*(del(1,n)/del(1,k));
    vs(:,n+mc)=vs(:,n+mc)-v(:,k)*T(k,n+m);
    end
end
T(n,n+m)=(dot(w(:,n),vs(:,n+mc)))/del(1,n);
vs(:,n+mc)=vs(:,n+mc)-v(:,n)*T(n,n+m);
ws(:,n+qc)=transpose(K)*w(:,n);
  [n1,n2]=size(Iv);
for i=1:n2
    T(n,Iv(i)+m)=(dot(vsd(:,Iv(i)+m),w(:,n)))/del(1,n);
    if Iv(i)>0
        Td(Iv(i)+q,n)=(dot(vsd(:,Iv(i)+m),w(:,n)))/del(1,Iv(i));
    ws(:,n+qc)=ws(:,n+qc)-w(:,Iv(i))*Td(Iv(i)+q,n);
    end
end
kw=max(1,n-mc);
  if n>1
for k=kw:n-1
    Td(k,n+q)=T(n,k+m)*(del(1,n)/del(1,k));
    ws(:,n+qc)=ws(:,n+qc)-w(:,k)*Td(k,n+q);
```

```
end
  end
Td(n,n+q)=T(n,n+m);
ws(:,n+qc)=ws(:,n+qc)-w(:,n)*Td(n,n+q);
for i=1:n
     for k=1:n
         Tn(i,k)=T(i,k+m);
     end
end
kp=m+min(0,n-mc);
for i=1:n
     for k=1:kp
         Pn(i,k)=T(i,k);
     end
end
kr=q+min(0,n-qc);
for i=1:n
     for k=1:kr
         Rn(i,k)=Td(i,k);
     end
end
for i=1:n
     for k=1:n
         if i==k
            Dn(i,k)=del(1,i);
         else
            Dn(i,k)=0;
```

```
        end
      end
    end
  end
```

Chapter 3

Model Reduction Using Proper Orthogonal Decomposition

3.1 Proper Orthogonal Decomposition

POD is an efficient and a powerful model order reduction method. The main objective in proper orthogonal decomposition is to extract low dimensional basis functions from an ensemble of experimental or detailed simulation data of high dimensional systems. POD provides efficient tools to derive surrogate models for high-dimensional dynamical systems and for many partial differential equations when it is used with Galerkin projection. The reason is that when the dynamical system and PDEs are projected onto a subspace of the original phase space, in combination with Galerkin projection the subspace inherits special characteristics of overall solutions. Although POD is applied to nonlinear problems it requires only standard matrix calculations, and this is a big advantage for us.

(Lumley 1967) introduced POD as an objective of coherent structures. This method is also known as Karhunen-Loeve decomposition, and it was used to study turbulence phenomena within the area of Computational Fluid Dynamics. Also, accurate results were obtained in pattern recognition and signal analysis. Especially, during the last decades it has been used in optimal control of PDEs.

3.2 POD Basis

Let V be a finite (or infinite) dimensional vector space. If we consider a dynamical system which consists of partial differential equations the phase space of an ordinary differential system which we find after a spatial discretization is resembled by V. Although we can choose V as an infinite vector space we will use finite dimensions and set $V = \mathbb{R}^n$. To begin with we need a set of sampled data $W = \{w_1(t), w_2(t), ..., w_M(t)\}$ where $w_j(t) \in \mathbb{R}^n$ $(j = 1, ..., M)$ are trajectories and $t \in [0, T]$. Here, our goal is to find a d-dimensional subspace $V_d \subset V$ by an orthogonal projection $\mathbb{P}_d : V \mapsto V_d$ of fixed rank d which minimizes this equation

$$\|W - \mathbb{P}_d W\|^2 := \sum_{j=1}^{M} \int_0^T \|w_j(t) - \mathbb{P}_d w_j(t)\|^2 dt \tag{3.1}$$

In order to find $\mathbb{P}_d$, let's define a matrix $C \in \mathbb{R}^{n \times n}$. This is the correlation matrix, and the solution of (3.1) depends on C.

$$C = \sum_{j=1}^{M} \int_0^T w_j(t) w_j^*(t) dt \tag{3.2}$$

Here, C is a symmetric positive semi-definite matrix. Its eigenvalues are real and nonnegative ordered, i.e. $\mu_1 \geq \mu_2 \geq ... \geq \mu_n \geq 0$.

Let υ_j be the corresponding eigenvectors given by

$$C\upsilon_j = \mu_j \upsilon_j \qquad j = 1, 2, ...n \tag{3.3}$$

The eigenvectors can be chosen as an orthonormal basis because of structure of C.

Applying POD we obtain the optimal subspace $V_d = span\{\upsilon_1, ..., \upsilon_d\}$ which are called POD modes. The following theorem is efficient to find $\mathbb{P}_d w_i(t)$. For details we refer the reader to SANJAY LALL, PETR KRYSL, JERROLD E. MARSDEN [2003] and RENNE PINNAU [2008].

Theorem 1 : Let the optimal orthogonal projection $\mathbb{P}_d : V \mapsto V_d$ be given by

$$\mathbb{P}_d = \sum_{j=1}^{d} \upsilon_j \upsilon_j^* \quad with \quad \mathbb{P}_d \mathbb{P}_d^* = I \tag{3.4}$$

Each $w_i(t) \in V$ can be shown as

$$w_i(t) = \sum_{j=1}^{n} \upsilon_j^* w_i(t) \upsilon_j \tag{3.5}$$

Then the equation holds

$$\mathbb{P}_d w_i(t) = \sum_{j=1}^{d} \upsilon_j \upsilon_j^* (\sum_{k=1}^{n} \upsilon_k^* w_i(t) \upsilon_k) = \sum_{j=1}^{d} \upsilon_j^* w_i(t) \upsilon_j \tag{3.6}$$

because $\upsilon_j^* \upsilon_i = \delta_{ij}$

3.3 Finding the Optimal Dimension

In order to get an optimal approximation of our data set, we have to select the dimension d of the subspace V_d carefully. Theorem 1 can help us to find d because using it we obtain the overall least-square error which is related with the eigenvalues of C. These eigenvalues can guide us to get d because large eigenvalues give us the main characteristics of a dynamical system. Small ones do not make important changes on the system. For this reason, to find the smallest optimal d we can say that the ratio

$$R(I) = \frac{\sum\limits_{i=1}^{d} \mu_i}{\sum\limits_{i=1}^{n} \mu_i} \quad where \quad \mu_1 \geq \mu_2 \geq ... \geq \mu_n \geq 0 \tag{3.7}$$

must be near to one. Especially in fluid dynamics and heat transfer, the eigenvalues usually decrease exponentially. Therefore, we can obtain low-order approximate models easily.

3.4 Snapshots

There are many methods for deriving low order models of dynamical systems and partial differential equations. POD is one of the most efficient methods, and snapshots take part as an important concept in it. We can think POD as a Galerkin approximation in the spatial variable which is constructed from the trajectories of the dynamical system at prespecified time instances. These trajectories build snapshots, and they may not be appropriate as a basis due to the possibility of linear dependancy. For this reason, an eigenvalue decomposition (EVD) or a singular value decomposition (SVD) is carried out to obtain POD basis. Let the snapshots $w_i = w(t_i)\in\mathbb{R}^n$ be given at the certain time instants $t_1, t_2, ..., t_M\in[0, T]$. Now, we can build a new correlation matrix C which is shown as

$$C = \sum_{j=1}^{M} w_j(t)w_j^*(t)dt \tag{3.8}$$

We should consider the possibility that the snapshots may be linearly dependent. Also, we do not want to get more than n linearly independent vectors, so we should be careful when we choose them. Let $W = (w(t_1), w(t_2), ..., w(t_M))\in\mathbb{R}^{n\times M}$ be a matrix which consists of the dynamical system in its columns and shows the trajectories in its rows. By this matrix, the sum (3.8) can be shown as $C = WW^*\in\mathbb{R}^{n\times n}$ where $M << n$. Finding the eigenvalues of C is computationally expensive, hence, the 'method of snapshots' is designed. Since the eigenvalues are same, $C = W^*W\in\mathbb{R}^{M\times M}$ is used instead of WW^*. Solving the eigenvalue problem (3.9) is easier than before.

$$W^*Wu_j = \mu_j u_j \qquad j = 1, 2, ..., M \qquad u_j\in\mathbb{R}^M \tag{3.9}$$

Now, we can take the eigenvectors $\{u_1, u_2, ..., u_M\}$ as an orthonormal basis, and we can find the POD modes as

$$v_j = \frac{1}{\sqrt{\mu_j}}Wu_j \qquad j = 1, 2, ..., M \tag{3.10}$$

3.5 POD and Singular Value Decomposition

Singular value decomposition (SVD) is a popular technique to obtain dominant characteristics and coherent structures from data. While eigenvalue decomposition is defined only for square matrices, SVD can be used for rectangular matrices. There is a strong connection between POD and SVD for non-square matrices. Now, we compute the SVD of the matrix $W \in \mathbb{R}^{n \times M}$ with rank r. From standard SVD we get these real numbers $\sigma_1 \geq \ldots \geq \sigma_r > 0$ and unitary matrices $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{M \times M}$ such that $W = USV^*$ where $S = \begin{pmatrix} S_d & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n \times M}$ and $S_d = diag(\sigma_1, \ \ldots \ , \sigma_r) \in \mathbb{R}^{r \times r}$. The $\sigma' s$ are called the singular values of W (and also of W^*). They are unique. Here, the number of singular values equals the rank of W.

We call $u_i \in \mathbb{R}^n$ the left singular vectors and $v_i \in \mathbb{R}^M$ the right singular vectors where $U = (u_1, u_2, ..., u_n)$ and $V = (v_1, v_2, ..., v_M)$

Moreover, for $1 \leq i \leq r$

$$Wv_i = \sigma_i u_i \tag{3.11}$$

$$W^* u_i = \sigma_i v_i \tag{3.12}$$

If we multiply both sides of (3.11) and (3.12) respectively by W^* and W we find that

$W^* W v_i = \sigma_i W^* u_i = \sigma_i \sigma_i v_i = \sigma_i^2 v_i$

$W W^* u_i = \sigma_i W v_i = \sigma_i \sigma_i u_i = \sigma_i^2 u_i$

u_i and v_i are the eigenvectors of WW^* and W^*W with eigenvalue $\mu_i = \sigma_i^2$ i=1,2,...,r

Now, we can write the problem of approximating the snapshot vectors w_i. Let $w_1, w_2, ..., w_M \in \mathbb{R}^n$ be given, and set $\widetilde{\mathcal{V}} = span\{w_1, w_2, ..., w_M\} \subset \mathbb{R}^n$. Our problem is to find $p \leq dim \mathcal{V}$ orthonormal vectors $\{\psi_i\}_{i=1}^p$ which minimize $\sum\limits_{j=1}^{M} \|w_j - \sum\limits_{i=1}^{p} (w_j^* \psi_i) \psi_i\|^2$ with Euclidean norm $\|w\| = \sqrt{w^* w}$.

Here the constrained optimization problem is to find $min \sum_{j=1}^{M} \|w_j - \sum_{i=1}^{p}(w_j^* \psi_i)\psi_i\|^2$ subject to $\psi_i^* \psi_j = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$

For the optimality, we have some conditions. Let $L(\psi_1, \psi_2, ..., \psi_p, \mu_{11}, \mu_{22}, ..., \mu_{pp}) = \sum_{j=1}^{M} \|w_j - \sum_{i=1}^{p}(w_j^* \psi_i)\psi_i\|^2 + \sum_{i,j=1}^{p} \mu_{ij}(\psi_i^* \psi_j - \delta_{ij})$ where $\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$

The conditions are

$$\frac{\partial L}{\partial \psi_i} = 0 \in \mathbb{R}^n \quad for \quad i = 1, ..., p \tag{3.13}$$

and

$$\frac{\partial L}{\partial \mu_{ij}} = 0 \in \mathbb{R} \quad for \quad i, j = 1, ..., p \tag{3.14}$$

From (3.13) we obtain that $\sum_{j=1}^{M} w_j(w_j^* \psi_i) = \mu_{ii}\psi_i$ and $\mu_{ij} = 0$ for $i \neq j$

From (3.14) we obtain that $\psi_i^* \psi_j = \delta_{ij}$

If we set $\mu_i = \mu_{ii}$ and $W = [w_1, w_2, ..., w_M] \in \mathbb{R}^{n \times M}$ we obtain $WW^* \psi_i = \mu_i \psi_i$ for $i = 1, ..., p$ as a necessary for our problem. Here we can see that for $i = 1, ..., p \leq r = dim\widetilde{\mathcal{V}}$ the approximation of the columns $W = (w_1, w_2, ..., w_n)$ by the first r singular vectors $\psi_i = u_i$ is optimal in the least square sense among all rank r approximations to w_i.

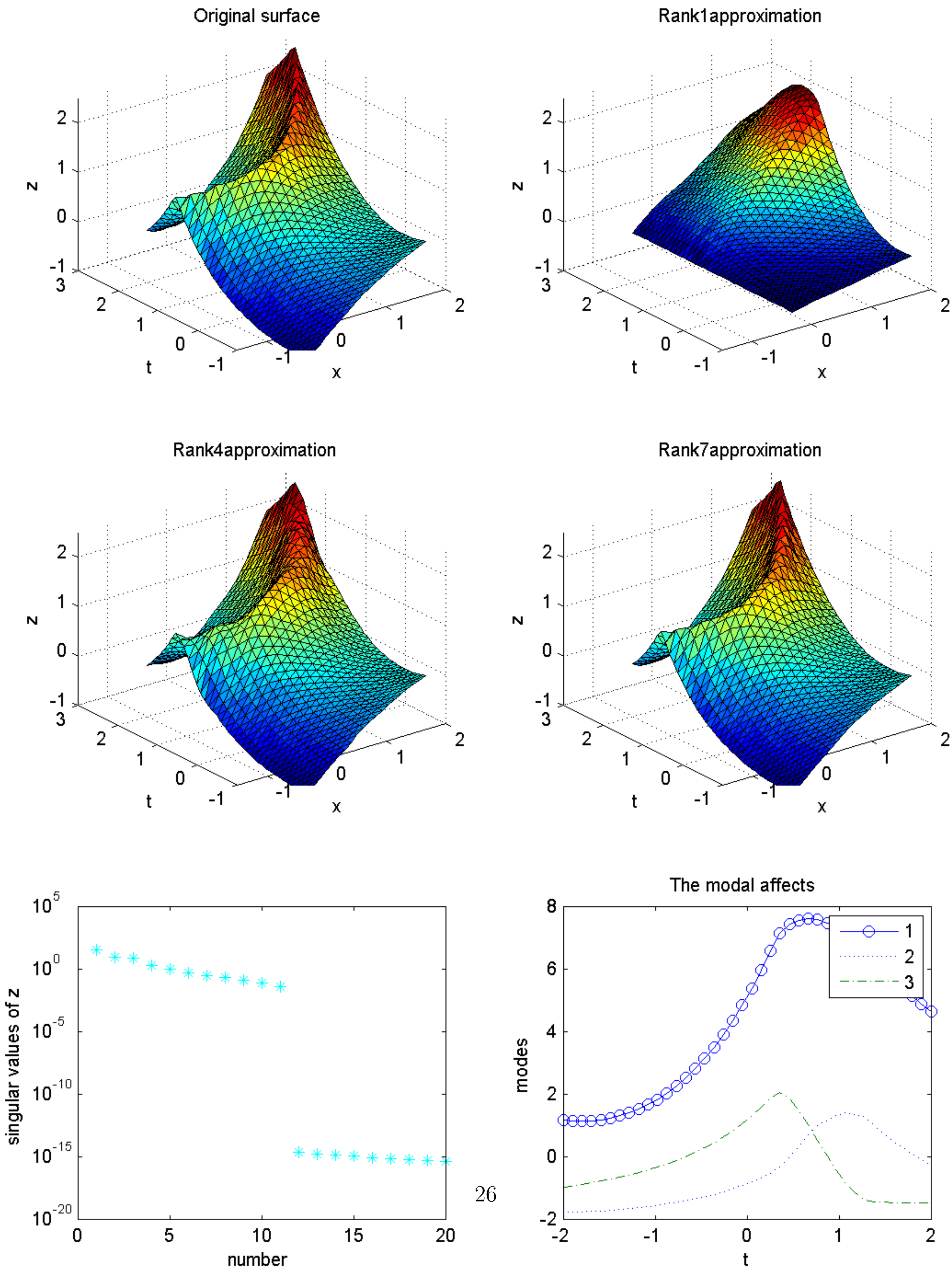
The error formula for the POD basis rank p : $\sum_{j=1}^{M} \|w_j - \sum_{i=1}^{p}(w_j^* \psi_i)\psi_i\|^2 = \sum_{i=p+1}^{r} \lambda_i$.

The following example can show us how to use SVD to approximate a surface. Also POD is used to see the mod shapes in the contex of POD.

```
clear all
x=linspace(-1,1,20);
t=linspace(-2,2,40);
[X,T]=meshgrid(x,t);
Tri=delaunay(X,T);
```

$z = exp(X-1).^*sin(T-.4)/2 + 2^*exp(-abs(X.^\wedge 2-(T-0.3))) + X;$

```
subplot(3,2,1);
trisurf(Tri, X, T, z)
axis([-1.5, 2, -1, 3, -1, 2.5])
xlabel('x'), ylabel('t'), zlabel('z')
title('Original surface')
[u, d, v] = svd(z);
for i=1:3
j=3*i-2;
z1=u(:,1:j)*d(1:j,1:j)*v(:,1:j)';
subplot(3,2,i+1)
Tri1=delaunay(X,T);
trisurf(Tri1,X,T,z1), axis([-1.5,2,-1,3,-1,2.5])
xlabel('x'), ylabel('t'), zlabel('z')
title(['Rank', num2str(j), 'approximation'])
end
subplot(3,2,5)
d=diag(d); semilogy(d,'c*')
xlabel('number'), ylabel('singular values of z')
subplot(3,2,6)
v=v(:,1:3);
k=v'*z';
plot(t,k(1,:),'-bo',t,k(2,:),':',t,k(3,:),'-.')
xlabel('t'), ylabel('modes')
title ('The modal affects')
legend('1','2','3')
```


Original surface
Rank1approximation
Rank4approximation
Rank7approximation
z
t
x
singular values of z
number
The modal affects
modes
t
1
2
3

Bibliography

[1] Lumley, J. L. [1967] “The structure of inhomogeneous turbulent flows”. In: A.M. Yaglom, V. I. Tatarski Athmospheric Turbulence and Wave Propagation, 166-178.

[2] Feldman, P., Freund, R. W. [1994] “Efficient linear circuit analysis by Pàde approximation via the Lanczos process ”. In: Proceedings of EURO-DAC ’94 with EURO-VHDL ’94. IEEE Computer Society Press, Los Alamitos, California, pp. 170-175.

[3] Feldman, P., Freund, R. W. [1995] “Efficient linear circuit analysis by Pàde approximation via the Lanczos process ”, IEEE Trans. Computer-Aided Design 14, 639-649.

[4] Freund R. W. [1999] “Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation”, In: Datta, B. N. (Ed.), Applied and Computational Control, Signals, and Circuits. Vol. 1. Birkhäuser, Boston, pp. 435-498.

[5] Aliaga, J. I., Boley, D. L., Freund, R. W., Hernàndez, V. [2000] “A Lanczos-type method for multiple starting vectors. ”, Math. Comp. 69, 1577-1601.

[6] Chatterjee, A. [2000] “An Introduction to The Proper Orthogonal Decomposition ”, *Current Science*, Vol. 78, No. 7, pp. 808-817.

[7] Freund R. W. [2000a] “Band Lanczos method (Section 7.10)”. In Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (Eds.), Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide. SIAM Publications, Philadelphia, Pennsylvania, pp. 205-216, also available online from http://cm.bell-labs.com/cs/doc/99.

[8] Freund R. W. [2000b] “Krylov-subspace methods for reduced-order modeling in circuit simulation.”, J. Comput. Appl. Math. 123 (1-2), 395-421.

[9] Bai Z., Freund R. W. [2001] “A partial Pàde-via-Lanczos method for reduced-order modeling.” Linear Algebra Appl. 332-334, 139-164.

[10] Bai Z. [2002] “Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems”, Appl. Numer. Math. 43 (1-2), 9-44.

[11] Ravindran [2002] “Adaptive reduced-order conrollers for a thermal flow system using proper orthogonal decomposition”, SIAM J. Sci. Comput., 23(6):1924-1942.

[12] Bai Z. Dewilde P. M. and Freund R. W. [2003] “Reduced-Order Modeling”, *In Numerical Analysis Manuscript*, 02-4-13, Bell Laboratories.

[13] Freund R. W. [2003] "Model reduction methods based on Krylov subspaces", Acta Numerica 12, 267-319.

[14] Cordier L., Bergman M. [2003] "Proper Orthogonal Decomposition: An Overciew", Von Kàrmàn Institut for Fluid Dynamics.

[15] Rathinam M., Petzold L. R. [2003] "A new look at proper orthogonal decomposition.", SIAM J. Numer. Anal., 1893-1925.

[16] Sanjay Lall, Petr Krysl, and Jerrold E. Marsden. [2003] "Structure-preserving model reduction for mechanical systems." Phys. D, 184(1-4):304-318.

[17] Freund R. W. [2004] "Krylov subspaces associated with higher-order linear dynamical systems.", Technical report, Department of Mathematics, University of California, Davis, California.

[18] Reis, T., Stykel, T., [2007] "Stability analysis and model order reduction of coupled systems.", Mathematical and Computer Modeling of Dynamic Systems 13 (5), 413-436.

[19] Pinnau, R. [2008] "Model Reduction Via Proper Orthogonal Decomposition", *Model Order Reduction: Theory, Research Aspects and Applications, Springer*, 94-109.

[20] Schilders W. [2008] "Introduction To Model Order Reduction", *Model Order Reduction: Theory, Research Aspects and Applications, Springer*, 3-32.

[21] Vorst H. [2008] "Linear Systems, Eigenvalues, and Projection", *Model Order Reduction: Theory, Research Aspects and Applications, Springer*, 33-45.

[22] Freund R. W. [2008] "Structure-Preserving Model Order Reduction of RCL Circuit Equations", *Model Order Reduction: Theory, Research Aspects and Applications, Springer*, 49-73.

[23] Luchtenburg, D. M., Noack B. R. and Schlegel M. [2009] "An Introduction To The POD Galerkin Method for Fluid Flows With Analytical Examples and MATLAB Source Codes", *Berlin Institute of Technology MB1*, Muller-Breslau-Strabe 11, D-10623, Berlin.

[24] Li X., Chen X., Hu B. X. and Navon I. M [2013] "Model Reduction of A Coupled Numerical Model Using Proper Orthogonal Decomposition" *Journal of Hydrology* 507, pp. 227-240.