

# A real time procedure for affinely dependent parametric model order reduction using interpolation on Grassmann manifolds

Nguyen Thanh Son<sup>\*,†</sup>

*Center for Industrial Mathematics, University of Bremen, 28359 Bremen, Germany*

## SUMMARY

Model order reduction helps to reduce the computational time in dealing with large dynamical systems, for example, during simulation, control, optimization. In many cases, the considered model depends on parameters; Model order reduction techniques are, therefore, preferred to symbolically preserve this dependence or to be adaptive to the change of the model caused by the variation in the values of the parameters. In this paper, we first present the application of the interpolation technique on Grassmann manifolds to this problem. We then improve the method for the models whose system matrices depend affinely on parameters by considerably reducing the computational complexity on the basis of analyzing the structure of sums of singular value decompositions and decomposing the whole procedure into *offline* and *online* stages. A numerical example is shown to illustrate the method as well as to prove its effectiveness. Copyright © 2012 John Wiley & Sons, Ltd.

Received 17 October 2011; Revised 6 July 2012; Accepted 29 July 2012

KEY WORDS: dynamical systems; model order reduction; offline–online decomposition; real time; interpolation; Grassmann manifolds

## 1. INTRODUCTION

Numerical simulation is nowadays indispensable to manufacturers. This step helps the designers to create the models that meet the requirements of the producers. In steel industry, for example, one wants to make a unit that has different hardness in different parts. This can be partly achieved by controlling the location and magnitude of the heat source and the cooling process. Experiments can be carried out, but it is too expensive in the real life. Therefore, one is expected to know or to predict the properties of the workpiece without or before running the real process.

For systems that are large or complex or both, even with modern computers, the numerical simulation still remains unfeasibly time-consuming. The dominant time is spent on solving very large systems of differential equations or linear equations, which usually come from the spatial discretization of the mathematical models. The vector of unknowns of these equations is called the *state vector*. For systems whose high-fidelity is required or those that have complicated-shaped acting domain, the order of the state vector may reach thousands or millions. There appears the need to replace the large systems with ones that have lower order and approximate the original in some sense. This field, called *model order reduction* (MOR), has attracted much attention, for example, [1–6], just to name a few.

In fact, besides approximating the original system, the reduced order model (ROM) is expected to inherit important properties. These may be stability [7, 8], passivity [9–11], some special structures [12, 13] or the dependence on parameters. The last problem arises when one is dealing with the

\*Correspondence to: Nguyen Thanh Son, Center for Industrial Mathematics, University of Bremen, Bibliothekstr. 1, MZH 2340, 28359 Bremen, Germany.

†E-mail: [nguyen@math.uni-bremen.de](mailto:nguyen@math.uni-bremen.de)

systems involving the change of the design parameters, for example, the length or the thickness of a beam [14], or the environment parameters. For example, the film coefficients of a microthruster unit [15, 16] and the velocity field in convection–diffusion equations [17]. The systems may act very differently when the parameters vary [18], therefore, it is a challenge that MOR methods should preserve the dependence on the parameters or be easily adapted to their variation.

Interpolation is frequently utilized in combination with a MOR method to carry out the task. The difference in which data to be interpolated and which MOR method to be used makes the approaches quite distinct in the sense of their manners and their outcomes. The authors of [17, 19] chose to interpolate the reduced transfer functions (TF). First, the reduced TF is constructed by balanced truncation [1] at some given points in the parameter domain. Some interpolation polynomials, which contain parameters as variables, are then used, and the reduced TF is recovered over the whole parameter range. In [17], Lagrange and Hermite polynomials are used, whereas linear and cubic splines are mobilized in [19]. In both approaches, the authors demonstrated that stability is preserved and various error bounds are obtained.

For the same purpose, the method developed in [20] interpolates the matrices of the reduced systems. At first, the reduced system matrices are locally computed at a given discrete set of parameter values by projection. The parameter-dependent matrices of the reduced system are then calculated by interpolating the corresponding precomputed reduced matrices. The authors pointed out that this procedure may be meaningless if it is conducted directly, that is, without adjustment to the chosen projected subspaces. To avoid this, some strategies were given to adjust the local reduced models.

Matrix interpolation was also used in [21–23]. These authors considered the systems whose matrices have some constraints such as the symmetric positive definiteness or the nonsingularity. They argued that the direct interpolation of system matrices may result in matrices violating these constraints. For this reason, interpolation is not carried out among the reduced system matrices, which actually lie on a manifold, but on the tangent spaces to that manifold. Other parametric MOR (PMOR) methods can be found in, for example, [18, 24–27].

Unlike the aforementioned approaches, the method proposed in [28] interpolates projection subspaces. These subspaces, which depend on parameters, are located on a Grassmann manifold. The interpolation process was performed on the tangent spaces to that manifold. It turns out that this approach is a generalization of the one proposed in [29]. The method was then applied to the analysis of fighter aircraft configurations. This procedure can, however, hardly be used in online simulations because the computation of the ROM for each newly given value of parameters depends on the order of the original system, which is supposed to be very large.

In this paper, we develop a strategy to accelerate the computation and therefore enable it to be used in real time, that is, the complexity of the computation of the reduced system for each new parameter value is independent of the original order. To this end, the remainder of this paper is organized as follows. In Section 2, some facts on the interpolation technique on Grassmann manifolds and the Krylov subspace method are recalled. The procedure for applying them to PMOR is presented in Section 3. Section 4 illuminates how we can reduce the computational complexity in order to accelerate the computation. A numerical example is considered in Section 5. Finally, the conclusion is given in Section 6.

## 2. PRELIMINARIES

### 2.1. Grassmann manifolds and related facts

The Grassmann manifold  $\mathcal{G}(r, N)$  is defined as the set of all  $r$ -dimensional subspaces of  $\mathbb{R}^N$ . Each element of  $\mathcal{G}(r, N)$  is spanned, not uniquely, by the columns of a full-rank  $N \times r$ -matrix, called a matrix representation. These matrices constitute the Stiefel manifold  $\mathcal{ST}(r, N)$ . It is sometimes more convenient to consider the compact Stiefel manifold, which consists of columnwise orthogonal  $N \times r$ -matrices. Even though Grassmann manifolds are more frequently mentioned, the actual computation is performed on the corresponding Stiefel manifold.

Two matrices  $M_1, M_2 \in \mathcal{ST}(r, N)$  represent one point in  $\mathcal{G}(r, N)$ , if and only if their columns span the same subspace. In other words, there exists a nonsingular  $r \times r$ -matrix  $A$  such that  $M_1 = M_2A$ , hence,  $\mathcal{G}(r, N)$  can be defined as the quotient manifold  $\mathcal{ST}(r, N)/\mathcal{GL}(r)$ , where  $\mathcal{GL}(r)$  is the general linear group of degree  $r$ . It is shown [30, 31] that  $\mathcal{G}(r, N)$  is a differentiable manifold equipped with the quotient topology. One can, therefore, mention the tangent space  $\mathcal{T}_S\mathcal{G}(r, N)$  to  $\mathcal{G}(r, N)$  at point  $S$ . In fact, it is demonstrated [32, 33] that there exists one tangent space at any point, which is of the same dimension as the Grassmann manifold. Its elements, henceforth called vectors, are represented by  $N \times r$ -matrices. In [33], a Riemann structure for a Grassmann manifold is constructed according to which the distance on the manifold is defined. We follow this approach in formulating the mappings mentioned later.

There is a one-to-one correspondence between a neighborhood of a point  $S$  on  $\mathcal{G}(r, N)$  and the tangent space  $\mathcal{T}_S\mathcal{G}(r, N)$ . This relation is composed of the exponential mapping,  $\text{Exp}_S$ , which maps the tangent space to the manifold, and its inverse, the logarithmic mapping,  $\text{Log}_S$ , which maps a neighborhood of  $S$  to the tangent space. The formulations of these mappings are essential to our algorithm and hence are recalled here. For more information, the reader is referred to [32, 33].

Let  $W_0 \in \mathcal{ST}(r, N)$  be a columnwise orthogonal matrix whose columns span the subspace  $S_0 \in \mathcal{G}(r, N)$ . Let  $Y$  be a vector of the space  $\mathcal{T}_{S_0}\mathcal{G}(r, N)$  tangent to  $\mathcal{G}(r, N)$  at  $S_0$ , which is represented by  $Z \in \mathbb{R}^{N \times r}$ . If

$$Z = U \Sigma V^T, \tag{1}$$

where  $U \in \mathbb{R}^{N \times r}$ ,  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$ ,  $V \in \mathbb{R}^{r \times r}$  is the thin singular value decomposition (SVD), then  $\text{Exp}_{S_0}(Y) \in \mathcal{G}(r, N)$  is represented by

$$W = W_0 V \cos(\Sigma) + U \sin(\Sigma), \tag{2}$$

where  $\sin$  and  $\cos$  only act on the diagonal of  $\Sigma$ .

The formulation of the exponential mapping is a consequence of the parametric representation of a geodesic path on Grassmann manifolds given the origin and the initial velocity. In [33], a more general formula than (2) is given and applied to the elements of  $\mathcal{ST}(r, N)$  which are not necessary to be columnwise orthogonal.

Based on the formula of the geodesic path connecting two points on Grassmann manifolds, the formulation of the logarithmic mapping is as follows. Let  $S_0, S_1 \in \mathcal{G}(r, N)$  be represented by two columnwise orthogonal elements  $W_0, W_1$  of  $\mathcal{ST}(r, N)$ . Suppose that

$$(I - W_0 W_0^T) W_1 (W_0^T W_1)^{-1} = U \Sigma V^T \tag{3}$$

is the thin SVD. Then  $\text{Log}_{S_0}(S_1)$  is a vector determined by

$$Z = U \arctan(\Sigma) V^T. \tag{4}$$

Let us take an example by considering  $\mathcal{G}(2, 3)$ . Suppose that  $S_0$  and  $S_1$  are two distinct points in  $\mathcal{G}(2, 3)$  spanned by

$$W_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad W_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ 0 & \frac{2}{\sqrt{6}} \end{bmatrix}, \text{ respectively.}$$

We will compute  $S_2 = \text{Exp}_{S_0}(\text{Log}_{S_0}(S_1))$ . In the following presentation, we keep only four decimal digits. First, we have the thin SVD

$$(I - W_0 W_0^T) W_1 (W_0^T W_1)^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2.2361 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -0.4472 & 0.8944 \\ 0.8944 & 0.4472 \end{bmatrix}.$$

Applying (3) and (4), the matrix representing  $\text{Log}_{s_0}(S_1)$  is

$$Z = \begin{bmatrix} 0 & 0 \\ -0.5144 & 1.0288 \\ 0 & 0 \end{bmatrix}.$$

Then by (1) and (2),  $S_2$  is spanned by

$$W_2 = \begin{bmatrix} -0.1826 & 0.8944 \\ 0.9129 & 0 \\ 0.3651 & 0.4472 \end{bmatrix}.$$

One can easily verify that  $W_2$  and  $W_0$  span the same subspace, that is,  $S_2 \equiv S_0$  on  $\mathcal{G}(2, 3)$ .

### 2.2. Krylov subspace method

Given is an LTI dynamical system

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{5}$$

with homogeneous initial condition.  $E, A \in \mathbb{R}^{N \times N}$ ,  $B \in \mathbb{R}^{N \times m}$ ,  $C \in \mathbb{R}^{l \times N}$  are constant and  $E$  is nonsingular.  $B$  and  $C$  are called the input or load matrix and the output matrix, respectively, whereas the four of them are generally referred to as the system matrices.  $x(t)$  is the state vector,  $u(t)$  represents the input function, and  $y(t)$  stands for the output of interest. The TF of system (5) is formulated as

$$H(s) = C(sE - A)^{-1}B.$$

It is a  $l \times m$ -matrix of rational functions of  $s$ , determining the input–output behavior of system (5) in the frequency domain. If  $s_0$  is not a pole,  $H(s)$  can be expanded as a Taylor series in a neighborhood of  $s_0$

$$H(s) = M_0 + M_1(s - s_0) + M_2(s - s_0)^2 + \dots.$$

$M_i$ ,  $i = 1, \dots, \infty$ , as mentioned before, are called the moments of  $H(s)$  about  $s_0$ . It is straightforward to verify that

$$M_i = -C^T((A - s_0E)^{-1}E)^{i-1}(A - s_0E)^{-1}B.$$

One wants to find a reduced system of the order  $r$ ,  $r \ll N$ ,

$$\begin{aligned} \hat{E}\hat{x}(t) &= \hat{A}\hat{x}(t) + \hat{B}u(t) \\ \hat{y}(t) &= \hat{C}\hat{x}(t), \end{aligned} \tag{6}$$

whose TF  $\hat{H}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B}$  shares some first moments with  $H(s)$ . One then, however, realizes that direct computation of coefficients of  $H$  is doomed to failure due to the numerical instability [34, 35]. The main contribution of the Krylov subspace method is to numerically effectively construct the reduced system: moments can be matched without computing them by projecting the original system on a Krylov subspace, that is, the system matrices of (6) are computed as

$$\hat{E} = Z^T E W, \hat{A} = Z^T A W, \hat{B} = Z^T B, \hat{C} = C W,$$

where  $Z, W$  are  $N \times r$ -matrices spanning the Krylov subspace of the form

$$\mathcal{K}_r(K, v) = \text{span}\{v, Kv, K^2v, \dots, K^{r-1}v\}.$$

The following result is the theoretical basis for *multipoint implicit moment matching method* or *rational interpolation*.

*Theorem 2.1* ([4])

If

$$\text{colspan}(W) \supseteq \bigcup_{i=1}^I \mathcal{K}_{r_i}((A - s_i E)^{-1} E, (A - s_i E)^{-1} B) \quad (7)$$

and

$$\text{colspan}(Z) \supseteq \bigcup_{i=1}^I \mathcal{K}_{q_i}((A - s_i E)^{-T} E, (A - s_i E)^{-T} C^T), \quad (8)$$

then

$$-C^T((A - s_i E)^{-1} E)^{k_i-1} (A - s_0 E)^{-1} B = -\hat{C}^T((\hat{A} - s_i \hat{E})^{-1} \hat{E})^{k_i-1} (\hat{A} - s_0 \hat{E})^{-1} \hat{B}$$

for  $k_i = 1, \dots, r_i + q_i$ , and  $i = 1, \dots, I$ . In words,  $r_i + q_i$  moments about  $s_i$  will be matched if the projection subspaces used for order reduction are constructed satisfying (7) and (8).

*Remark 2.1*

The above theorem holds true only for SISO systems, that is,  $l = m = 1$ . For MIMO systems, according to [36], the number of matched moments about  $s_i$  is  $[r_i/m] + [q_i/l]$ ,  $i = 1, \dots, I$ , where  $[\cdot]$  denotes the integer part of a number.

The choice for  $Z$  in (8) can be relaxed by picking any full-rank matrix that has the same size as  $W$ . Note that in this case, because condition (8) is not satisfied, only  $r_i$  (in SISO case), or  $[r_i/m]$  (in MIMO case) moments are matched. One common choice is  $Z = W$ ; we have one-sided projection. To avoid the ill-conditionedness in the computation, matrix  $W$  is constructed to be orthogonal, that is,  $W^T W = I$  using Arnoldi process (e.g., [5]).

### 3. PARAMETRIC MODEL ORDER REDUCTION USING INTERPOLATION ON GRASSMANN MANIFOLDS (IGM)

Our PMOR problem is formulated as follows: Given a parameter-dependent linear dynamical system

$$\begin{aligned} E(p)\dot{x}(t) &= A(p)x(t) + B(p)u(t) \\ y(t) &= C(p)x(t), \end{aligned} \quad (9)$$

where  $E(p)$ ,  $A(p)$ ,  $B(p)$ ,  $C(p)$  have the same size as in system (5) and depend on parameter  $p \in \Omega$ . We assume moreover that  $\Omega$  is a connected subset of  $\mathbb{R}^d$  and the dependence of  $E$ ,  $A$ ,  $B$ ,  $C$  on  $p$  is smooth enough for the interpolation used later, for example, continuously differentiable if the linear Lagrange polynomial interpolation is used. Seek a reduced system

$$\begin{aligned} \hat{E}(p)\hat{x}(t) &= \hat{A}(p)\hat{x}(t) + \hat{B}(p)u(t) \\ \hat{y}(t) &= \hat{C}(p)\hat{x}(t), \end{aligned} \quad (10)$$

where  $E, A \in \mathbb{R}^{r \times r}$ ,  $B \in \mathbb{R}^{r \times m}$ ,  $C \in \mathbb{R}^{l \times r}$ ,  $r \ll N$  such that it approximates the original (9) in some sense for all  $p$  in some subset (not necessarily a strict subset) of  $\Omega$ .

We will use one-sided projection;  $Z$  in (8) need not be computed. The projection subspace in (7) is no longer a constant subspace because the system matrices depend on  $p$ . The direct computation of  $W(p)$  for all  $p \in \Omega$  is impossible. Thus, interpolation is invoked as a tool to construct an approximation of  $W(p)$ . Let us state the task of interpolating subspaces as follows:

*Given  $p_0, \dots, p_k \in \Omega$ , denote by  $W_0, \dots, W_k$  the columnwise orthogonal matrices spanning the projection subspaces  $S_0, \dots, S_k$ , respectively. Construct a parameter-dependent basis  $W(p)$  for  $S(p)$  by interpolation of data  $(p_0, W_0), \dots, (p_k, W_k)$ .*

A simple idea is to form  $W(p)$  as a weighted sum of  $W_0, \dots, W_k$ ,

$$W(p) = \sum_{i=1}^k \omega_i(p) W_i,$$

where  $\omega_i(p)$  are some weight functions. This solution may lead to a situation in which the resulting matrix  $W(p)$  is no longer a basis for a subspace. In other words, the direct interpolation on Grassmann manifolds may result in a point that is not included in it. This can be illustrated in the following example. Consider  $S_1, S_2 \in \mathcal{G}(2, 3)$  represented by

$$W_1 = \begin{bmatrix} 0 & 0 \\ 9 & 3 \\ 18 & 0 \end{bmatrix} \text{ and } W_2 = \begin{bmatrix} 2 & \frac{1}{3} \\ 1 & 0 \\ 0 & \frac{1}{3} \end{bmatrix}.$$

Then  $S = \frac{1}{10} S_1 + \frac{9}{10} S_2$  is represented by

$$\begin{aligned} W &= \frac{1}{10} W_1 + \frac{9}{10} W_2 \\ &= \begin{bmatrix} 0 & 0 \\ \frac{9}{10} & \frac{3}{10} \\ \frac{18}{10} & 0 \end{bmatrix} + \begin{bmatrix} \frac{18}{10} & \frac{3}{10} \\ \frac{9}{10} & 0 \\ 0 & \frac{3}{10} \end{bmatrix} \\ &= \begin{bmatrix} \frac{9}{5} & \frac{3}{10} \\ \frac{9}{5} & \frac{3}{10} \\ \frac{9}{5} & \frac{3}{10} \end{bmatrix}. \end{aligned}$$

Because  $\text{rank}(W) = 1$ ,  $S \notin \mathcal{G}(2, 3)$ . The reason is that Grassmann manifolds are, in general, not spaces; they are not flat. Hence, interpolation must be modified or performed in an approximation sense.

One cannot interpolate the functions whose values lie on a Grassmann manifold, but one can do that on its tangent spaces. However, the data are given on the manifold; one has to transfer them between the manifold and its tangent spaces. This can be done, thanks to the two mappings mentioned in Section 2. The following procedure proposed in [28] helps to complete the aforementioned task.

*Step 1* Choose the contact point for the tangent space, for example,  $S_0$ .

*Step 2* Map points  $S_1, \dots, S_k$  to  $\mathcal{T}_{S_0} \mathcal{G}(r, N)$  by  $\text{Log}_{S_0}$ . By (3)–(4),  $\text{Log}_{S_0}(S_i) = Y_i$  is a vector represented by

$$Z_i = U_i \arctan(\Sigma_i) V_i^T,$$

where

$$(I - W_0 W_0^T) W_i (W_0^T W_i)^{-1} = U_i \Sigma_i V_i^T, i = 1, \dots, k$$

are thin SVD.

*Step 3* Interpolate on  $\mathcal{T}_{S_0} \mathcal{G}(r, N)$  using some standard interpolation technique. Note that  $\text{Log}_{S_0} \mathcal{G}(S_0) = Y_0 = 0$ . Given a parameter value  $p \in \Omega$ , denote by  $Y(p)$  the vector on  $\mathcal{T}_{S_0}(r, N)$  corresponding to  $p$ . By any common interpolation type,  $Y(p)$  is represented by the matrix

$$Z(p) = \sum_{i=1}^k f_i(p) Z_i. \tag{11}$$

*Step 4* Map the interpolated result back to the Grassmann manifold. Using exponential mapping, (1)–(2), one has to first compute the thin SVD

$$Z(p) = U(p)\Sigma(p)V(p)^T, \quad (12)$$

and then the matrix representation of the subspace

$$W(p) = W_0 V(p) \cos(\Sigma(p)) + U(p) \sin(\Sigma(p)). \quad (13)$$

Finally, the system matrices of (10) are constructed as

$$\begin{aligned} \hat{E}(p) &= W^T(p)E(p)W(p), \\ \hat{A}(p) &= W^T(p)A(p)W(p), \\ \hat{B}(p) &= W^T(p)B(p), \\ \hat{C}(p) &= C(p)W(p). \end{aligned} \quad (14)$$

*Remark 3.1*

This method can be combined with any MOR method that can be formulated as one-sided projection such as proper orthogonal decomposition or one-sided Krylov subspace method.

One condition that has to be fulfilled when using this method is that  $S_1, \dots, S_k$  are not so far from  $S_0$ . This is because the connection between the Grassmann manifold and its tangent spaces is based on geodesic paths, which are determined by second-order differential equations [33]. The closeness of  $S_i$ ,  $i = 1, \dots, k$ , to  $S_0$  is a requirement for the existence of the solution of the underlying equations. In the case that the distance, which is defined in [33], between  $S_i$  and  $S_0$  is rather large, one can partition the parameter domain into some subdomains and choose one contact point for each subdomain.

#### 4. REDUCTION OF COMPUTATIONAL COMPLEXITY

In online simulations, for a given  $p \in \Omega$ , one needs to compute the reduced system at  $p$  as fast as possible. Applying the aforementioned 4-step procedure, one has to compute

- Interpolation on the tangent space at Step 3 ( $\mathcal{O}(Nr)$ );
- Thin SVD (12) ( $\mathcal{O}(Nr^2)$ );
- Matrix multiplication (13), ( $\mathcal{O}(Nr^2)$ );
- Reduced system matrices (14) ( $\mathcal{O}(N^2r)$ ).

In MOR framework,  $N$  is supposed to be large. With the computational complexity of  $\mathcal{O}(N^2r)$ , the online computation is in general rather slow. This is the reason why the result in [28] failed to be used in real time. To enable this, the only way is to terminate the dependence of the computational complexity on  $N$ . The presentation of our solution will start with a simple case.

##### 4.1. Linear interpolation and single parameter

Let  $\Omega = [a, b]$  and  $a = p_0 < \dots < p_k = b$ . Because the procedure can be applied to each subinterval  $[p_{i-1}, p_i]$ , we can restrict ourselves, without loss of generality, to the case  $k = 1$ . The process of interpolating on Grassmann manifolds for the case of linear interpolation and single parameter is illustrated via Figure 1.

At Step 3, using linear interpolation with two vectors  $Y_1$  and  $Y_0 (\equiv 0)$  leads to

$$Y(p) = \frac{p - p_0}{p_1 - p_0} Y_1.$$



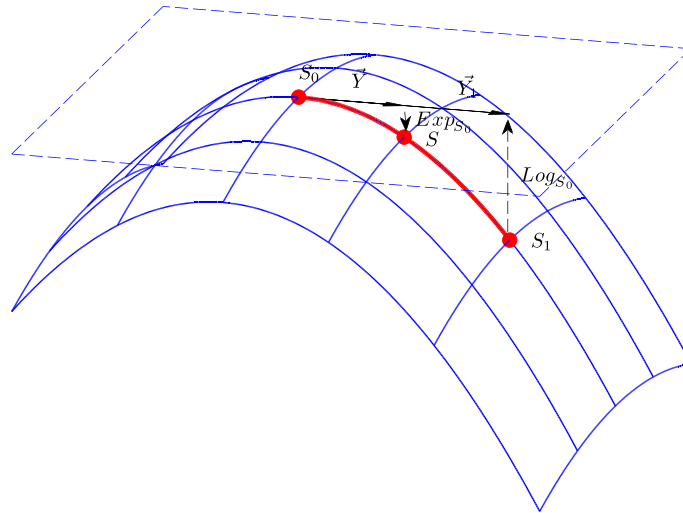


Figure 1. Interpretation of interpolation on Grassmann manifolds.

Therefore

$$Z(p) = U_1 \frac{p - p_0}{p_1 - p_0} \arctan(\Sigma_1) V_1^T. \tag{15}$$

Note that (15) is still the thin SVD of  $Z(p)$ , we do not have to compute the SVD (12) in Step 4; the basis for the projection subspace at  $p$  is straightforwardly written down as

$$W(p) = W_0 V_1 \cos\left(\frac{p - p_0}{p_1 - p_0} \arctan(\Sigma_1)\right) + U_1 \sin\left(\frac{p - p_0}{p_1 - p_0} \arctan(\Sigma_1)\right).$$

Inspired by the reduced basis method [37], which was first proposed to deal with parameterized elliptic equations, now we assume that the system matrices of (9) affinely depend on  $p$ , that is,

$$\begin{aligned} E(p) &= \sum_{i=1}^{\Phi_E} f_i^E(p) E_i, \\ A(p) &= \sum_{i=1}^{\Phi_A} f_i^A(p) A_i, \\ B(p) &= \sum_{i=1}^{\Phi_B} f_i^B(p) B_i, \\ C(p) &= \sum_{i=1}^{\Phi_C} f_i^C(p) C_i, \end{aligned} \tag{16}$$

where  $E_i, A_i, B_i, C_i$  are independent of  $p$ . For the effectiveness of the method presented later, we assume, moreover, that  $\Phi_E, \Phi_A, \Phi_B, \Phi_C$  are very small in comparison with  $N$ , and the evaluations of  $f_i^E, f_i^A, f_i^B, f_i^C$  for all  $p$  are cheap. Indeed many mathematical models fulfill this conditions, for example, Helmholtz problem [38], heat conduction problem [39], and thermal flow [40].



For the sake of brevity, we denote by  $\Xi(p)$  the diagonal matrix  $((p - p_0)/(p_1 - p_0))\arctan(\Sigma_1)$ . Accordingly, the reduced matrices (14) are written as

$$\begin{aligned}
 \hat{E}(p) &= W^T(p)E(p)W(p) = \sum_{i=1}^{\Phi_E} f_i^E(p)W^T(p)E_iW(p) \\
 &= \sum_{i=1}^{\Phi_E} f_i^E(p) (\cos(\Xi(p))V_1^T W_0^T + \sin(\Xi(p))U_1^T) E_i (W_0 V_1 \cos(\Xi(p)) + U_1 \sin(\Xi(p))) \\
 &= \sum_{i=1}^{\Phi_E} f_i^E(p) \cos(\Xi(p)) \mathbf{V}_1^T \mathbf{W}_0^T \mathbf{E}_i \mathbf{W}_0 \mathbf{V}_1 \cos(\Xi(p)) \\
 &\quad + \sum_{i=1}^{\Phi_E} f_i^E(p) \cos(\Xi(p)) \mathbf{V}_1^T \mathbf{W}_0^T \mathbf{E}_i \mathbf{U}_1 \sin(\Xi(p)) \\
 &\quad + \sum_{i=1}^{\Phi_E} f_i^E(p) \sin(\Xi(p)) \mathbf{U}_1^T \mathbf{E}_i \mathbf{W}_0 \mathbf{V}_1 \cos(\Xi(p)) \\
 &\quad + \sum_{i=1}^{\Phi_E} f_i^E(p) \sin(\Xi(p)) \mathbf{U}_1^T \mathbf{E}_i \mathbf{U}_1 \sin(\Xi(p)).
 \end{aligned} \tag{17}$$

$\hat{A}(p)$  is computed analogously. Likewise,

$$\begin{aligned}
 \hat{B}(p) &= W^T(p)B(p) = \sum_{i=1}^{\Phi_B} f_i^B(p)W^T(p)B_i \\
 &= \sum_{i=1}^{\Phi_B} f_i^B(p) \cos(\Xi(p)) \mathbf{V}_1^T \mathbf{W}_0^T \mathbf{B}_i + \sum_{i=1}^{\Phi_B} f_i^B(p) \sin(\Xi(p)) \mathbf{U}_1^T \mathbf{B}_i.
 \end{aligned} \tag{18}$$

$$\begin{aligned}
 \hat{C}(p) &= C(p)W(p) = \sum_{i=1}^{\Phi_C} f_i^C(p)C_iW(p) \\
 &= \sum_{i=1}^{\Phi_C} f_i^C(p) \mathbf{C}_i \mathbf{W}_0 \mathbf{V}_1 \cos(\Xi(p)) + \sum_{i=1}^{\Phi_C} f_i^C(p) \mathbf{C}_i \mathbf{U}_1 \sin(\Xi(p)).
 \end{aligned} \tag{19}$$

All the matrices in (17)–(19) emphasized with bold letters are independent of  $p$ ; they can be computed and stored before starting the online stage. In the following, we summarize the whole PMOR process in terms of offline–online decomposition.

*Offline* We compute and store

- Two columnwise orthogonal projection matrices  $W_0, W_1$  at  $p_0$  and  $p_1$  by the Krylov subspace method.
- The thin SVD

$$(I - W_0 W_0^T) W_1 (W_0^T W_1)^{-1} = U \Sigma V^T.$$

- Parameter-independent terms:  $\mathbf{V}^T \mathbf{W}_0^T \mathbf{E}_i \mathbf{W}_0 \mathbf{V}$ ,  $\mathbf{V}^T \mathbf{W}_0^T \mathbf{E}_i \mathbf{U}$ ,  $\mathbf{U}^T \mathbf{E}_i \mathbf{W}_0 \mathbf{V}$ ,  $\dots$ .

The most expensive computation is the SVD and matrix multiplication, which are of the order  $\mathcal{O}(N^2 r)$ .

*Online* Given a parameter value  $p$ , compute the reduced system matrices via (17)–(19).

The computation cost of the online stage is  $\mathcal{O}(r^2)$ , totally independent of  $N$ .

4.2. General case

The key point of the solution for the case of linear interpolation and single parameter is the formula (15). Thanks to the simplicity, we have an explicit expression for  $W(p)$  without the computation of SVD (12) as in the general case. To extend the result to the general case, we have to deal with the SVD of the sum (11). More concretely, a suitable strategy to compute the SVD of (11) has to be set up, and a careful combination of it with the offline–online decomposition later on must be performed.

One can observe that no matter what the dimension of parameter domain is and/or no matter how high the order of interpolation reaches, by (11), one derives the interpolant of the form

$$Z(p) = \sum_{i=1}^k U_i \alpha_i(p) \arctan(\Sigma_i) V_i^T. \tag{20}$$

Obviously, each term in (20) is still a thin SVD. Hence, the problem we are dealing with is the SVD of a sum of SVDs. Based on the modification technique for thin SVD proposed in [41], which considered the SVD of the sum of an SVD and a low rank updating matrix, we succeeded in solving the problem in the general case. One can write

$$\begin{aligned} Z(p) &= \sum_{i=1}^k U_i \alpha_i(p) S_i V_i^T \\ &= [U_1 \ \cdots \ U_k] \begin{bmatrix} \alpha_1(p) S_1 & & \\ & \ddots & \\ & & \alpha_k(p) S_k \end{bmatrix} \begin{bmatrix} V_1^T \\ \vdots \\ V_k^T \end{bmatrix}. \end{aligned} \tag{21}$$

Denote by  $P$  the matrix whose columns are the left singular vectors of  $(I - U_1 U_1^T) [U_2 \ \cdots \ U_k]$ . It is an orthonormal basis of the intersection of the orthogonal complement of the subspace spanned by the columns of  $U_1$  and the subspace spanned by that of  $U_2, \dots, U_k$ . Note that the number of columns of  $P$ ,  $n$ , satisfies  $n \leq (k - 1)r$ . Thanks to projection, we have

$$[U_1 \ \cdots \ U_k] = [U_1 \ P] \begin{bmatrix} I & U_1^T U_2 & \cdots & U_1^T U_k \\ 0 & P^T (I - U_1 U_1^T) U_2 & \cdots & P^T (I - U_1 U_1^T) U_k \end{bmatrix}.$$

For the same reason, we can write

$$[V_1 \ \cdots \ V_k] = V_1 [I \ V_1^T V_2 \ \cdots \ V_1^T V_k].$$

Now replace the first and the last factors of (21) by the corresponding quantities, we get

$$\begin{aligned} Z(p) &= [U_1 \ P] \begin{bmatrix} I & U_1^T U_2 & \cdots & U_1^T U_k \\ 0 & P^T (I - U_1 U_1^T) U_2 & \cdots & P^T (I - U_1 U_1^T) U_k \end{bmatrix} \\ &\quad \begin{bmatrix} \alpha_1(p) S_1 & & \\ & \ddots & \\ & & \alpha_k(p) S_k \end{bmatrix} \begin{bmatrix} I \\ \vdots \\ V_k^T V_1 \end{bmatrix} V_1^T \\ &= [U_1 \ P] K(p) V_1^T, \end{aligned}$$

where

$$K = \begin{bmatrix} \alpha_1(p) S_1 + \alpha_2(p) U_1^T Z_2 V_1 + \cdots + \alpha_k(p) U_1^T Z_k V_1 \\ \alpha_2(p) P^T (I - U_1 U_1^T) Z_2 V_1 + \cdots + \alpha_k(p) P^T (I - U_1 U_1^T) Z_k V_1 \end{bmatrix} \in \mathbb{R}^{(r+n) \times r}. \tag{22}$$

Let us denote the thin SVD of  $K(p)$  by

$$K(p) = \Phi(p)\Sigma(p)\Psi(p)^T.$$

Then the SVD of  $Z(p)$  is therefore

$$Z(p) = ([U_1 \ P]\Phi(p))\Sigma(p)(V_1\Psi(p))^T.$$

Accordingly, the basis for the projection subspace is

$$W(p) = W_0 V_1 \Psi(p) \cos(\Sigma(p)) + [U_1 \ P]\Phi(p) \sin(\Sigma(p)). \quad (23)$$

Now, using the assumption on the affine dependence (16), the reduced system is constructed similarly

$$\begin{aligned} \hat{E}(p) &= W^T(p)EW(p) \\ &= \sum_{i=1}^{\Phi_E} f_i^E(p) \cos(\Sigma(p))\Psi(p)^T \mathbf{V}_1^T \mathbf{W}_0^T \mathbf{E}_i \mathbf{W}_0 \mathbf{V}_1 \Psi(p) \cos(\Sigma(p)) \\ &\quad + \sum_{i=1}^{\Phi_E} f_i^E(p) \cos(\Sigma(p))\Psi(p)^T \mathbf{V}_1^T \mathbf{W}_0^T \mathbf{E}_i [U_1 \ P]\Phi(p) \sin(\Sigma(p)) \\ &\quad + \sum_{i=1}^{\Phi_E} f_i^E(p) \sin(\Sigma(p))\Phi(p)^T [U_1 \ P]^T \mathbf{E}_i \mathbf{W}_0 \mathbf{V}_1 \Psi(p) \cos(\Sigma(p)) \\ &\quad + \sum_{i=1}^{\Phi_E} f_i^E(p) \sin(\Sigma(p))\Phi(p)^T [U_1 \ P]^T \mathbf{E}_i [U_1 \ P]\Phi(p) \sin(\Sigma(p)). \end{aligned} \quad (24)$$

$\hat{A}(p)$  is constructed exactly the same as the previous equation. The load matrix and output matrix are

$$\begin{aligned} \hat{B}(p) &= W^T(p)B = \sum_{i=1}^{\Phi_B} f_i^B(p) \cos(\Sigma(p))\Psi(p)^T \mathbf{V}_1^T \mathbf{W}_0^T \mathbf{B}_i \\ &\quad + \sum_{i=1}^{\Phi_B} f_i^B(p) \sin(\Sigma(p))\Phi(p)^T [U_1 \ P]^T \mathbf{B}_i \end{aligned} \quad (25)$$

and

$$\begin{aligned} \hat{C}(p) &= CW(p) = \sum_{i=1}^{\Phi_C} f_i^C(p) \mathbf{C}_i \mathbf{W}_0 \mathbf{V}_1 \Psi(p) \cos(\Sigma(p)) \\ &\quad + \sum_{i=1}^{\Phi_C} f_i^C(p) \mathbf{C}_i [U_1 \ P]\Phi(p) \sin(\Sigma(p)), \end{aligned} \quad (26)$$

respectively. One can realize that all quantities emphasized with bold letters in (22), (24), (25), (26) are independent of  $p$  and therefore can be computed and stored beforehand. We now summarize the procedure in the form of offline–online decomposition as follows.

*Offline* We compute and store:

- $W_0, W_1, \dots, W_k$  corresponding to  $p_0, p_1, \dots, p_k$ .
- $[U_1, \Sigma_1, V_1], \dots, [U_k, \Sigma_k, V_k]$  representing  $\text{Log}_{S_0}(S_1), \dots, \text{Log}_{S_0}(S_k)$ , respectively.
- $P \in \mathbb{R}^{N \times n}$  by SVD.
- All necessary quantities (in bold letters in (22)) for matrix  $K$ .
- All necessary quantities (in bold letters in (24), (25), (26)) for reduced matrices  $\hat{E}, \hat{A}, \hat{B}, \hat{C}$ .

*Online* Given any value  $p$  in the range of  $p_0, \dots, p_k$ , we compute

- Matrix  $K$  as in (22).
- The thin SVD of  $K$ :  $\Phi \Sigma \Psi^T = K$ .
- The reduced system matrices by (24), (25) and (26).

The computation cost of the online stage is  $\mathcal{O}((r+n)r^2)$ , which can be considered as  $\mathcal{O}(r^3)$ .

*Remark 4.1*

Matrix  $P$  and therefore matrix  $K$  depend on the choice of the order of  $U_1, \dots, U_k$  in the sum (21). However,  $W(p)$  in (23) always spans the same subspace because  $W_0$  and the subspace spanned by the columns of the first factor in the second term of  $W(p)$  are the same with respect to any order of  $U_1, \dots, U_k$ .

In [27], an offline–online decomposition was also used for reduction of parameter-dependent systems. That approach, however, constructed a pair of constant projection matrices, which are considered valid for any value of parameter. No interpolation scheme has been considered. Our method here interpolates parameter-dependent projection subspaces, produces a subspace for each value of the parameters and therefore can catch the dynamics of the projection subspaces. To get a better approximation, one has to, in general, increase the number of interpolation points; this does not result in a higher reduced order in our method whereas in [27], the issue of constructing the projection subspaces has not been discussed clearly. Because its authors used only constant projection subspaces, we can deduce that adding more information into the projection subspaces means increasing their dimensions. As a consequence, the reduced order becomes higher. Nevertheless, the procedure used in [27] is simpler than here.

## 5. NUMERICAL EXAMPLE

In this section, the applicability of the proposed method is illustrated through an example taken from Oberwolfach model reduction benchmark collection. This models the temperature distribution on a chip, in which the difference in heat exchange of different parts of the chip with the surrounding has to be taken into account. The spatial discretization of the heat transfer partial differential equation gives a system of the order 4257:

$$E \frac{dT}{dt} = (A - h_{\text{top}} A_{\text{top}} - h_{\text{bot}} A_{\text{bot}} - h_{\text{sid}} A_{\text{sid}}) T(t) + B u(t)$$

$$y(t) = C T(t),$$

where  $E$  and  $A$ , the heat capacity and the heat conductivity matrices, are symmetric,  $B$  is the load vector,  $C$  is the output matrix. We, however, retain only the first row of  $C$  in order to simplify the error evaluation.  $A_{\text{top}}$ ,  $A_{\text{bot}}$ , and  $A_{\text{sid}}$  are the diagonal matrices derived from the discretization of the convection boundary conditions on the top, at the bottom and on the side of the chip with the corresponding film coefficients  $h_{\text{top}}$ ,  $h_{\text{bot}}$ ,  $h_{\text{sid}}$ .  $T$  is the vector of unknown temperatures. All system matrices are sparse. The reader is referred to [15, 16] for more details.

As the first test for linear interpolation and single parameter, we fix two parameters  $h_{\text{top}}$ ,  $h_{\text{bot}}$  and let  $h_{\text{sid}}$  vary from 10 to  $10^9$ . The projection matrices corresponding to  $h_{\text{sid}} = 10$  and  $h_{\text{sid}} = 10^9$  are computed by the Krylov subspace method with the intention of matching moments about  $s_0 = 100$ . The reduced orders are 20 and 40. To check the quality of the approximation, we compute the relative errors of the reduced TF with respect to  $\mathcal{H}_\infty$ -norm. We use the approximation of the form

$$\|H(\cdot)\|_{\mathcal{H}_\infty} \approx \max_{w \in [w_{\min}, w_{\max}]} |H(iw)|,$$

where  $i$  denotes the imaginary unit. These relative errors are then compared with the corresponding ones caused by direct computation, that is, the reduced system is constructed by fixing the parameter at points of interest. These errors are plotted in Figure 2. We then perform the same test with  $h_{\text{bot}}$ ; the errors are plotted in Figure 3.

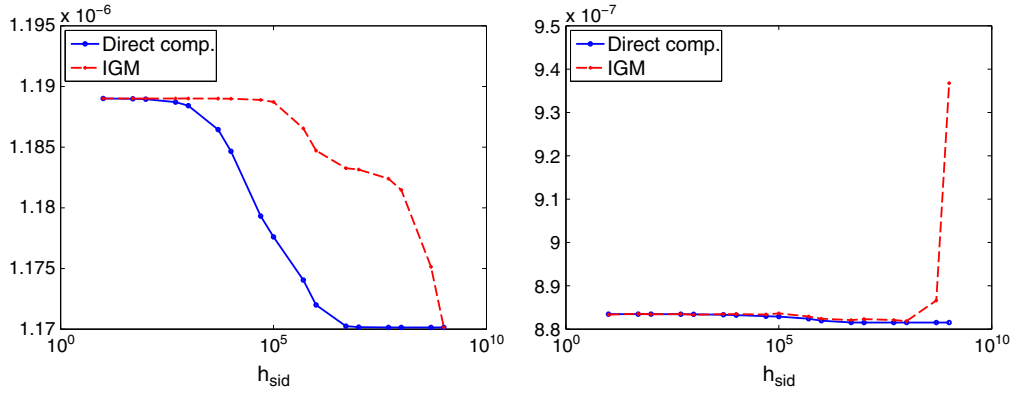


Figure 2. Relative errors using interpolation on Grassmann manifolds versus direct method; reduced order: 20 (left), 40 (right).

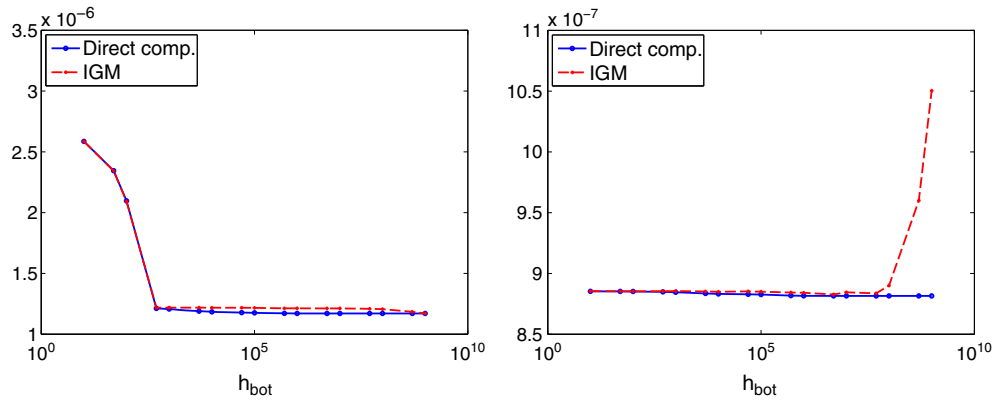


Figure 3. Relative errors using Interpolation on Grassmann manifolds versus direct method; reduced order: 20 (left), 40 (right).

Table I. Computational time: linear interpolation.

Reduced order	10	20	30	40
With off–on decomp.	0.0479	0.0508	0.0563	0.0675
Without off–on decomp.	0.9468	3.0626	5.6700	7.2910
Acceleration factor	19.7851	60.3121	100.7854	107.9589

To verify the reduction in computational time, the reduced system is computed at different parameter points. All the computations are performed with MATLAB R2010b (MathWorks, Natick, MA, USA) on a computer using Linux/Debian 5.0, and equipped with processor 2-GHz 2-GB AMD Athlon 64 X2 (GlobalFoundries Inc., Milpitas, CA, USA). Because the computational time can slightly vary from point to point, we compute the reduced system at 99 points in the interval  $[10, 10, 000]$ . The time, counted in seconds, consumed by the procedure with offline–online decomposition and that without offline–online decomposition at different reduced orders is listed in Table I.

In the second test, to simplify the error illustration, we consider the case of two parameters (instead of all three parameters of the considered model) and bilinear interpolation. To this end, we fix  $h_{top}$  and let  $h_{bot}$  and  $h_{sid}$  vary from 50 to  $5 \times 10^4$  and 5 to  $5 \times 10^4$ , respectively. We will examine the reduced system at totally 100 grid points corresponding to the typical values of parameters  $h_{bot}$  and  $h_{sid}$  mentioned in [15]. First of all, we compute four projection subspaces at  $(h_{bot}, h_{sid}) = (50, 5)$ ,  $(5 \times 10^4, 5)$ ,  $(50, 5 \times 10^4)$  and  $(5 \times 10^4, 5 \times 10^4)$  with the intention of matching moments about

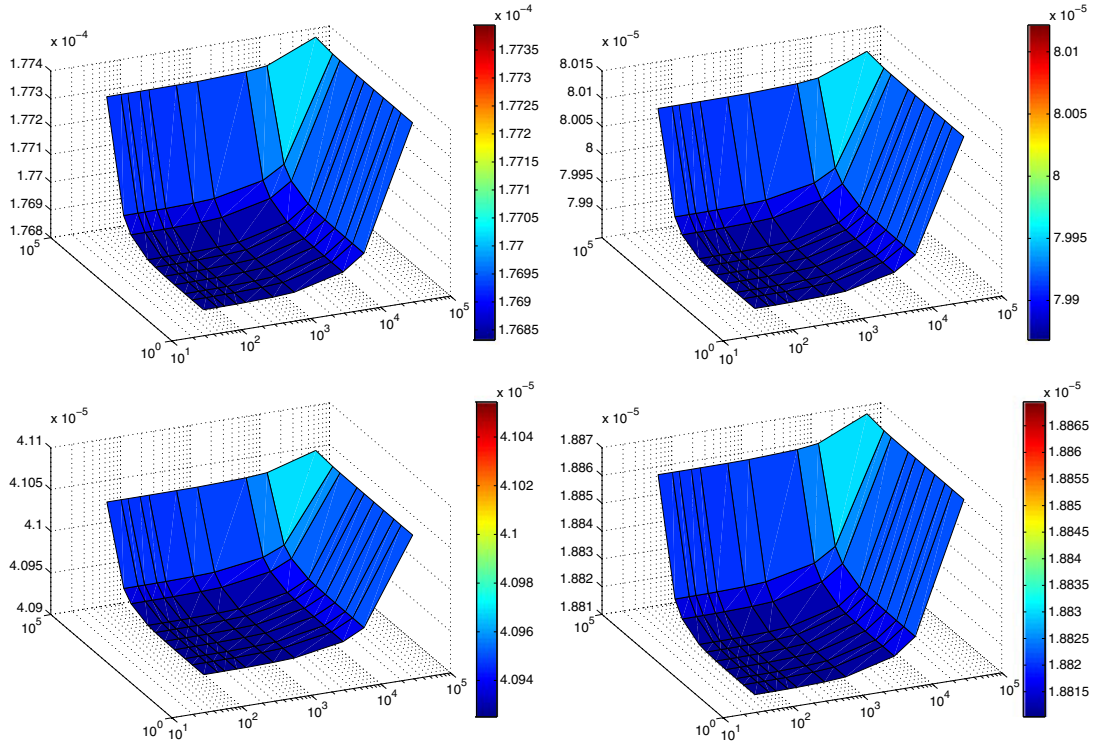


Figure 4. Relative errors using bilinear interpolation; reduced order: 10 (top-left), 20 (top-right), 30 (bottom-left), 40 (bottom-right).

Table II. Computational time: bilinear interpolation.

Reduced order	10	20	30	40
With off-on decomp.	0.0674	0.1982	0.4562	0.8372
Without off-on decomp.	1.0480	3.0708	5.8994	7.5586
Acceleration factor	15.5415	15.4934	12.9309	9.0287

$s_0 = 100$ . The reduced orders are 10, 20, 30 and 40. The subspace at (50, 5) will be used as the contact point. The relative errors of the reduced models are plotted in Figure 4. The computational time is listed in Table II.

We can realize that the advantage of using the proposed method is different in the linear case and the bilinear case as the reduced order varies. In the linear case, the higher the reduced order is, the bigger the acceleration factor is, whereas in the bilinear case, it gets smaller. The reason is that in the linear case, the procedure is simple, we do not have to compute matrix  $K$  as well as its SVD. Therefore, when the reduced order is higher, we can take advantage of this fact. Meanwhile, in the bilinear case, the computation of  $K$  and its SVD slows down the online stage as the reduced order increases.

## 6. CONCLUSIONS

Interpolation on Grassmann manifolds can be used to produce ROMs of parameter-dependent linear dynamical systems. By examining the formulation of exponential and logarithmic mappings, analyzing the structure of SVD sums, and decomposing the process into offline and online stages, we reduced the computational time and therefore enabled the usage of the procedure in real time.

Although the investigated method partly improved the approach proposed in [28] in the sense of computational time, some problems still remain open. The first one is a strategy to choose the grid

points. A uniform grid is preferred, but it is unclear that this is always the best choice. The second issue is the contact point  $S_0$ . It was proposed that  $S_0$  is chosen such that the distances between it and the other grid points are small. This, however, only ensures that the other grid points belong in a small neighborhood of  $S_0$ . It has not been proven that this is the optimal choice in the sense of approximation quality. And even though the mentioned distances are small, one cannot theoretically ensure the existence of the geodesic paths connecting the contact point and the other grid points.

## ACKNOWLEDGEMENTS

This research was funded by the Scientific Computing in Engineering (SCiE) program at the University of Bremen. The author also would like to thank the referees for valuable suggestions.

## REFERENCES

1. Moore BC. Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Transactions on Automatic Control* 1981; **26**(1):17–32. DOI: 10.1109/TAC.1981.1102568.
2. Glover K. All optimal Hankel-norm approximations of linear multivariable systems and their  $L^\infty$ -error bounds. *International Journal of Control* 1984; **39**(6):1115–1193. DOI: 10.1080/00207178408933239.
3. Holmes P. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press: Cambridge, 1996.
4. Grimme EJ. Krylov projection methods for model reduction. *PhD Thesis*, University of Illinois, Urbana-Champaign, 1997.
5. Freund RW. Krylov subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics* 2000; **123**(1):395–421. DOI: 10.1016/S0377-0427(00)00396-4.
6. Gallivan K, Vandendorpe A, Van Dooren A. Model reduction of MIMO systems via tangential interpolation. *SIAM Journal on Matrix Analysis and Applications* 2004; **26**(2):328–349. DOI: 10.1137/S0895479803423925.
7. Prajna S. POD model reduction with stability guarantee. *Proceedings of IEEE Conference on Decision and Control* 2003; **5**:5254–5258. DOI: 10.1109/CDC.2003.1272472.
8. Bond BN, Daniel L. Stable reduced models for nonlinear descriptor systems through piecewise-linear approximation and projection. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 2009; **28**(10):1467–1480. DOI: 10.1109/TCAD.2009.2030596.
9. Phillips JR, Daniel L, Silveira LM. Guaranteed passive balancing transformations of model order reduction. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 2003; **22**(8):1027–1041. DOI: 10.1109/TCAD.2003.814949.
10. Sorensen DC. Passivity preserving model reduction via interpolation of spectral zeros. *Systems & Control Letters* 2005; **54**(4):347–360. DOI: 10.1016/j.sysconle.2004.07.006.
11. Antoulas AC. A new result on passivity preserving model reduction. *System Control Letters* 2005; **54**(4):361–374. DOI: 10.1016/j.sysconle.2004.07.007.
12. Li R, Bai Z. Structure-preserving model reduction using a Krylov subspace projection formulation. *Communications in Mathematical Sciences* 2005; **3**(2):179–199.
13. Salimbahrami B, Lohmann B. Order reduction of large scale second-order systems using Krylov subspace methods. *Linear Algebra and its Applications* 2006; **415**(2-3):385–405. DOI: 10.1016/j.laa.2004.12.013.
14. Panzer H, Hubele J, Eid R, Lohmann B. Generating a parametric finite element model of a 3D cantilever Timoshenko beam using MATLAB. *Technical Report on Automatic Control*, Technical Munich University, 2009. **TRAC-4**.
15. Lasance CJM. Two benchmarks to facilitate the study of compact thermal modeling phenomena. *IEEE Transactions on Components and Packaging Technologies* 2001; **24**(4):559–565. DOI: 10.1109/6144.974943.
16. Rudnyi EB, Korvink JG. Thermal model. *Oberwolfach Model Reduction Benchmark Collection*, 2011. (Available from: <http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark>).
17. Baur U, Benner P. Model reduction for parametric systems via balanced truncation and interpolation. *Automatisierungstechnik* 2009; **57**(8):411–419. DOI: 10.1524/auto.2009.0787.
18. Feng LH, Rudnyi EB, Korvink JG. Preserving the film coefficient as a parameter in the compact thermal model for fast electrothermal simulation. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 2005; **24**(12):1838–1847. DOI: 10.1109/TCAD.2005.852660.
19. Nguyen TS, Bunse-Gerstner A. Error bounds for spline interpolation based parametric model order reduction. *Workshop on Model Reduction for Complex Dynamical Systems MODRED 2010*, Berlin, 2010.
20. Panzer H, Mohring J, Eid R, Lohmann B. Parametric model order reduction by matrix interpolation. *Automatisierungstechnik* 2010; **58**(8):475–484. DOI: 10.1524/auto.2010.0863.
21. Amsallem D, Cortial J, Carlberg K, Farhat C. A method for interpolating on manifolds structural dynamics reduced-order models. *International Journal for Numerical Methods in Engineering* 2009; **80**(9):1241–1258. DOI: 10.1002/nme.2681.



22. Degroote J, Vierendeels J, Willcox K. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis'. *International Journal for Numerical Methods in Fluids* 2010; **63**(2):207–230. DOI: 10.1002/fld.2089.
23. Amsallem D, Farhat C. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing* 2011; **33**(5):2169–2198. DOI: 10.1137/100813051.
24. Weile DS, Michielssen E, Grimme EJ, Gallivan K. Method for generating rational interpolant reduced-order models of two-parameter linear systems. *Applied Mathematics Letters* 1999; **12**(5):93–102. DOI: 10.1016/S0893-9659(99)00063-4.
25. Daniel L, Siong OC, Chay LS, Lee KH, White J. Multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 2004; **23**(5):678–693. DOI: 10.1109/TCAD.2004.826583.
26. Leung AT, Khazaka R. Parametric model order reduction technique for design optimization. *IEEE International Symposium on Circuits and Systems* 2005; **2**:1290–1293. DOI: 10.1109/ISCAS.2005.1464831.
27. Haasdonk B, Ohlberger M. Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Mathematical and Computer Modelling of Dynamical Systems* 2011; **17**(2):145–161. DOI: 10.1080/13873954.2010.514703.
28. Amsallem D, Farhat C. Interpolation method for adapting reduced-order models and application to aeroelasticity. *AIAA Journal* 2008; **46**(7):1803–1813. DOI: 10.2514/1.35374.
29. Lieu T, Farhat T. Adaptation of aeroelastic reduced-order models and application to an F-16 configuration. *AIAA Journal* 2007; **45**(6):1244–1257. DOI: 10.2514/1.24512.
30. Boothby WM. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press: London, 1975.
31. Ferrer J, García MI, Peurta F. Differentiable families of subspaces. *Linear Algebra and its Applications*. 1994; **199**:229–252. DOI: 10.1016/0024-3795(94)90351-4.
32. Edelman A, Arias TA, Smith ST. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications* 1998; **20**(2):303–353. DOI: 10.1137/S0895479895290954.
33. Absil PA, Mahony R, Sepulchre R. Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Applicandae Mathematicae* 2004; **80**(2):199–220. DOI: 10.1023/B:ACAP.0000013855.14971.91.
34. Gallivan K, Grimme E, Van D, P. Asymptotic waveform evaluation via a Lanczos method. *Applied Mathematics Letters* 1994; **7**(5):75–80. DOI: 10.1016/0893-9659(94)90077-9.
35. Feldmann P, Freund RW. Efficient linear circuit analysis by Padé approximation via Lanczos process. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 1995; **14**(5):639–649. DOI: 10.1109/43.384428.
36. Freund RW. Computation of matrix Padé approximations of transfer functions via a Lanczos-type process. In *Approximation Theory VIII , Vol. 1: Approximation and Interpolation*, Chui CK, Schumaker LL (eds). World Scientific Publishing Co., Inc.: Singapore, 1995; 215–222.
37. Patera AT, Rozza G. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations, 2007. MIT Pappalardo Graduate Monographs in Mechanical Engineering, V1.0.
38. Maday Y, Patera AT, Rovas DV. A blackbox reduced-basis output bound method for noncoercive linear problems. *Studies in Mathematics and Its Applications* 2002; **31**:533–569. DOI: 10.1016/S0168-2024(02)80025-X.
39. Sen S, Veroy K, Huynh DBP, Deparis S, Nguyen NC, Patera AT. Natural norm a posteriori error estimators for reduced basis approximations. *Journal of Computational Physics* 2006; **217**(1):37–62. DOI: 10.1016/j.jcp.2006.02.012.
40. Rozza G, Lassila TM, Manzoni A. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. *Lecture Notes in Computational Science and Engineering* 2011; **76**:307–315. DOI: 10.1007/978-3-642-15337-2-28.
41. Brand M. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and Its Applications* 2006; **415**(1):20–30. DOI: 10.1016/j.laa.2005.07.021.