6

A Robust Algorithm for Parametric Model Order Reduction Based on Implicit Moment Matching

Peter Benner and Lihong Feng

Abstract Parametric model order reduction (PMOR) has received a tremendous amount of attention in recent years. Among the first approaches considered, mainly in system and control theory as well as computational electromagnetics and nanoelectronics, are methods based on multi-moment matching. Despite numerous other successful methods, including the reduced-basis method (RBM), other methods based on (rational, matrix, manifold) interpolation, or Kriging techniques, multi-moment matching methods remain a reliable, robust, and flexible method for model reduction of linear parametric systems. Here we propose a numerically stable algorithm for PMOR based on multi-moment matching. Given any number of parameters and any number of moments of the parametric system, the algorithm generates a projection matrix for model reduction by implicit moment matching. The implementation of the method based on a repeated modified Gram-Schmidt-like process renders the method numerically stable. The proposed method is simple yet efficient. Numerical experiments show that the proposed algorithm is very accurate.

6.1 Introduction

The modeling of many engineering and scientific applications leads to dynamical systems depending on parameters varying in different design stages or computer experiments. For example, in a thermal model [16], the film coefficient k changes

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

L. Feng (✉)
Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany
e-mail: feng@mpi-magdeburg.mpg.de

A. Quarteroni, G. Rozza (eds.): *Reduced Order Methods for Modeling and Computational Reduction*,
MS&A 9. DOI 10.1007/978-3-319-02090-7_6, © Springer International Publishing Switzerland 2014

with the temperature, this results in a parametric mathematical model

$$C\frac{dx(t)}{dt}+Gx(t)+kDx(t)=B, \quad y(t)=L^{\mathrm{T}}x(t). \tag{6.1}$$

In integrated circuits, due to process variations, the width of the interconnects is in fact a random variable, such that a non-parametric model

$$C\frac{dx(t)}{dt}=Gx(t)+Bu(t), \quad y(t)=L^{\mathrm{T}}x(t), \tag{6.2}$$

is not sufficient to describe the random variation. Therefore, in [22,32], a linearized parametric system

$$\begin{aligned}(C_0+\lambda_1C_1+\lambda_2C_2)\tfrac{dx(t)}{dt} &= (G_0+\lambda_1G_1+\lambda_2G_2)x(t)+Bu(t),\\ y(t) &= L^{\mathrm{T}}x(t),\end{aligned} \tag{6.3}$$

is constructed. Here and below, the system matrices are $C, C_i, G, G_i \in \mathbb{R}^{n\times n}, i = 0,1,2$. $B \in \mathbb{R}^{n\times m_I}$ is the input matrix, $L \in \mathbb{R}^{n\times m_O}$ is the output matrix. $u(t) \in \mathbb{R}^{m_I}$ is the vector of input signals. $x(t) \in \mathbb{R}^n$ is the unknown vector. $y(t) \in \mathbb{R}^{m_O}$ is the vector of output responses. Many more examples for parametric systems can be found in the engineering literature, see, e.g., the benchmark examples in the recently published MOR wiki[1].

The above mentioned parametric systems are usually of very large dimensions as they often result from finite element discretizations of instationary partial differential equations (PDEs) defined on complex geometries. Solving the parametric systems by conventional simulation methods is often very time-consuming. On the one hand the parameters have to be provided as fixed values and these values cannot be changed during the simulation. On the other hand, if the dimension of the system is large, simulating such a system already once will be costly, and the cost of a design study requiring many runs with different parameter values ("many-query context") may be overwhelming.

Model order reduction (MOR) is an increasingly popular approach to overcome the obstacles posed by the computational demands in a many-query context. By MOR, a small dimensional approximate system can be derived, so that it can reliably replace the original system during the simulation. This can often save much simulation time and computer memory, see [2, 5, 6, 30] for some introductory texts on the topic and the presentation of the state-of-the-art.

The main goal of parametric model order reduction (PMOR) is to preserve parameters in the system as symbolic quantities in the reduced-order model. Thus, a change in parameters does not require to compute a new reduced-order model, but simply the evaluation of the reduced-order model for the new parameter values. If the error in the whole feasible parameter domain can be proven to satisfy an acceptable error tolerance, design and optimization of systems and devices can be significantly accelerated. First attempts at deriving MOR for linear parametric systems were based on extending the popular moment-matching methods (aka *Padé*

[1] See http://modelreduction.org.

approximation, Krylov subspace-based MOR methods) to parametric systems by multivariate power series expansions around appropriate interpolation points. Early references include [7, 8, 10, 16, 19, 21, 22, 27, 32, 34]. Later, other variants of (rational) interpolation techniques were derived, combining, e.g., balanced truncation and (sparse grid) interpolation [3], employing $\mathscr{H}_2$-optimal interpolation techniques [4], or using matrix and manifold interpolation techniques (e.g., [1,9,26]). Another large class of PMOR techniques is based on the Reduced Basis Method (RBM), originating in the fast approximation of parametric partial differential equations. The methods are also applicable in the context discussed here [20], but a dedicated comparison to the approaches mentioned here is deferred to future work. Therefore, we will not discuss this approach here any further and refer the reader to the survey [29] and other chapters in this volume.

In the following, we will discuss a robust implementation of the multi-moment matching methods first discussed in [7,8,19,34]. They have some advantages making them still the most popular approach used in practical applications:

- They are easy to implement and require almost no assumptions on system properties.
- Their cost is limited to a few (according to the number of employed expansion points) factorizations of sparse matrices and forward/backward solves using the computed factors. They do not require generation of trajectories and are therefore called "simulation-free" (in contrast to RBM and proper orthogonal decomposition (POD) methods). As a consequence, the "offline-phase" for computing the reduced-order model is cheap compared to RBM and POD, and it is often possible to achieve the goal encountered in practical industrial engineering design that the time for constructing the reduced-order model plus a simulation should be smaller than a single simulation of the full-order model.
- As they are simulation-free, no training inputs $u(t)$ need to be chosen so that the approximation quality is usually good for all feasible input signals, not only close to training inputs as in RB and POD methods.

Certainly, there are also some disadvantages: one has to first linearize parameter-dependencies (though polynomial forms are also possible, see, e.g., [10]), and the order of the reduced system may not be optimal. Nevertheless, improvements on these aspects are in progress, so that it is to be expected that multi-moment matching methods will remain competitive with other approaches in the future.

The MOR methods discussed here are based on projecting the unknown vector x onto a small dimensional subspace. We use system (6.2) to briefly introduce the concept. If a projection matrix $V \in \mathbb{R}^{n \times q}$ has been determined, using $x \approx Vz$ we obtain the perturbed system

$$CV\frac{dz(t)}{dt} = GVz(t) + Bu(t) + e(t), \quad \hat{y}(t) = L^{\mathrm{T}}Vz(t),$$

with $e(t)$ the introduced residual. By Galerkin projection $V^{\mathrm{T}}e(t) = 0$, we get the reduced-order model:

$$V^{\mathrm{T}}CV\frac{dz(t)}{dt} = V^{\mathrm{T}}GVz(t) + V^{\mathrm{T}}Bu(t), \quad \hat{y}(t) = L^{\mathrm{T}}Vz(t),$$

where $z(t) \in \mathbb{R}^q$ is the unknown vector of the reduced model. The space dimension q is often called the *order* of the reduced model. Therefore, the key step for MOR is how to get the projection matrix V, which determines the dimension and the accuracy of the reduced order model.

This paper is based on the ideas in [8], where the projection matrix V is obtained by computing an orthonormal basis of the subspace spanned by the moment vectors. No detailed algorithm of computing the orthonormal basis is proposed in [8]. A simple way of generating V is to first obtain the moment vectors by explicit matrix multiplications, and then all the columns of the computed moment vectors are orthonormalized to get the basis. However, this *explicit moment matching* procedure may lead to numerical instability, because higher order moment vectors usually become linearly dependent quickly as already observed in the non-parametric case, see [18]. We will demonstrate this effect in Sect. 6.2 for a practical example.

Our intention therefore is to develop an algorithm which computes the moment vectors implicitly rather than explicitly. In this way, good numerical stability can be preserved and an accurate orthonormal basis of the subspace spanned by the moment vectors can be obtained. The proposed algorithm can deal with both single-input and multiple-input systems without any limitation on the parameters in the system. It should be noted that this work dates back to first variants in 2007 [12, 13], and other comparable variants of implicit moment matching methods have been proposed [10, 21]. Here, we want to give a full account on the method discussed initially for only one parameter in [13].

In the following, we first review the method from [8] in Sect. 6.2 and explain the numerical instability resulting from explicit computation of the moments. In Sect. 6.3, we propose a numerical stable algorithm applicable to both single-input and multiple-input systems. The efficiency of the proposed algorithms is shown in Sect. 6.5 by simulating two examples from micro-electrical-mechanical systems (MEMS) and electrochemistry. Conclusions are given in the end.

6.2 Explicit Multi-Moment Matching PMOR

In this section, we give a short review of the method in [8] in order to explain the numerical instability of explicitly computing the moment vectors. A parametric system in time domain can be written as below,

$$\begin{aligned} C(s_1,s_2,\cdots,s_{p-1})\tfrac{dx}{dt}(t) &= G(s_1,s_2,\cdots,s_{p-1})x(t) + Bu(t), \\ y(t) &= L^{\mathrm{T}}x(t), \end{aligned} \tag{6.4}$$

where the system matrices $C(s_1,s_2,\cdots,s_{p-1})$, $G(s_1,s_2,\cdots,s_{p-1})$ are (maybe, non-linear, non-affine) functions of the parameters $s_1,s_2,\cdots,s_{p-1}$. A parametric system can also be stated in the frequency domain,

$$\begin{aligned} E(s_1,\ldots,s_p)x &= Bu(s_p), \\ y &= L^{\mathrm{T}}x, \end{aligned} \tag{6.5}$$

where the matrix $E \in \mathbb{R}^{n\times n}$ is parametrized. If the system in (6.5) is the Laplace transform of the system in (6.4), the new parameter s_p is in fact the frequency parameter s, which corresponds to time t. The state x is the Laplace transform of the unknown vector x in (6.4).

6.2.1 Review

The method in [8] is based on the representation of a parametric system in the frequency domain as in (6.5). In case of a nonlinear and/or non-affine dependence of the matrix E on the parameters, the system in (6.5) is first transformed to an affine form

$$\begin{aligned} (E_0+\tilde{s}_1E_1+\tilde{s}_2E_2+\ldots+\tilde{s}_pE_p)x &= Bu(s_p), \\ y &= L^{\mathrm{T}}x. \end{aligned} \tag{6.6}$$

Here the newly defined parameters $\tilde{s}_i, i=1,\ldots,p$, might be some functions (rational, polynomial) of the original parameters s_i in (6.5). To obtain the projection matrix V for the reduced model, the state x in (6.6) is expanded into a Taylor series at an expansion point $\tilde{s}_0=(\tilde{s}_1^0,\ldots,\tilde{s}_p^0)^T$ as below,

$$\begin{aligned} x &= [I-(\sigma_1M_1+\ldots+\sigma_pM_p)]^{-1}\tilde{E}^{-1}Bu(s_p) \\ &= \sum_{m=0}^{\infty}[\sigma_1M_1+\ldots+\sigma_pM_p]^m\tilde{E}^{-1}Bu(s_p) \\ &= \sum_{m=0}^{\infty}\sum_{k_2=0}^{m-(k_3+\ldots+k_p)}\cdots\sum_{k_{p-1}=0}^{m-k_p}\sum_{k_p=0}^{m}[F^m_{k_2,\ldots,k_p}(M_1,\ldots,M_p)B_Mu(s_p)\times \\ &\quad \sigma_1^{m-(k_2+\ldots+k_p)}\sigma_2^{k_2}\cdots\sigma_p^{k_p}, \end{aligned} \tag{6.7}$$

where $\sigma_i=\tilde{s}_i-\tilde{s}_i^0$, $\tilde{E}=E_0+\tilde{s}_1^0E_1+\ldots+\tilde{s}_p^0E_p$, $M_i=-\tilde{E}^{-1}E_i$, $i=1,2,\ldots p$, and $B_M=\tilde{E}^{-1}B$. The $F^m_{k_2,\ldots,k_p}(M_1,\ldots,M_p)$ can be generated recursively as

$$\begin{aligned} &F^m_{k_2,\ldots,k_p}(M_1,\ldots,M_p) \\ &=\begin{cases} 0, & \text{if } k_i\notin\{0,1,\ldots,m\}, i=2,\ldots,p, \\ 0, & \text{if } k_2+\ldots+k_p\notin\{0,1,\ldots,m\}, \\ I, & \text{if } m=0, \\ M_1F^{m-1}_{k_2,\ldots,k_p}(M_1,\ldots,M_p)+M_2F^{m-1}_{k_2-1,\ldots,k_p}(M_1,\ldots,M_p)+\ldots & \\ \quad\ldots+M_pF^{m-1}_{k_2,\ldots,k_p-1}(M_1,\ldots,M_p), & \text{else.} \end{cases} \end{aligned}$$

For example, if there are two parameters $\tilde{s}_1, \tilde{s}_2$ in (6.6), $F^m_{k_2,\ldots,k_p}(M_1,\ldots,M_p) = F^m_{k_2}$ are:

$$
\begin{aligned}
&F^0_0 = I,\\
&F^1_0 = M_1F^0_0 = M_1,\ F^1_1 = M_2F^0_0 = M_2 \qquad (6.8)\\
&F^2_0 = M_1F^1_0 = (M_1)^2,\ F^2_1 = M_1F^1_1 + M_2F^1_0 = M_1M_2 + M_2M_1,\ F^2_2 = M_2F^1_1 = (M_2)^2,\\
&\ldots
\end{aligned}
$$

For the general case, the projection matrix V is constructed as

$$
\begin{aligned}
&\operatorname{range}\{V\}\\
&= \operatorname{colspan}\{\bigcup_{m=0}^{m_q}\ \bigcup_{k_2=0}^{m-(k_p+\ldots+k_3)} \cdots \bigcup_{k_{p-1}=0}^{m-k_p}\ \bigcup_{k_p=0}^{m} F^m_{k_2,\ldots,k_p}(M_1,\ldots,M_p)B_M\}\\
&= \operatorname{colspan}\{B_M, M_1B_M, M_2B_M, \ldots, M_pB_M, (M_1)^2B_M, (M_1M_2+M_2M_1)B_M, \ldots,\\
&\qquad (M_1M_p+M_pM_1)B_M, (M_2)^2B_M, (M_2M_3+M_3M_2)B_M, \ldots\}.
\end{aligned} \tag{6.9}
$$

We call the coefficients in the series expansion of the state x in (6.7) the *moment vectors* of the parametric system. The corresponding moments of the transfer function are the moment vectors multiplied by L^{T} from the left. For example:

- $L^{\mathrm{T}}B_M$ is the 0th order moment; the columns in B_M are the 0th order moment vectors.
- Similarly, $L^{\mathrm{T}}M_iB_M,\ i = 1,2,\ldots,p$, are the first order moments, and the columns in $M_iB_M,\ i = 1,2,\ldots,p$, are the first order moment vectors, which are the coefficients of $\tilde{s}_i,\ i = 1,\ldots,p$.
- The columns in $M_i^2B_M,\ i = 1,2,\ldots,p$, $(M_1M_i+M_iM_1)B_M,\ i = 2,\ldots,p$, $(M_2M_i+M_iM_2)B_M,\ i = 3,\ldots,p,\ \ldots,\ (M_{p-1}M_p+M_pM_{p-1})B_M$ are the second order moment vectors, which are the coefficients of $\tilde{s}_i^2,\ i = 1,2,\ldots,p$, $\tilde{s}_1\tilde{s}_i,\ i = 2,\ldots,p$, $\tilde{s}_2\tilde{s}_i,\ i = 3,\ldots,p,\ \ldots,\ \tilde{s}_{p-1}\tilde{s}_p$.

Since by moments we not only denote the Taylor coefficients corresponding to the Laplace variable $s = s_p$, but also those associated with the other parameters $s_i, i = 1,\ldots,p-1$, we consider them as multi-moments of the transfer function. To sum up, the set of coefficients corresponding to terms with powers summing up to i is the set of the i-th order moment vectors. From the above construction of V, the subspace in (6.9) includes the 0-th order moment vectors till the m_q-th order moment vectors. The reduced model is computed as

$$
\begin{aligned}
(\hat{E}_0 + \tilde{s}_1\hat{E}_1 + \tilde{s}_2\hat{E}_2 + \ldots + \tilde{s}_p\hat{E}_p)z &= \hat{B}u,\\
\hat{y} &= \hat{L}^{\mathrm{T}}z,
\end{aligned} \tag{6.10}
$$

where $\hat{E}_i = V^{\mathrm{T}}E_iV,\ i = 0,1,2,\ldots,p, \hat{B} = V^{\mathrm{T}}B, \hat{L} = V^{\mathrm{T}}L$. Here we assume real expansion points, i.e., $\tilde{s}_i \in \mathbb{R}$ for all $i = 1,\ldots,p$. Otherwise, complex conjugate transposition might be needed to apply V from the left. This results in a reduced model with complex system matrices, which is undesired in some applications. Alternatives to obtain a real reduced order model even for complex expansion points exist [14], but we leave out these technical details for clarity of presentation.

In time domain, the reduced system (6.4) is

$$\begin{aligned} V^T C(s_1,s_2,\dots,s_{p-1})V\tfrac{dz}{dt} &= V^T G(s_1,s_2,\dots,s_{p-1})Vz+V^T Bu(t), \\ \hat{y}(t) &= L^{\mathrm{T}}Vz. \end{aligned} \tag{6.11}$$

Ideally, if the matrix V forms an orthonormal basis of the subspace in (6.9), the multi-moments of the reduced model in (6.10) match the multi-moments of the original system in (6.6) up to m_q-th order [8]. However, if V cannot be computed with sufficient numerical accuracy, the multi-moment matching property might be lost.

6.2.2 Analysis

Note that the subspace in (6.9) is not a Krylov subspace, therefore an orthonormal basis of the subspace spanned by the moment vectors cannot be computed by the standard Arnoldi algorithm. In [8], no algorithm for computation of the matrix V is presented. If the moment vectors are computed explicitly by simple matrix-matrix/vector multiplication, the high order moments will become linearly dependent, so that it is difficult or even impossible to obtain an orthonormal basis for the subspace considered.

We employ the thermal model (6.1) with parameter $k \in [1,10^9]$ (see Fig. 6.1) to illustrate this phenomenon. We observe the output of the system for $k = 10^9$. The moments vectors are first computed through explicit matrix multiplications (hence, explicit multi-moment matching)[2], then an orthogonalization process is applied to the moment vectors to get the final projection matrix V. Here we use the modified Gram-Schmidt process (with tolerance 10^{-11}) to get a V with orthonormal columns.

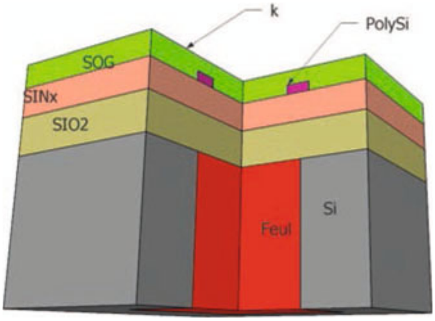

Fig. 6.1. Physical model of a microthruster unit for which a thermal MEMS model (6.1) is derived. Note that the film coefficient k is applied at the top

[2] Here we use a nonzero expansion point for the Laplace variable s, $s_0 = 0.001$, a zero expansion point for k, $k_0 = 0$, to ensure that the matrix $\tilde{E}$ is nonsingular. For all the simulation results in Sect. 6.5.1, the same expansion points are taken for all the tested MOR methods: the non-parametric moment-matching MOR, the explicit multi-moment matching and the proposed Algorithm 6.1.

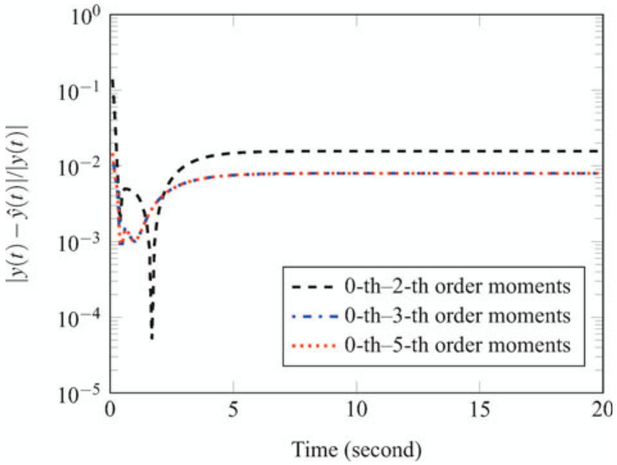

Fig. 6.2. Errors of the reduced models for the output responses of the thermal model (6.1), using explicit multi-moment matching (like in [28])

At first, we compute the moment vectors in (6.9) till the 2nd order to get the first reduced model. There are no vectors deleted during the orthogonalization process. The dashed line in Fig. 6.2 is the corresponding output error of the reduced model. If the moment vectors are computed up to order 3, the obtained second reduced model has smaller error. There is no deflation of the moment vectors either. If a more accurate model is to be derived, more moment vectors should be included. A third reduced model is obtained by computing the 0-th till the 5-th order moment vectors. This time, there are deflations during the modified Gram-Schmidt process. As a result, there is no increase in the number of the columns in the matrix V. The error of the reduced model is not further reduced, as can be seen from the dotted line in the figure. If the 6-th or higher order moment vectors are computed, the number of the columns in the matrix V still remains unchanged, and the accuracy of the corresponding reduced model cannot be improved.

The work in [11] first points out the numerical instability of explicitly computing the moments of the linear non-parametric system (6.2) in the method AWE [28]. It explains the numerical problem of AWE from the eigenvector and eigenvalue point of view. The moment vectors of the non-parametric system (6.2) are

$$G^{-1}B, G^{-1}CG^{-1}B, (G^{-1}C)^2G^{-1}B, \ldots, (G^{-1}C)^qG^{-1}B, \ldots.$$

These vectors are used to construct the projection matrix V and are computed explicitly in the method AWE. The computation of the kth moment vector $(G^{-1}C)^{k-1}G^{-1}B$ in fact corresponds to the power iteration $u_k = A^{k-1}b$, with $A = G^{-1}C$ and $b = G^{-1}B$ (we assume that B is a vector for simplicity). This process converges rapidly to an eigenvector of A associated to the eigenvalue of largest magnitude (assuming a simple eigenvalue). In the end, the computed vector u_k contains only information of this "dominant" eigenvector, and the later computed vectors are all numerically linearly dependent to this eigenvector. This explanation also applies to the numerical instability of the explicit computation of the moment vectors in (6.9). Some part of the moment vectors in (6.9) are also of the power iteration form. For example,

$$M_iB_M, M_i^2B_M, M_i^3B_M, \ldots, M_i^qB_M,\ i = 1,2,\ldots,p.$$

If directly computed, they quickly converge to the respective dominant eigenvector of each matrix $M_i,\ i = 1,2,\ldots,p$.

In the next section, a numerically stable algorithm for implicitly computing the moment vectors is presented. The algorithm is applicable for both single-input, single-output systems and multiple-input, multiple-output systems. An orthonormal basis of the subspace spanned by the moment vectors can be obtained implicitly so that a more accurate reduced model can be derived.

6.3 A Robust Algorithm for Multi-Moment Matching PMOR

Taking a closer look at the power series expansion of x in (6.7), we get the following equivalent, but different formulation,

$$\begin{aligned}
x &= [I-(\sigma_1M_1+\ldots+\sigma_pM_p)]^{-1}\tilde{E}^{-1}Bu\\
&= \sum_{m=0}^{\infty}[\sigma_1M_1+\ldots+\sigma_pM_p]^mB_Mu\\
&= B_Mu+[\sigma_1M_1+\ldots+\sigma_pM_p]B_Mu+[\sigma_1M_1+\ldots+\sigma_pM_p]^2B_Mu+\ldots\\
&\quad+[\sigma_1M_1+\ldots+\sigma_pM_p]^jB_Mu+\ldots
\end{aligned}\tag{6.12}$$

By defining

$$\begin{aligned}
x_0 &= B_M,\\
x_1 &= [\sigma_1M_1+\ldots+\sigma_pM_p]B_M,\\
x_2 &= [\sigma_1M_1+\ldots+\sigma_pM_p]^2B_M,\ldots,\\
x_j &= [\sigma_1M_1+\ldots+\sigma_pM_p]^jB_M,\ldots,
\end{aligned}$$

we have $x = (x_0+x_1+x_2+\cdots+x_j+\cdots)u$ and obtain the recursive relations

$$\begin{aligned}
x_0 &= B_M,\\
x_1 &= [\sigma_1M_1+\ldots+\sigma_pM_p]x_0,\\
x_2 &= [\sigma_1M_1+\ldots+\sigma_pM_p]x_1,\ldots,\\
x_j &= [\sigma_1M_1+\ldots+\sigma_pM_p]x_{j-1},\ldots.
\end{aligned}$$

If we define a vector sequence based on the coefficient matrices of $x_j,\ j = 0,1,\ldots$ as below,

$$\begin{aligned} R_0 &= B_M, \\ R_1 &= [M_1R_0, M_2R_0, \ldots, M_pR_0], \\ R_2 &= [M_1R_1, M_2R_1, \ldots, M_pR_1], \\ &\vdots \\ R_j &= [M_1R_{j-1}, M_2R_{j-1}, \ldots, M_pR_{j-1}], \\ &\vdots \end{aligned} \tag{6.13}$$

and let R be the subspace spanned by the vectors in $R_j,\ j = 0,1,\cdots,m$:

$$R = \text{colspan}\{R_0, \ldots, R_j, \ldots, R_m\},$$

we have $x \approx \hat{x} \in R$. We see that the terms in $R_j,\ j = 0,1,\ldots,m$ are the coefficients of the parameters in the series expansion (6.12). They are also the j-th order moment vectors.

The next step is to construct an orthonormal basis V of the subspace R by taking use of the recursive relations between the R_j in (6.13), such that the multi-moments of the original system are matched by those of the reduced model. A numerically stable algorithm for computing V is given in Algorithm 6.1. All the vectors included in R are orthogonalized to each other by the modified Gram-Schmidt (MGS) process once when constructed and then again after all R_j have been computed. In this sense, the algorithm can be understood as a *repeated MGS process*. There is no limitation on the number of parameters, and the essential cost of applying $\tilde{E}^{-1}E_j$ only grows linearly in the number of parameters, while the cost for orthogonalization step essentially grows quadratically with p.

Some remarks on Algorithm 6.1 are in order.

Remark 6.1 a) The application of $\tilde{E}^{-1}$ in Steps 2 and 18 is usually performed by computing once a (sparse) matrix factorization (Cholesky or LU, depending on the system structure) before the algorithm starts. Then each application of $\tilde{E}^{-1}$ means a forward/backward solve step. Hence, the whole algorithm requires only 1 matrix factorization, rendering it fairly cheap compared to other PMOR methods.

b) The application of E_t in Step 18 is a (usually sparse) matrix-vector multiplication and precedes the forward/backward solve step, which is then applied to the resulting vector E_tv_j using the precomputed factors of $\tilde{E}$.

c) For systems with multiple inputs, the input matrix B has more than one column. All the columns in $R_0 = \tilde{E}^{-1}B$ are orthogonalized in Step 5 before the columns in $R_i, i > 0$ are computed. The variable *sum* counts the number of columns in V.

d) m denotes the highest order of moments to be computed and is prescribed by the user.

Algorithm 6.1 Compute $V = [v_1, v_2, \ldots, v_q]$ for a parametric system (6.6), where B is generally considered as a matrix

```
Initialize $a_1 = 0$, $a_2 = 0$, $sum = 0$.
Compute $R_0 = \tilde{E}^{-1}B$.
**if** (multiple input) **then**
    Orthogonalize the columns in $R_0$ using MGS: $[v_1, v_2, \ldots, v_{q_1}] = \mathrm{orth}\{R_0\}$ with respect to a user given tolerance $\varepsilon > 0$ specifying the deflation criterion for numerically linearly dependent vectors.
    $sum = q_1$        % $q_1$ is the number of columns remaining after deflation w.r.t. $\varepsilon$.)
**else**
    Compute the first column in V: $v_1 = R_0/||R_0||_2$
    $sum = 1$
**end if**
% Compute the orthonormal columns in $R_1, R_2, \ldots, R_m$ iteratively as below
**for** $i = 1, 2, \ldots, m$ **do**
    $a_2 = sum$;
    **for** $t = 1, 2, \ldots, p$ **do**
        **if** $a_1 = a_2$ **then**
            $stop$
        **else**
            **for** $j = a_1 + 1, \ldots a_2$ **do**
                $w = \tilde{E}^{-1} E_t v_j$;
                $col = sum + 1$;
                **for** $k = 1, 2, \ldots, col - 1$ **do**
                    $h = v_k^T w$
                    $w = w - h v_k$
                **end for**
                **if** $\|w\|_2 > \varepsilon$ **then**
                    $v_{col} = \frac{w}{\|w\|_2}$;
                    $sum = col$;
                **end if**
            **end for**
        **end if**
    **end for**
    $a_1 = a_2$;
**end for**
Orthogonalize the columns in $V$ by MGS w.r.t. $\varepsilon$.
```

e) The index t is used to refer to computations related to the t-th parameter s_t corresponding to the coefficient $\tilde{E}^{-1}E_t$.

f) $a_2 - a_1$ is the number of columns added to V corresponding to R_{i-1}.

g) $a_2 - a_1 = 0$ means that all the vectors corresponding to R_{i-1} are deflated because they are linearly dependent (w.r.t. ε) to previous columns in V. In this case, there is no vector left which corresponds to R_{i-1}. As for a breakdown in a Krylov subspace method, we cannot continue to compute the columns in V corresponding to R_i, hence the algorithm stops.

h) In Step 17, j refers to the j-th column in V and corresponds to a vector in R_{i-1}.

i) Steps 20–27 implement the MGS process. col is the subscript of the current column v_{col} in V; it is orthogonalized to all the previous columns in V by MGS.

j) In Step 24, $||w||_2 < \varepsilon$ is the criterion used to deflate vectors in R_i that are linearly dependent (w.r.t. ε) to the previous vectors in V. It does not mean that all the vectors in R_i are linearly dependent on the previous vectors in V. If linear dependence is determined by this criterion, we delete the vector w and continue the algorithm till $a_1 = a_2$.

k) In Step 32, we orthonormalize all the columns in V again using MGS to reduce $||V^T V - I||_2$ (where I is the identity matrix of appropriate size) and to possibly further deflate columns. In this way, we perform a repeated MGS procedure. The final matrix V has q columns, which is equal to or less than the total number of vectors in R_i, $i = 0, 1, \cdots, m$.

l) When $p = 1$, the algorithm reduces to a block-Arnoldi-type process, with $R_0 = B_M$ being the starting block (the vectors in R_0 are the starting vectors), which can be used in moment-matching MOR for multiple input, non-parametric systems (see [17,25] for other variants of block Arnoldi processes used in moment-matching MOR).

It should be noted that analogously to moment-matching methods for non-parametric systems, a Petrov-Galerkin or oblique projection method can be constructed applying Algorithm 6.1 to B replaced by L and $\tilde{E}, E_t$ by $\tilde{E}^T, E_t^T$ (and not by complex conjugate transposition which would not yield the desired moment matching property). One would then obtain another orthogonal matrix W whose columns form an orthogonal basis of a complementary subspace. The reduced-order model is then computed by oblique projection $\hat{E}_t = W^T E_i V$, $t = 0, \dots, p$, etc., assuming the expansion point is chosen real. Technical issues as in standard oblique moment-matching methods will occur here even more pronounced, e.g., the number of computed columns for V and W may differ, the reduced-order model might loose stability, etc. We therefore restrict ourselves here to the presentation of the 1-sided (Galerkin/orthogonal) projection method to not obscure the presentation by too much technical details.

6.4 Multi-Moment Matching Property

In this section, we show that the reduced model obtained with the proposed Algorithm 6.1 has indeed the moment matching property derived in [8].

From the analysis in Sect. 6.3, the R_i defined in (6.13) are composed of the coefficients in the series expansion of the state x in frequency domain. The power series expansion of the transfer function of the original model (6.6) is, except for the left multiplication by L^T, the same due to the fact that for any feasible square-integrable function $u(\,.\,)$,

$$H(s_1,\ldots,s_p)u(s_p) = L^{\mathrm{T}}x(s_p).$$

(Note that x depends implicitly on $s_1,\ldots,s_{p-1}$ which we omit for the ease of notation.) Hence, the i-th order multi-moments of the parametric transfer function H are just the terms $L^{\mathrm{T}}R_i,\ i=0,1,2,\ldots$, where we recall that R_i includes the set of the i-th order moment vectors of x. For the reduced model in (6.10), there are corresponding power series expansions of the state z and the corresponding transfer function $\hat{H}$. We denote the coefficients in the series expansion of z as

$$\begin{aligned}
\hat{R}_0 &= \hat{B}_M, \\
\hat{R}_1 &= [\hat{M}_1\hat{R}_0,\hat{M}_2\hat{R}_0,\ldots,\hat{M}_p\hat{R}_0], \\
\hat{R}_2 &= [\hat{M}_1\hat{R}_1,\hat{M}_2\hat{R}_1,\ldots,\hat{M}_p\hat{R}_1], \\
&\ \vdots \\
\hat{R}_j &= [\hat{M}_1\hat{R}_{j-1},\hat{M}_2\hat{R}_{j-1},\ldots,\hat{M}_p\hat{R}_{j-1}] \\
&\ \vdots
\end{aligned}$$

where $\hat{E} = \hat{E}_0 + \tilde{s}_1^0\hat{E}_1 + \ldots + \tilde{s}_p^0\hat{E}_p$, $\hat{B}_M = \hat{E}^{-1}\hat{B}$, and $\hat{M}_i = -\hat{E}^{-1}\hat{E}_i,\ i=1,\ldots,p$. The transfer function of the reduced model can be expressed by z as

$$\hat{H}(s_1,\ldots,s_p)u(s_p) = \hat{L}^{\mathrm{T}}z(s_p).$$

Therefore, by the same variational argument as for the full-order system, the multi-moments of $\hat{H}$ are $\hat{L}^{\mathrm{T}}\hat{R}_i,\ i=0,1,2,\ldots$. Here, $\hat{E}_i,\ i=0,1,\ldots,p$, and $\hat{B},\hat{L}$ are defined in (6.10). Next we will prove that the multi-moments of $\hat{H}$ match the multi-moments of the original transfer function H. We summarize our analysis, using Lemma 6.1 and Lemma 6.2, in Theorem 6.1.

Suppose we construct the projection matrix V by

$$\mathrm{range}(V) = \mathrm{colspan}\{R_0,R_1,R_2,\cdots,R_m\} =: \wp.$$

The following Lemma 6.1 is used to prove Lemma 6.2 (Lemma 6.1 recalls a known fact and appears in several papers, see e.g. [8]).

Lemma 6.1 *If the column span of V forms an orthonormal basis of $\wp$, then for any vector $\xi \in \wp$,*

$$\xi = VV^{\mathrm{T}}\xi. \tag{6.14}$$

Lemma 6.2 *If the orthonormal projection matrix V satisfies* $\mathrm{range}(V) = \wp$,*, then* $\hat{R}_i = V^{\mathrm{T}}R_i,\ i=0,1,\ldots,m.$

Proof Recall that $\hat{E} = V^T\tilde{E}V$. Thus, for $i=0$,

$$\hat{E}V^{\mathrm{T}}R_0 = V^{\mathrm{T}}\tilde{E}VV^{\mathrm{T}}R_0.$$

Since $\mathrm{colspan}\{R_0\} \subseteq \wp$, we have $VV^{\mathrm{T}}R_0 = R_0$ by Lemma 6.1. Therefore, from the definition of R_0,

$$\hat{E}V^{\mathrm{T}}R_0 = V^{\mathrm{T}}\tilde{E}R_0 = V^{\mathrm{T}}\tilde{E}\tilde{E}^{-1}B = V^{\mathrm{T}}B = \hat{B}.$$

Hence, considering only the first and the last expression, we get,

$$V^{\mathrm{T}}R_0 = \hat{E}^{-1}\hat{B} = \hat{R}_0.$$

Thus, Lemma 6.2 is true for $i = 0$. Next, we assume that Lemma 6.2 is true for $i \leq j$, so that $\hat{R}_j = V^{\mathrm{T}}R_j$. We will prove that it is then also true for $i = j+1$. Since $\mathrm{colspan}\{R_{j+1}\} \subseteq \wp$, by Lemma 6.1 and the definition of R_{j+1}, we get

$$\begin{aligned}\hat{E}V^{\mathrm{T}}R_{j+1} &= V^{\mathrm{T}}\tilde{E}VV^{\mathrm{T}}R_{j+1}\\ &= V^{\mathrm{T}}\tilde{E}R_{j+1} = V^{\mathrm{T}}\tilde{E}[-\tilde{E}^{-1}E_1R_j, -\tilde{E}^{-1}E_2R_j, \ldots, -\tilde{E}^{-1}E_pR_j] \\ &= V^{\mathrm{T}}[-E_1R_j, -E_2R_j, \ldots, -E_pR_j].\end{aligned} \tag{6.15}$$

Because $\mathrm{colspan}\{R_j\} \subseteq \wp$, we know that $R_j = VV^{\mathrm{T}}R_j$ by Lemma 6.1. Hence, the last term of the above equation equals

$$V^{\mathrm{T}}[-E_1VV^{\mathrm{T}}R_j, -E_2VV^{\mathrm{T}}R_j, \cdots, -E_pVV^{\mathrm{T}}R_j]. \tag{6.16}$$

Therefore, by the definition of $\hat{E}_i$, $i = 1, \ldots, p$, and the assumption $\hat{R}_j = V^{\mathrm{T}}R_j$, (6.16) is equal to

$$[-\hat{E}_1\hat{R}_j, -\hat{E}_2\hat{R}_j, \cdots, -\hat{E}_p\hat{R}_j]. \tag{6.17}$$

Combining (6.15), (6.16) and (6.17), we obtain

$$\hat{E}V^{\mathrm{T}}R_{j+1} = [-\hat{E}_1\hat{R}_j, -\hat{E}_2\hat{R}_j, \cdots, -\hat{E}_p\hat{R}_j]. \tag{6.18}$$

Then from the definition of $\hat{R}_{j+1}$ we get

$$V^{\mathrm{T}}R_{j+1} = [-\hat{E}^{-1}\hat{E}_1\hat{R}_j, -\hat{E}^{-1}\hat{E}_2\hat{R}_j, \cdots, -\hat{E}^{-1}\hat{E}_p\hat{R}_j] = \hat{R}_{j+1}. \qquad \square$$

Theorem 6.1 *If V satisfies* $\mathrm{range}(V) = \mathrm{colspan}\{R_0, R_1, R_2, \cdots, R_m\}$*, then the multi-moments of the transfer function of the reduced model in* (6.10) *match those of the full system in* (6.6) *up to order m, i.e. $L^{\mathrm{T}}R_i = \hat{L}^{\mathrm{T}}\hat{R}_i$, $i = 0, 1, \ldots, m$.*

Proof From Lemma 6.2, and by the definition of $\hat{L}$, we have $\hat{L}^{\mathrm{T}}\hat{R}_i = L^{\mathrm{T}}VV^{\mathrm{T}}R_i$, $i = 0, 1, \ldots, m$. By Lemma 6.1, $VV^{\mathrm{T}}R_i = R_i$, therefore

$$\hat{L}^{\mathrm{T}}\hat{R}_i = L^{\mathrm{T}}R_i, \; i = 0, 1, \ldots, m. \qquad \square$$

6.5 Simulation Results

In this section, some simulation results are presented to show the efficiency of the proposed algorithm. We employ two examples, one being the thermal MEMS model considered before in Fig. 6.1 and the other one is from electrochemistry. Illustrated in Fig. 6.8 is the computational domain of the second model, where some chemical reactions take place.

6.5.1 Results for the thermal model

The thermal model is a generic example of a device with a single heat source, where the generated heat dissipates through the device to the surroundings. A heater is shown by the block made of PolySi. The exchange between surroundings and the device is modeled by convection boundary conditions with the film coefficient k at the top. The corresponding mathematical parametric model is given in (6.1), where k is the parameter. It is a single-input multiple-output system. For simplicity, we only observe a single output of the system, which is the temperature in the middle of the heater. As has been shown, the values of k change significantly, $k \in [1, 10^9]$. The size of the system is $n = 4725$.

To implement Algorithm 6.1, we first need to transform the system into the frequency domain by the Laplace transformation assuming $x|_{t=0} = 0$ for all k. The corresponding system in the frequency domain is

$$\begin{aligned}(sC + G + kD)X(s) &= BU(s),\\ y &= L^{\mathrm{T}}X(s),\end{aligned}$$

where s is considered as the second parameter of the system. Since B is a vector, the projection matrix V is constructed for the single input case in Algorithm 6.1.

Implicit vs. explicit moment vector computation

In Sect. 6.2.2, we have analyzed the accuracy of the PMOR method from [8] if the moment vectors are explicitly computed. Here, we show the efficiency of the proposed Algorithm 6.1, and compare it with the explicit moment-matching described in Sect. 6.2.2.

In Fig. 6.3, the errors of three reduced models computed by Algorithm 6.1 are plotted. The dashed line is the error of the output produced by the reduced model by matching the multi-moments upto order 2. The dash-dotted line is the error by matching multi-moments up to order 3. The dotted line is the one obtained by matching the multi-moments up to order 5. Different from the errors in Fig. 6.2, the errors of the reduced models keep decreasing with the increasing number of matched multi-moments, whereas the errors of the reduced models in Fig. 6.2 do not change after matching up to 3rd order moments. In Fig. 6.4, the accuracy of the reduced models computed by explicit and implicit moment-matching is compared. The solid line and the dashed line represent the accuracy of the reduced model computed by explicit moment-matching. By matching multi-moments up to the same order, the implicit moment-matching method implemented in Algorithm 6.1 is more accurate than the explicit moment-matching.

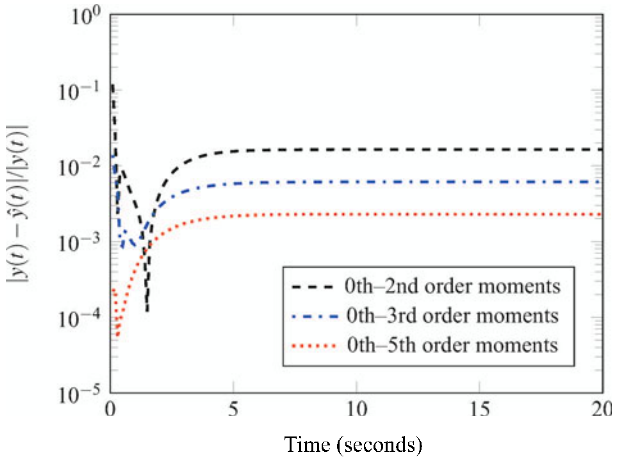

Fig. 6.3. Accuracy of Algorithm 6.1, the implicit multi-moment matching method

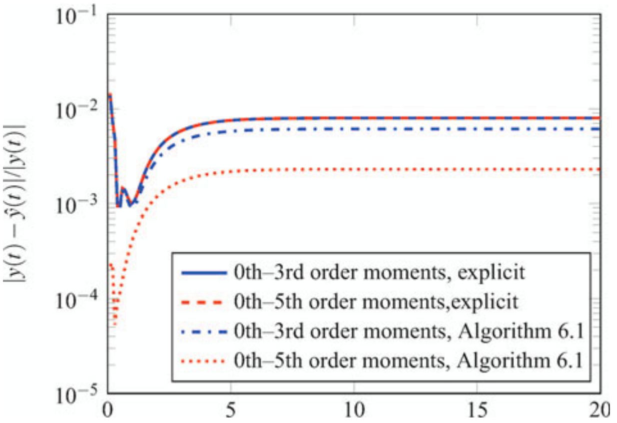

Fig. 6.4. Comparison between explicit and implicit multi-moment matching

PMOR vs. Non-Parametric MOR

In order to show the importance and the advantage of PMOR, we compare the proposed PMOR Algorithm 6.1 with the standard non-parametric moment-matching MOR method (see e.g. [25]). For non-parametric MOR, all the parameters except for the Laplace variable s must be fixed, such that the system becomes non-parametric. Hence, a standard non-parametric moment-matching method can be applied. Here the parameter k is fixed to $k = 1$, and the moments are the simple moments associated with the Laplace variable s. The reduced models constructed by both methods are in the same form as in (6.11).

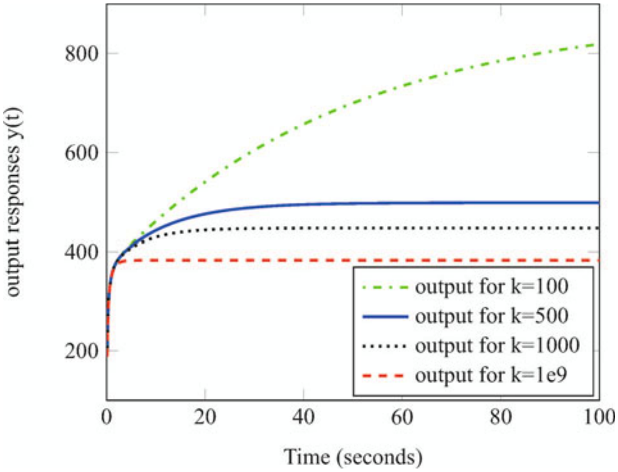

Fig. 6.5. Output responses of the original system (6.1) in the time-domain at different values of the parameter k

In Fig. 6.5, we plot the output responses corresponding to different values of k by simulating the original parametric system (6.1) for several times. We see that the time-dependent output response varies much with k.

In Algorithm 6.1, the 0th order till the 8th order multi-moments are matched. That is, range$(V) = \text{colspan}\{R_0, R_1, \dots, R_8\}$. The resulting reduced model is of order $q = 44$. For comparison, we could use the same order of moments associated with s for the non-parametric MOR. However, the resulting reduced model is only of dimension $q = 9$. Instead, the two methods are compared with respect to the same order of the reduced model. To this end, the 0th order till the 43rd order moments are matched by the non-parametric MOR method, and the reduced model is of the same dimension $q = 44$.

In Fig. 6.6, the relative errors of each reduced model changing with different values of k are plotted. Along the x-axis, the logarithm of the parameter k is taken. Along the y axis, the relative error defined as $||y(0,T;k) - \hat{y}(0,T;k)||_2/||y(0,T;k)||_2$ is plotted. Here $y(0,T,k) = (y(t_1;k),\dots,y(t_N;k))^{\mathrm{T}}$ is a vector of the output responses at different time steps in the interesting time interval, $t_i \in [0,T], i = 1,\dots,T_N$, for the current value of the parameter k, obtained by full simulation of the original system. The vector $\hat{y}(0,T;k)$ is obtained analogously from the output responses computed with the reduced model.

The solid line in the figure represents the errors produced by the reduced model with $q = 44$, obtained by non-parametric moment-matching MOR. It has good accuracy at the values of k close to $k = 1$, the fixed value. However, when the value of k grows, the error generally keeps increasing. As expected, the reduced model cannot catch the behavior of the output responses corresponding to values of k far away from the fixed value. The accuracy of the reduced model computed with the proposed PMOR method is much higher, though there is still a very slow trend of

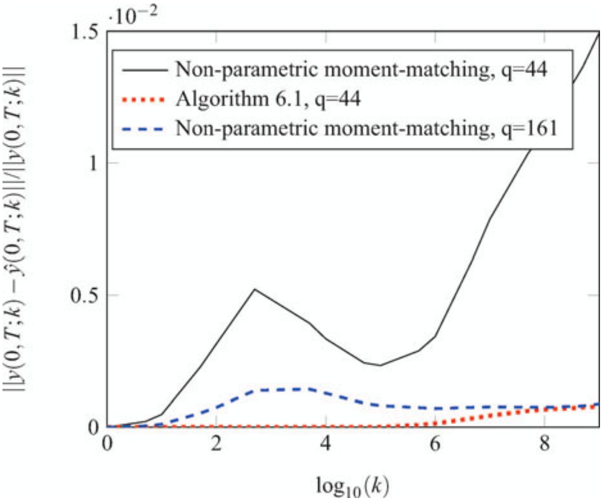

Fig. 6.6. Relative errors of the output responses of the thermal model in the time domain for different values of k, computed from the reduced models derived by non-parametric MOR and Algorithm 6.1, respectively. The orders of the three reduced models are $q = 44, 44, 161$, respectively

error increase with increasing value of k, see the dotted line. This is because a single expansion point for k, $k_0 = 0$, is used during the series expansion of the state vector x (see (6.7) and (6.12)). Multiple point expansion can be used in combination with Algorithm 6.1 to further decrease the error of the reduced model for very large values of k, see [14].

To achieve the same level of accuracy as for the reduced model resulting from PMOR, a reduced model with dimension $q = 161$ must be constructed with the non-parametric moment-matching MOR, where the 0th till the 160th order moments are matched. The error of the reduced model is plotted using dashes. This shows that the PMOR method provides a more compact reduced model over the entire parameter domain.

Robustness of the Proposed Algorithm

In Fig. 6.7, relative errors of three different reduced models constructed by Algorithm 6.1 are plotted. Each line represents the relative error between the output response of the reduced system and that of the original system according to different values of the parameter k. The definition of the relative error is the same as defined for Fig. 6.6. The line with the smallest error represents the error of the reduced system of order $q = 44$, for which the reduced system is obtained by $\mathrm{range}(V) = \mathrm{colspan}\{R_0, R_1, \ldots, R_8\}$. The line in the middle is the error of the reduced system with $q = 28$; it is derived from $\mathrm{range}(V) = \mathrm{colspan}\{R_0, R_1, \ldots, R_6\}$.

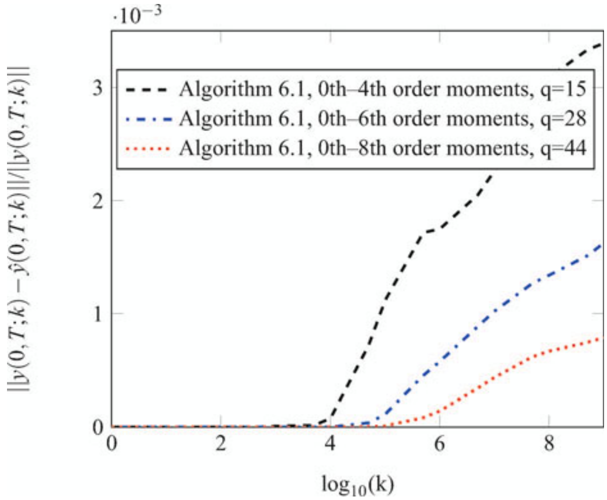

Fig. 6.7. Relative errors of the output responses of the thermal model in the time-domain computed from the reduced models with different order q, using Algorithm 6.1, for different values of k

The line on the top corresponds to the reduced system computed from range(V) = colspan$\{R_0, R_1, \ldots, R_4\}$. One can see that the error becomes smaller with increasing number of moment vectors used. All in all, the errors at all the values of the parameter k are very small, and satisfy the accuracy requirement in real applications. Compared with the explicit multi-moment matching, and the non-parametric MOR, the proposed Algorithm 6.1 produces a much more accurate reduced model.

6.5.2 Results for the electrochemistry model

The detailed description and derivation of the model for the application depicted in Fig. 6.8 is available from the MORwiki[3]. The mathematical model after spatial discretization is

$$\begin{aligned} E\tfrac{dc}{dt}+Gc+s_1D_1c+s_2D_2c=F, \quad c(0)=c_0\neq 0 \\ y=l^{\mathrm{T}}c,. \end{aligned} \tag{6.19}$$

The dimension of the system is $n = 16912$. Here, $E, G, D_1, D_2 \in \mathbb{R}^{n\times n}$ are system matrices. $l, F \in \mathbb{R}^n$ are constant vectors. $c(t) \in \mathbb{R}^n$ is the unknown vector. The two parameters $s_1 = e^{\beta u(t)}$, $s_2 = e^{-\beta u(t)}$ are functions of the voltage, where $\beta = 21.243036728240824$ is a constant. The voltage $u(t,\alpha)$ which is a function of

[3] http://morwiki.mpi-magdeburg.mpg.de/morwiki/index.php/Scanning_Electrochemical_Microscopy.

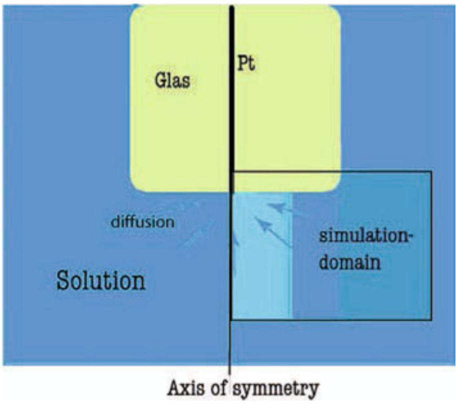

Fig. 6.8. Example from electrochemistry. The computational domain (indicated as simulation domain in the figure) under the 2D-axisymmetrical approximation includes the electrolyte under the electrode (the square at the middle top). Some chemical reactions take place in the computational domain. The interesting output is the total current over the electrode surface

time and α, follows a symmetric, triangular waveform:

$$\begin{aligned} u(t,\alpha) &= u_0 + \alpha t, \quad 0 < t < t_\alpha, \\ u(t,\alpha) &= u_0 - \alpha t, \quad t_\alpha < t < 2t_\alpha. \end{aligned}$$

Here, the variable α takes four possible values $\alpha = 0.5, 0.05, 0.005, 0.0005$. The time point t_α actually varies with α by $t_\alpha = 4 \times 10^i$, when $\alpha = 0.5 \times 10^{-i}$, for $i = 0, 1, 2, 3$.

The output $y(t)$ is the total current over the electrode surface, changing with the voltage $u(t,\alpha)$. The waveforms of the two parameters s_1 and s_2 as functions of time and the voltage $u(t,\alpha)$ are given in Fig. 6.9 and Fig. 6.10, respectively. Although both s_1 and s_2 are functions of the voltage u, hence are not independent, they are considered as two independent parameters in Algorithm 6.1. They can further be simply treated as two parameters independent of any argument, e.g. the time variable t, during the implementation of Algorithm 6.1, since the projection matrix V is generated independently of the parameters.

To deal with the system with nonzero initial condition, we employ the transformation method in [15]. That is, we first transform the system into a system with zero initial condition by $\tilde{c} = c - c_0$. The resulting transformed system is

$$\begin{aligned} E\tfrac{d\tilde{c}}{dt} + G\tilde{c} + s_1 D_1 \tilde{c} + s_2 D_2 \tilde{c} = F - Gc_0 - s_1 D_1 c_0 - s_2 D_2 c_0, \\ y = l^{\mathrm{T}}(\tilde{c} + c_0), \quad \tilde{c}(0) = c(0) - c_0 = 0. \end{aligned} \tag{6.20}$$

By Laplace transform, the above system in frequency domain becomes

$$\begin{aligned} (sE + G + s_1 D_1 + s_2 D_2)x &= \tilde{F} u(s), \\ y &= l^{\mathrm{T}}(x + c_0 u(s)), \end{aligned} \tag{6.21}$$

where x is the Laplace transform of the time-domain unknown vector $\tilde{c}$, $u(s) = 1/s$ is the Laplace transform of the constant 1, and $\tilde{F} = F - Gc_0 - s_1 D_1 c_0 - s_2 D_2 c_0$. As

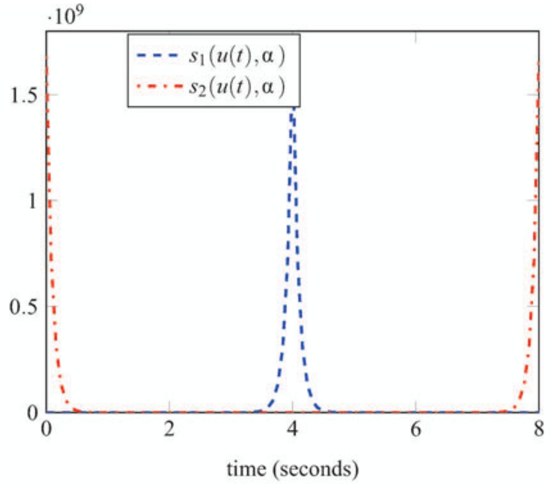

Fig. 6.9. $s_1(u(t,\alpha))$ and $s_2(u(t,\alpha))$ changing with time, $\alpha = 0.5$

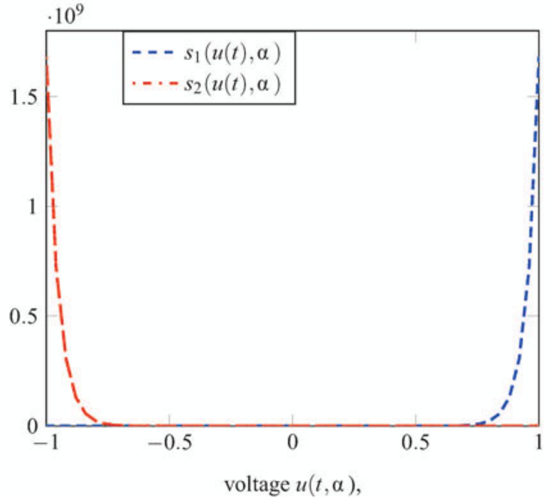

Fig. 6.10. $s_1(u(t,\alpha))$ and $s_2(u(t,\alpha))$ changing with the voltage $u(t,\alpha)$

explained above, $s_1(u(t))$ and $s_2(u(t))$ are treated as two constant parameters during the execution of Algorithm 6.1. As they are preserved in the reduced model, they can then again be varied with time according to their original definition when simulating the reduced model.

Note that the right-hand side of the system also depends on the two parameters s_1, s_2, which, however, is not a problem. Since the function $u(s)$ and the parameters

s_1, s_2 are both scalars, the right-hand side of the system (6.21) can be written as

$$(F - Gc_0 - s_1 D_1 c_0 - s_2 D_2 c_0)u(s) = [F - Gc_0, D_1 c_0, D_2 c_0] \begin{pmatrix} u(s) \\ -s_1 u(s) \\ -s_2 u(s) \end{pmatrix} = B\tilde{U},$$

where $B = [F - Gc_0, D_1 c_0, D_2 c_0]$, $\tilde{U} = [u(s), -s_1 u(s), -s_2 u(s)]^{\mathrm{T}}$. Therefore, the system in (6.21) can be considered as a multiple input system, so that the multiple input case in Algorithm 6.1 can be applied to construct the projection matrix V^4. The time domain reduced model in the form of (6.11) is obtained by applying Galerkin projection, using V, to the transformed system in (6.20) [15].

Figures 6.11–6.15 show the simulation results of the original model (6.19) and the reduced model obtained by Algorithm 6.1. The figures display the currents as functions of the voltages $u(t, \alpha)$, which is the usual way to represent the so-called cyclic *voltammograms* of the electro-chemical reaction. The solid line is the result obtained by full simulation of the original large model, the dashed line is the result computed using the small reduced model. The results of the reduced model are accurate for a wide range of the dynamic behavior when the value of α changes by three orders of magnitude (0.005–0.5).

The dashed lines in the Figs. 6.11–6.13 show the simulation results of three different reduced models with $\alpha = 0.5$. As we have already seen, the projection matrix V depends on the moment vectors of the system. If more moment vectors are used, the reduced model should become more accurate, at least in theory. The simulation

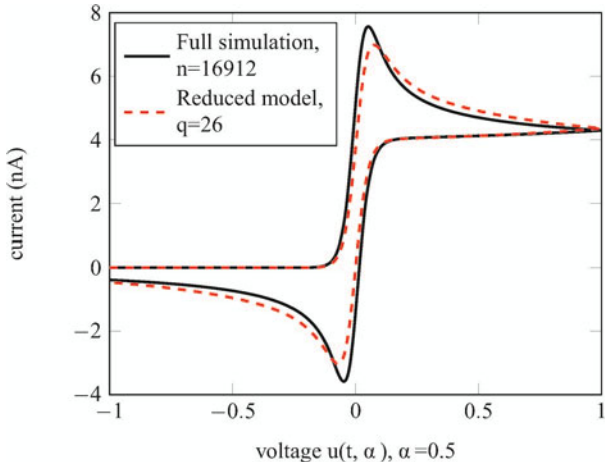

Fig. 6.11. The current y as a function of the voltage $u(t, \alpha)$, for $\alpha = 0.5$, for both the full simulation and the PMOR method Algorithm 6.1 using the multiple input variant. The moments are matched up to 4th order, yielding a reduced model of dimension $q = 26$

[4] For this example, the zero expansion points $s^0 = 0$, $s_1^0 = 0$ and $s_2^0 = 0$ are used for all the three parameters s, s_1, s_2 in (6.21).

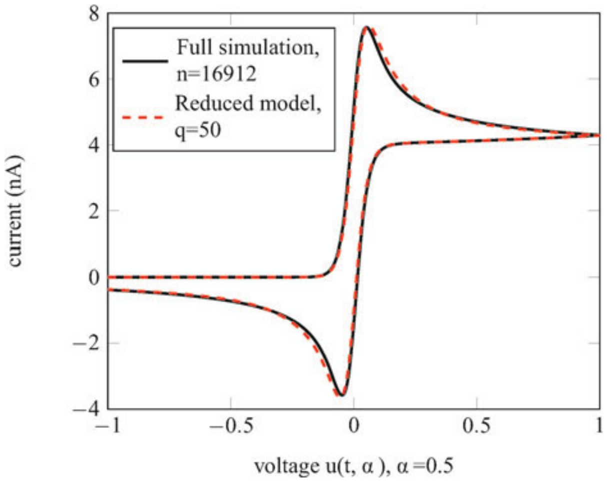

Fig. 6.12. The current y as a function of the voltage $u(t,\alpha)$, for $\alpha = 0.5$, for both the full simulation and the PMOR method Algorithm 6.1 using the multiple input variant. The moments are matched up to 6th order, yielding a reduced space of dimension $q = 50$

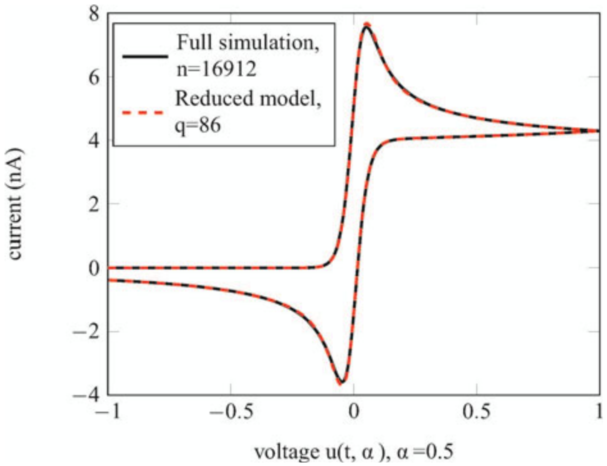

Fig. 6.13. The current y as a function of the voltage $u(t,\alpha)$, for $\alpha = 0.5$, for both the full simulation and the PMOR method Algorithm 6.1 using the multiple input variant. The moments are matched up to 9th order, yielding a reduced space of dimension $q = 86$

results in Fig. 6.3 show this fact for the previous thermal MEMS problem. For the current problem, the simulation results in Figs. 6.11–6.13 further justify it. In contrast, if V is computed by explicit matrix multiplications, the accuracy of the reduced model cannot be improved by using more moment vectors. In Fig. 6.11, the moment vectors from R_0 to R_4 are employed to compute V. In Fig. 6.12, R_0 till R_6 are used for the reduced model. The moment vectors from R_0 to R_9 are used in Fig. 6.13 to get V. The result in Fig. 6.13 is most accurate. The waveform of the current computed from the reduced model shows little difference from the solid line. In this case the order of the reduced model is $q = 86$. The relative error between the two currents is $\varepsilon|_{\alpha=0.5} = ||y-\hat{y}||_2/||y||_2 = 6.3 \times 10^{-4}$, where y is the vector of the current at dense samples of the interesting time interval by full simulation, and $\hat{y}$ is the vector of the current at the same samples obtained by simulating the reduced model. The reduced model is good enough to replace the original model with space dimension $n = 16912$ in practical applications of the model.

Figures 6.14–6.15 show additional outcomes for other values of α. Here we used the most accurate reduced model with range$(V) = $ colspan$\{R_0, R_1, \ldots, R_9\}$ and study the effect when varying α. The order of each reduced model is the same: $q = 86$.

The relative errors ε are listed in Table 6.1 for a selection of different values of α. All these simulation results show that accurate reduced models can be obtained with the proposed algorithm.

Table 6.1. ε vs. α

α	$\alpha = 0.5$	$\alpha = 0.05$	$\alpha = 0.005$	$\alpha = 0.0005$
ε	6.3×10^{-4}	1.8×10^{-5}	1.64×10^{-6}	1.38×10^{-6}

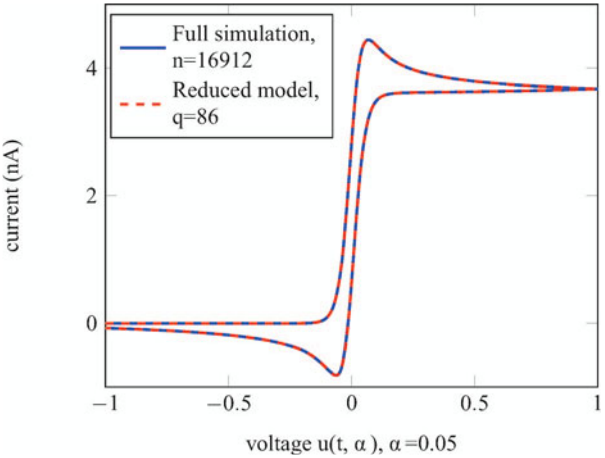

Fig. 6.14. The current y as a function of the voltage $u(t,\alpha)$, for $\alpha = 0.05$, for both the full simulation and the PMOR method Algorithm 6.1 using the multiple input variant. The moments are matched up to 9th order, yielding a reduced space of dimension $q = 86$

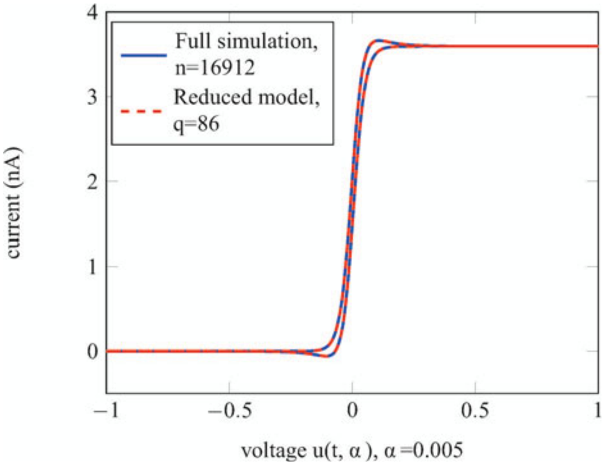

Fig. 6.15. The current y as a function of the voltage $u(t,\alpha)$, for $\alpha = 0.005$, for both the full simulation and the PMOR method Algorithm 6.1 using the multiple input variant. The moments are matched up to 9th order, yielding a reduced space of dimension $q = 86$

6.6 Conclusions

A numerical stable algorithm for PMOR is explored in this paper. The algorithm is used to construct a projection matrix V whose columns form an orthonormal basis of the subspace spanned by the moment vectors of the parametric system. Instead of explicit matrix-vector multiplications, a new moment vector is orthogonalized to all the previous ones during a (repeated) Modified Gram-Schmidt process. Numerical simulation results for both single input and multiple input parametric systems show that the proposed algorithm is very accurate and robust. Applications of the algorithm to parametric systems with more than three parameters can be found in [14].

The reduced parametric model can be used in optimization [35], in statistics [24], and in coupled simulations [23]. When used in statistics, it is important that quantities like mean and variance are well approximated. In applying PMOR for uncertainty quantification, one thus seeks to have a "statistics-preserving PMOR".

In some cases, the parameters may not be explicitly available. For instance, in modeling of electromagnetic problems, varying geometry may result in different meshes. For an approach to deal with this see [31].

Future research will focus on how to adaptively choose proper nonzero expansion points to attain a more compact model for systems with many (more than three) parameters. An error estimation for the state x of the parametric system is proposed in [33] for an automatic sampling selection. For many applications, the output y or the transfer function of the system is of interest, and an output-oriented error estimation for the proposed PMOR method is preferred, such that a more reliable reduced model can be obtained, automatically.

References

1. Amsallem, D., Farhat, C.: An online method for interpolating linear parametric reduced-order models. SIAM J. Sci. Comput. **33**(5), 2169–2198 (2011). DOI 10.1137/100813051
2. Antoulas, A.: Approximation of Large-Scale Dynamical Systems. SIAM Publications, Philadelphia, PA (2005)
3. Baur, U., Benner, P.: Model reduction for parametric systems using balanced truncation and interpolation. at–Automatisierungstechnik **57**(8), 411–420 (2009)
4. Baur, U., Benner, P., Beattie, C., Gugercin, S.: Interpolatory projection methods for parameterized model reduction. SIAM J. Sci. Comput. **33**, 2489–2518 (2011)
5. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation, Lecture Notes in Electrical Engineering, vol. 74. Springer-Verlag, Dordrecht, NL (2011)
6. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems, Lecture Notes in Computational Science and Engineering, vol. 45. Springer-Verlag, Berlin Heidelberg (2005)
7. Bond, B., Daniel, L.: Parameterized model order reduction of nonlinear dynamical systems. In Proc. International Conference on Computer-Aided Design pp. 487– 494 (2005)
8. Daniel, L., Siong, O., Chay, L., Lee, K., White, J.: A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **22**(5), 678–693 (2004)
9. Eid, R., Castañé-Selga, R., Panzer, H., Wolf, T., Lohmann, B.: Stability-preserving parametric model reduction by matrix interpolation. Math. Comp. Model. Dyn. Syst. **17**(4), 319–335 (2011)
10. Farle, O., Hill, V., Ingelström, P., Dyczij-Edlinger, R.: Multi-parameter polynomial order reduction of linear finite element models. Math. Comput. Model. Dyn. Syst. **14**(5), 421–434 (2008)
11. Feldmann, P., Freund, R.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **14**, 639–649 (1995)
12. Feng, L., Benner, P.: Parametrische Modellreduktion durch impliziten Momentenabgleich (Parametric Model Reduction by Implicit Moment Matching). In: B. Lohmann, A. Kugi (eds.) Tagungsband GMA-FA 1.30 “Modellbildung, Identifizierung und Simulation in der Automatisierungstechnik”, Workshop in Anif, 26.–28.9.2007, 34–47 (2007)
13. Feng, L., Benner, P.: A robust algorithm for parametric model order reduction based on implicit moment matching. Proc. Appl. Math. Mech. **7**, 1021.501–1021.502 (2008)
14. Feng, L., Benner, P., Korvink, J.G.: Subspace recycling accelerates the parametric macro-modeling of MEMS. Int. J. Numer. Meth. Engrg. **94**, 84–110 (2013)
15. Feng, L., Koziol, D., Rudnyi, E., Korvink, J.: Model order reduction for scanning electrochemical microscope: The treatment of nonzero initial condition. In Proc. Sensors pp. 1236–1239 (2004)
16. Feng, L., Rudnyi, E., Korvink, J.: Preserving the film coefficient as a parameter in the compact thermal model for fast electro-thermal simulation. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **24**(12), 1838–1847 (2005)
17. Freund, R.: Krylov subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**, 395–421 (2000)
18. Gallivan, K., Grimme, E., Van Dooren, P.: Asymptotic waveform evaluation via a lanczos method. Appl. Math. Lett. **7**(5), 75–80 (1994)

19. Gunupudi, P., Khazaka, R., Nakhla, M., Smy, T., Celo, D.: Passive parameterized time-domain macromodels for high-speed transmission-line networks. IEEE Trans. Microwave Theory and Techniques **51**(12), 2347– 2354 (2003)
20. 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 **17**(2), 145–161 (2011). DOI 10.1080/13873954.2010.514703
21. Li, Y., Bai, Z., Su, Y.: A two-directional Arnoldi process and its application to parametric model order reduction. J. Comput. Appl. Math. **226**(1), 10–21 (2009)
22. Liu, Y., Pileggi, L., Strojwas, A.: Model order reduction of RC(L) interconnect including variational analysis. In Proc. Design Automation Conference pp. 201–206 (1999)
23. Lutowska, A.: Model order reduction for coupled systems using low-rank approximations. Ph.D. thesis, Eindhoven University of Technology (2012)
24. ter Maten, E., Pulch, R., Schilders, W., Janssen, H.: Efficient calculation of uncertainty quantification. CASA-Repprt 2012-38 (2012). http://www.win.tue.nl/analysis//reports/rana12-38.pdf
25. Odabasioglu, A., Celik, M., Pileggi, L.: PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **17**(8), 645–654 (1998)
26. Panzer, H., Mohring, J., Eid, R., Lohmann, B.: Parametric model order reduction by matrix interpolation. at–Automatisierungstechnik **58**(8), 475–484 (2010)
27. Phillips, J.: Variational interconnect analysis via PMTBR. In Proc. International Conference on Computer-Aided Design pp. 872–879 (2004)
28. Pillage, L., Rohrer, R.: Asymptotic waveform evaluation for timing analysis. IEEE Trans. Comput.-Aided Design **9**, 325–366 (1990)
29. Rozza, G., Huynh, D., Patera, A.: Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. Arch. Comput. Methods Eng. **15**(3), 229–275 (2008). DOI 10.1007/s11831-008-9019-9
30. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model order reduction: Theory, research aspects and applications. Mathematics in Industry, vol. 13. Springer-Verlag, Berlin Heidelberg (2008)
31. Stavrakakis, K.: Model order reduction methods for parameterized systems in electromagnetic field simulations. Ph.D. thesis, TU Darmstadt (2012)
32. Tao, J., Zeng, X., Yang, F., Su, Y., Feng, L., Cai, W., Zhou, D., Chiang, C.: A one-shot projection method for interconnects with process variations. In Proc. International Symposium on Circuits and Systems pp. 333–336 (2006)
33. Villena, J.F., Silveira, L.M.: Multi-dimensional automatic sampling schemes for multi-point modeling methodologies. IEEE Trans. Comp-aid Design of Integrated Circuits and Systems **30**(8), 1141–1151 (2011). DOI 10.1080/13873954.2010.514703
34. Weile, D., Michielssen, E., Grimme, E., Gallivan, K.: A method for generating rational interpolant reduced order models of two-parameter linear systems. Appl. Math. Lett. **12**(5), 93–102 (1999)
35. Yue, Y.: The use of model order reduction in design optimization algorithms. Ph.D. thesis, KU Leuven (2012)