# Review: Automatic Model Reduction for Transient Simulation of MEMS-based Devices

*Evgenii B. Rudnyi and Jan G. Korvink*
*IMTEK—Institute for Microsystem Technology*
*Albert Ludwig University Freiburg*
*www.imtek.uni-freiburg.de/simulation*
*korvink@imtek.uni-freiburg.de*

preprint submitted to Sensors Update, Feb. 2002

## 1. Introduction

The goal of MEMS computer-aided design and simulation is to accurately and efficiently represent the behavior of the system in question. This allows technologists to develop a better understanding of the system, and as a result, to quickly choose an optimal design. A hugely successful example of the application of computer-aided design (CAD) is in the simulation of electrical integrated circuits, for which the simulator's output is almost the same as that produced by a real circuit prototype. This drives MEMS-designers to create similar techniques for MEMS simulations.

It so happens that electrical circuit and MEMS simulations are quite different in nature (see, for example, the discussion in Ref. [1]). A circuit is rather accurately described by lumped elements such as discrete resistors, capacitors, inductors, transistors and so on. The transient response of the circuit can be immediately written as a system of ordinary differential equations (ODEs) with the system's dimension of the order of the number of nodes connecting lumped elements in the circuit. On the other hand, the governing partial differential equations (PDEs) for MEMS-devices do not always lend themselves to intuitive lumping as ODEs, and hence are solved numerically by first spatially semi-discretising them by means of finite element, boundary element and similar methods. This also leads to a system of ODEs, but its dimension depends on the quality of discretization, and it could routinely lead to ODE system sizes of between ten thousand and a million equations, especially in the case of 3D simulations. The relation between differential equations, meshes and models are shown in Fig. 1.

Recent advances in computer power allows us to solve very large systems of ODEs by brute force, one of the most striking examples here being car crash simulations (see, for example, Ref. [2]). Nevertheless, this typically requires parallel computations (see the benchmark report in Ref. [3]) which increases the cost of simulation drastically, and as a result, limits simulation applicability considerably.

In order to facilitate computations, engineers often simplify the original rigorously derived governing equations or, instead, use simple empirical models: we use the term "quick-and-dirty" (QAD) calculations. Another approach, the topic of the present article, is to perform model reduction, that is, to formally reduce the dimension of a system of ODEs derived from a rigorous approach before integrating it in time.

To this end, taken from current mechanical engineering practice, there are two popular methods, and both are incorporated in some commercial software simulation tools: modal reduction [4] and dynamic condensation [5]. The idea behind modal reduction is to approximate a dynamic system response through a linear combination of several, often low-frequency, natural eigenmodes of the system. The second approach is based on the Guyan method [6], and is just an intuitive engineering extension of the Shur complement method from a stationary to a time-dependent formulation.

The main problem with all of the above order reduction methods is that their success primarily depends on engineering intuition, since they are not based on a solid mathematical background. Hence, they could be referred to as non-automatic model reduction methods, and there appears to be no way to improve this situation. Certainly, without experience and intuition, we do not recommend their use.

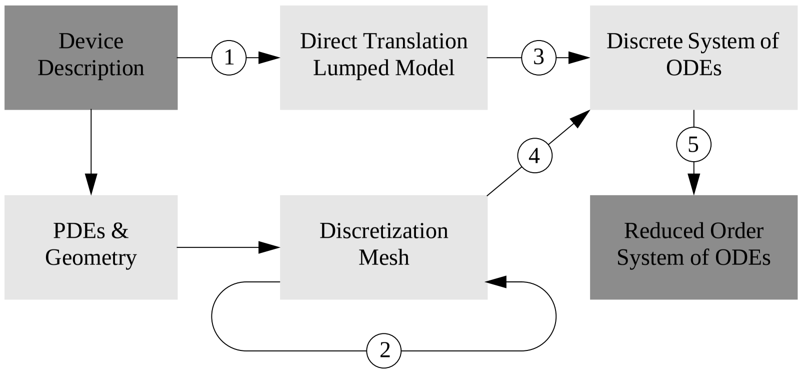

*Fig. 1 Some of the routes leading from a device description to a reduced order system of ODEs. The arrows represent translations between descriptions: 1) Lumping is done by hand, either as a circuit equivalent, or as an algebraic expression; 2) Adaptive meshing determines the size of the subsequent model; 3) Circuit equivalents or algebraic expressions are turned into a suitable set of ODEs; interconnecting many of these again lead to large systems; 4) Semidiscretization of the PDEs on a mesh result in a set of ODEs; 5) An algebraic model reducer take a large system of ODEs and produces a smaller (and hence reduced order) yet equivalent system of ODEs.*

On the other hand, model reduction has received a great deal of attention from mathematicians, who have developed a number of methods with which to approximate large-scale dynamics systems (for a mathematical review, see Ref. [7]), and which will be referred to as automatic model reduction. There are some spectacular examples where the dimension of ODEs could be reduced by several orders of magnitude, almost without sacrificing precision, see e.g. [8] and [9]. However, there still remains a certain gap between these ideas and common MEMS engineering practice, and the aim of the present review is to start to fill this gap. Our review complements Ref. [7] (where automatic model reduction is considered mathematically) on the engineering level.

Nevertheless, the classification in this review is made on the basis of a mathematical perspective and therefore follows the structure of Ref. [7]. What we have found is that, even though different engineering communities are facing quite different challenges, many solution techniques are related. At a first glance, the simulation of groundwater flow in discretely fractured porous media has nothing to do with MEMS-devices. It is therefore not surprising that these two engineering communities do not follow each other's work. However, the model reduction problem they are trying to solve is absolutely the same if we consider it from a mathematical viewpoint.

In principle, a system of ODEs can also be solved faster if it is possible to increase the efficiency of the time integrator. Recently, there have been some promising results in this direction based on matrix exponential approximations [10], but so far there are no engineering examples, and hence this will be outside of the scope of our review.

We start our review with a statement of the mathematical problem for model reduction, where we introduce terms and give them the equivalents used in the MEMS community. Then we consider low-dimensional linear systems of ODEs. It is safe to state that, for this case, the problem of automatic model reduction is almost completely solved. It appears that almost all modern model reduction methods for large-scale systems are based, in one way or another, on Krylov subspace methods [11], and therefore a short introduction to Krylov subspace methods is given. After that, we switch to large-scale linear systems of ODEs. The challenge faced here is that the computational time required for a model reduction of a linear system of ODEs depends on the problem dimension (the number of equations in the system) to the cubic power. Computationally speaking, the algorithms for model reduction appropriate for small linear systems do not scale to large systems. Here one can say that, in principle, the answer to automatic model reduction is known but the challenge remains as to how to compute it in reasonable time. Finally, we take a look at non-linear systems of ODEs. Here success depends on a particular problem, and there are almost no general results. Some algorithms for model reduction exist but, in contrast to linear systems, unfortunately, it seems that human intervention is inevitable.

It should be noted that we have not tried to reflect the priority of research groups in this field. In many cases, our citations should actually be read as “see, **for example,** Ref … ”.

## 2. Mathematical Statement for Model Reduction

In the present review we limit our consideration to a system of first order ODEs, written in the form

$$E \cdot \frac{d\boldsymbol{x}}{dt} = F \cdot \boldsymbol{x} + \boldsymbol{f} \tag{1}$$

where the unknown vector $\boldsymbol{x}(t) \in \Re^n$ contains unknowns functions in time, $E \in \Re^n \times \Re^n$ and $F \in \Re^n \times \Re^n$ are system matrices, typically sparse and often symmetric, and the vector $\boldsymbol{f} \in \Re^n$ describes the system load. If the matrices contain constant coefficients then the system of ODEs is linear, and otherwise we will call it nonlinear. (Strictly speaking this is not correct. There is an intermediate case when coefficients depend on time explicitly in which case it is termed a linear time-varying system [12]). Mechanical systems in motion, as well as general electrical circuits, are usually described by systems of ODEs of second order in time. It is a simple matter to convert them to the form of Eq (1) by increasing number of unknowns and equations by a factor of two, e.g., by treating the first derivatives in time as unknown. Thus

$$M \cdot \frac{d^2\boldsymbol{y}}{dt^2} + C \cdot \frac{d\boldsymbol{y}}{dt} + K \cdot \boldsymbol{y} = \boldsymbol{f} \tag{2}$$

together with the new variables $\boldsymbol{z} = d\boldsymbol{y}/dt$, becomes

$$\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \cdot \frac{d}{dt}\begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} = -\begin{bmatrix} C & K \\ -I & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} + \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix} \tag{3}$$

which is again in the form of Eq (1). In some cases the methods treated in the review can be generalized to second order systems of ODE directly.

The naming of system matrices as well as the notation is quite different for different engineering disciplines, but we hope that this does not pose an insurmountable problem. In order to perform a model reduction step, we rewrite Eq (1) from an implicit to an explicit system of ODEs

$$\frac{d\boldsymbol{x}}{dt} = A \cdot \boldsymbol{x} + \boldsymbol{b} \tag{4}$$

where

$$A = E^{-1} \cdot F,\ A \in \Re^n \times \Re^n \text{ and } \boldsymbol{b} = E^{-1} \cdot \boldsymbol{f},\ \boldsymbol{b} \in \Re^n \tag{5}$$

It is necessary to stress that Eq (5) should be read in a mathematical, and not in a computational sense. Mathematically this implies that matrix $E$ is not degenerate (i.e., it is invertible) and that this transformation is possible in principle. If matrix $E$ is degenerate then we do not have a system of ODEs, but rather a system of algebraic-differential equations (ADEs). From a computational point of view the operations in Eq (5) are highly disadvantageous: first, they are prohibitively expensive for large-scale systems, second, they destroy the sparsity of the original matrices. In other words, computationally it is necessary to work with the two original sparse matrices. The question on how to effectively compute Eq (5) for the case of Krylov subspace methods is discussed in Section 4.2.

The main problem with Eq (4) is the high dimensionality of the vector $\boldsymbol{x}$, which is typically equal to the product of the number of unknowns in a system of PDEs to be solved by the number of nodes introduced during the discretization process. This in turn leads to the high dimension of system matrices and finally to the huge computational cost to solve the system’s response.

In performing model reduction on Eq (4), the hope is that, for many systems of ODEs of practical importance, the behavior of vector $\boldsymbol{x}$ in time, $\boldsymbol{x}(t)$, is effectively described by some low-dimensional subspace as follows

$$\boldsymbol{x} = X \cdot \boldsymbol{z} + \boldsymbol{\varepsilon},\ \boldsymbol{z} \in \Re^k,\ k \ll n \tag{6}$$

Eq (6) states that, with the exception of a small error described by vector $\boldsymbol{\varepsilon} \in \Re^n$, the possible movement of the $n$-dimensional vector $\boldsymbol{x}$ belongs, for all times, to a $k$-dimensional subspace, with $k$ much smaller than $n$, and is determined by an $n \times k$ transformation matrix $X$. The matrix $X$ is composed from $k$ $n$-dimensional vectors that form a basis for the reduced subspace, and the $k$-dimensional vector $\boldsymbol{z}$ represents a new low order set of coordinates for the given basis.

The task of model reduction is to find such a subspace for which the error difference in Eq (6) is minimal according to some norm

$$min\|\boldsymbol{\varepsilon}(t)\| = min\|\boldsymbol{x}(t) - X \cdot \boldsymbol{z}(t)\| \tag{7}$$

Note that in Eq (7), we have functions in time, so that the norm in this case is represented by some integral over time [13]. When the subspace is found, Eq (4) should be projected onto it, and this projection process produces a system of ODEs of reduced order $k$

$$\frac{d\boldsymbol{z}}{dt} = \hat{A} \cdot \boldsymbol{z} + \hat{\boldsymbol{b}} \tag{8}$$

which can then be used later on, perhaps in another simulation package.

The physical background for model reduction so far is that the discretization grid used to solve the original PDEs is far from an optimal basis to represent the solution of the PDEs. From this point of view, the model reduction according to Eq (7) is, in a sense, similar to adaptive grid generation [14]. However, the opportunities of model reduction to minimize the problem dimensionality are much greater, because adaptive grid generation still deals with local shape functions (with local support), and the basis for the low-dimensional subspace in Eq (6) is formed from global domain functions, that is, each vector includes a contribution from the entire geometrical domain (much as eigenvectors do). Form this point of view, model reduction complements adaptive grid generation, or makes an alternative in a sense as will now be described.

An adaptive grid generation process starts with some initial grid, and then the grid in different parts of the computational domain gets refined or coarsened based on a priori or a posteriori local error estimators [15]. A model reduction strategy requires a fine initial grid, for which it produces an effective global low-dimensional basis, based on global error estimators. Then, in order to choose the best computational strategy, it is necessary to compare the time taken for model reduction of a system of ODEs built on the fine grid with the sum of times for adaptive grid generation and the subsequent model reduction of the refined grid system of ODEs.

We now take the next step and put the model reduction problem into a more general form. Often, engineers are not interested in the solution of Eq (4) over the entire computational domain, that is, for values at all nodes, but rather in only a few of their combinations. Control theorists [16] take this into account and convert Eq (4) to

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = A \cdot \boldsymbol{x} + B \cdot \boldsymbol{u} \\ \boldsymbol{y} = C \cdot \boldsymbol{x} \end{cases} \tag{9}$$

Equation (9) treats the system as a “black box”, which would the case when a system’s high-dimensional internal state vector $\boldsymbol{x}$, governed by ODEs, is not directly accessible to an external observer. The observer can influence the system state by some input functions, specified by the vector $\boldsymbol{u} \in \Re^m$, and which are distributed to the internal nodes in accordance to the *scatter* matrix $B \in \Re^n \times \Re^m$. The number of input signals $m \ll n$ is typically small, and this means that matrix $B$ has a small number of columns. On the other hand, the observer is interested in only a few outputs, specified by vector $\boldsymbol{y} \in \Re^p$ with the dimension $p \ll n$. The relationship between required outputs and the system state is given by the *gather* matrix $C \in \Re^p \times \Re^n$. As a result, we have a high-dimensional system of ODEs in relation to vector $\boldsymbol{x}$, the system state vector, which is governed by a small number of external inputs, and from the viewpoint of an external observer, contains a small number of relevant outputs. We will not describe here the well-known system-theoretic results of this equation, such as zero state and zero input, but refer the curious reader to the control theory literature [16].

Eq (9) is a generalization of Eq (4). If matrix $B$ in Eq (9) represents a single vector, equal to vector $\boldsymbol{b}$ of Eq (1), then vector $\boldsymbol{u}$ will contain only one element, a single input, and we can equate it to a step function. Now let us say that matrix $C$ is an identity matrix, that is, $\boldsymbol{y} = \boldsymbol{x}$, then we have a special case w. r. t. the original system of ODEs, which we call “single input - *complete* output” or SICO.

The multiple input case holds when matrix $B$ has several columns corresponding to multi-load simulations, or when the system is consecutively subject to a variety of loads distributed to different nodes. In this case, each function in vector $\boldsymbol{u}$ has a “step” shape limited by the application time of the load (see Fig. 2).

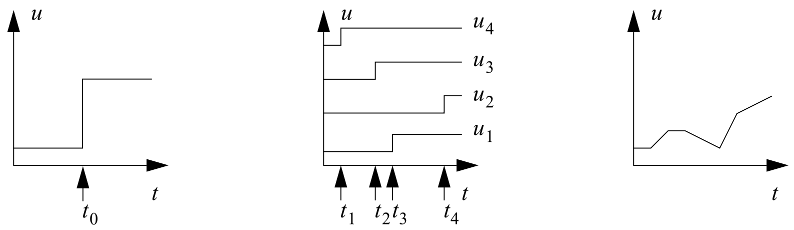

*Fig. 2 Different input functions, often provided in engineering simulation programs. a) A step function activating at $t_0$. b) A vector of step functions, each activating at a different time. c) A piecewise linear function.*

Matrix $C$ is usually formed by picking only those rows from the unit matrix which correspond to chosen nodes. In this case, vector $\boldsymbol{y}$ is just a small subset of the state vector $\boldsymbol{x}$.

The problem of model reduction in the case of Eq (9) consists in the reduction of the dimension of the state vector to order $k \ll n$, while retaining the same number of inputs and outputs

$$\begin{cases} \dfrac{d\boldsymbol{z}}{dt} = \hat{A} \cdot \boldsymbol{z} + \hat{B} \cdot \boldsymbol{u} \\ \hat{\boldsymbol{y}} = \hat{C} \cdot \boldsymbol{z} \end{cases} \tag{10}$$

The input vector $\boldsymbol{u}$ in Eq (10) is exactly the same as in Eq (9), but the output vector $\hat{\boldsymbol{y}} \in \Re^k$ is just some approximation of the original vector $\boldsymbol{y} \in \Re^k$. This transformation is sketched in Fig. 3.

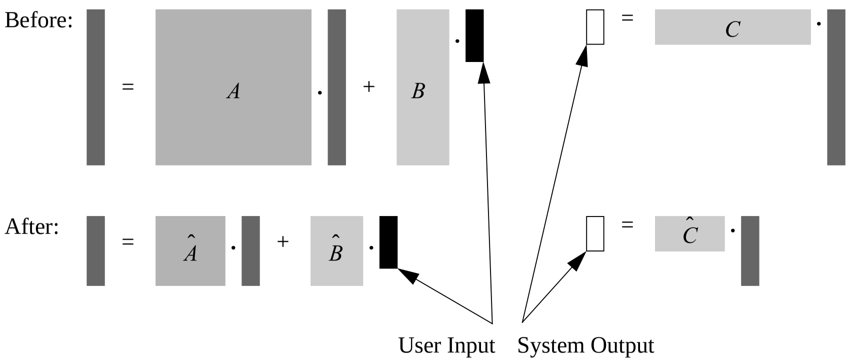

*Fig. 3 Sketch of the model reduction equations (9) before and (10) after the model reduction step. The dimensions of the system matrices $A$, $B$ and $C$ and the internal state vectors $\boldsymbol{z}$ and $\boldsymbol{z}$ are significantly smaller after model reduction. The input vector $\boldsymbol{u} \in \Re^m$ and output vector $\boldsymbol{y} \in \Re^k$ remain the same size.*

The quality of the model reduction step of Eq (10) is determined by a norm

$$min\|\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)\| = min\|\boldsymbol{y}(t) - \hat{C} \cdot \boldsymbol{z}(t)\| \tag{11}$$

which ideally should hold for any input vector $\boldsymbol{u} \in \Re^m$. The difference between Eq (11) and Eq (7) is that now we search for a reduced subspace given by Eq (6) to minimize the difference between given outputs only, and not for the whole state vector. Certainly, if we have found a subspace that minimizes Eq (7), then Eq (11) will be satisfied automatically. However, we expect that a subspace minimizing Eq (11) will have a much lower dimension than a subspace minimizing Eq (7).

If matrices $B$ and $C$ both consist of a single column and row respectively then the system is termed Single-Input-Single-Output (SISO), otherwise it is referred to as Multiple-Input-Multiple-Output (MIMO).

A dynamic system is often considered in the frequency domain, when the Laplace transform operator $L\{.\}$ is applied to the input and output vectors [13]

$$L\{\boldsymbol{y}(t)\} = \boldsymbol{Y}(s), L\{\boldsymbol{u}(t)\} = \boldsymbol{U}(s) \tag{12}$$

and where the relationship between input and output is described by the transfer function

$$\boldsymbol{Y}(s) = G(s) \cdot \boldsymbol{U}(s) \tag{13}$$

Most of the results in model reduction obtained so far concern the case of a linear system of ODEs and where all the matrices of Eqs (4) and (9) are composed of constant numbers. In this case, the transfer function is readily expressed via the system matrices as

$$G(s) = C \cdot (sI - A)^{-1} \cdot B \tag{14}$$

## 3. Small Linear Systems

Control theory has a very strong theoretical result for stable systems, *i. e.*, those systems for which the real parts of all the eigenvalues of the system matrix $A$ in Eq (9) are negative. Each linear dynamic system (9) has $n$ so-called Hankel singular values, $\sigma_i$ (see Ref. [17] for mathematical details), which can be computed if one solves two Lyapunov equations

$$A \cdot P + P \cdot A^T = -B \cdot B^T \tag{15}$$

$$A^T \cdot Q + Q \cdot A = -C^T \cdot C \tag{16}$$

for the controllability grammian $P$ and observability grammian $Q$. Then the Hankel singular values of the original dynamic system are equal to the square root of the eigenvalues of the product of the controllability and observability grammians

$$\sigma_i = \sqrt{\lambda_i(P \cdot Q)} \tag{17}$$

Once these values are known, there are a number of model reduction methods with guaranteed error bounds for the difference between the transfer function of an original $n$-dimensional system and its reduced $k$-dimensional system, as follows

$$\|G - \ddot{G}\|_\infty \le 2(\sigma_{k+1} + \ldots + \sigma_n) \tag{18}$$

provided that the Hankel singular values have been sorted in descending order. Note that this equation is valid for arbitrary input functions. This means that model reduction based on these methods can be made fully automatic. A user just sets an error bound and then, by means of Eq (18), the algorithm finds the smallest possible dimension of the reduced system, $k$, which satisfies that bound. Alternatively, a user specifies the dimension of the reduced system and the algorithm estimates the error bound for the reduced system.

Another practical consequence of this result is that the success of model reduction depends only on the decay rate of the Hankel values. Fig. 4 shows examples of the behavior of Hankel values for a few typical applications. If we can estimate this decay rate for a particular application, this would give us a complete answer as to the extent to which we could reduce the original system [18][19].

The SLICOT library implements three methods, a Balanced Truncation Approximation, a Singular Perturbation Approximation and the Hankel-Norm Approximation, as well as including a special benchmark problem [20][21]. All three methods and their variations are extensively used in control theory and there are numerous examples of their applications. However, they are out of the scope of the present review, since, due to computational reasons, they are limited to relatively small systems.

The time required to solve the Lyapunov equations as well as to perform a singular value decomposition grows as the cubic power in the number of equations, or is $O(n^3)$. Hence, if the system order increases twice, the time required to solve a new problem will increase about eight times. In other words, even though the results described above are valid all linear dynamic systems, practically we can use them for small order systems only.

The border between small and large systems depends on the computer power available and of course it steadily grows. According to Ref [20], the model reduction of a randomly generated linear system of order 512 takes 76 seconds on a 400 MHz Pentium II processor PC. Since processors now promise 1.2 GHz clock speed, this enables us to define current small systems as those with state vector dimensions in the range of 1000 to 2000.

## 4. Introduction to Krylov Subspaces

It happens that, in many cases, very good candidates for the required low-order subspace of Eq (6) are Krylov subspaces, and almost all modern model reduction methods for large-scale systems are based on them, one way or another. It should be noted that those iterative methods for solving a system of linear equations that are based on Krylov subspaces have been included in the list of the ten top algorithms of the 20th century [11].

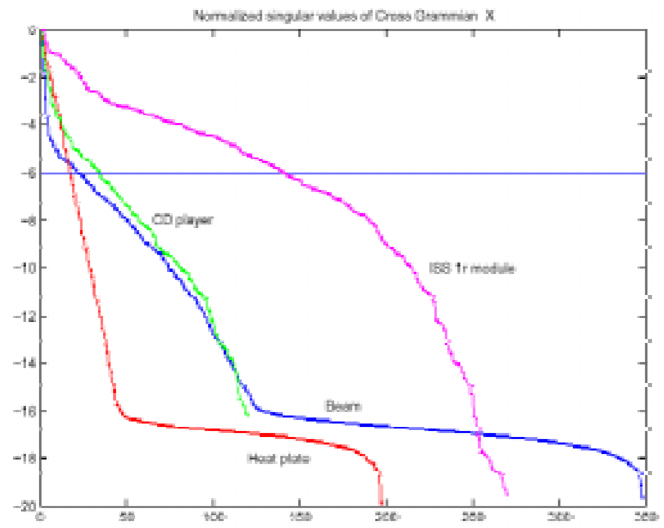

*Fig. 4 Decay of normalized Hankel singular values for four typical applications (from Ref. [7]). We expect these curves to also be typical for MEMS.*

A Krylov subspace of $k$-th dimension of the matrix $A \in \Re^n \times \Re^n$ and vector $\boldsymbol{v} \in \Re^n$ is defined as a subspace spanned by the original vector $\boldsymbol{v}$ and the vectors produced by consecutive multiplication of the matrix $A$ to this vector up to $k-1$ times, or

$$K_k^r(A, \boldsymbol{v}) = span\{\boldsymbol{v}, A \cdot \boldsymbol{v}, \ldots, A^{k-1} \cdot \boldsymbol{v}\} \tag{19}$$

The resulting vectors form a basis for $k$-dimensional subspace. However, if we compute them directly as written, then, because of rounding errors, they would become computationally linearly dependent even for relatively small $k$.

### 4.1 *Arnoldi and Lanczos algorithms to build the Krylov subspace*

A numerically stable procedure for building a Krylov subspace (19) is an Arnoldi process [11][22][23]. It generates an orthonormal basis $X \in \Re^n \times \Re^k$ for the Krylov subspace and a Hessenberg matrix, $H_A \in \Re^k \times \Re^k$, related to the original matrix as follows

$$X^* A X = H_A \tag{20}$$

The Hessenberg matrix for the Arnoldi process is made of an upper tridiagonal matrix plus one diagonal below the main diagonal. It can be considered as an orthogonal projection of the matrix $A$ onto the given Krylov subspace.

The main disadvantage of the Arnoldi method is that each new Arnoldi vector should be orthogonal to all previously generated vectors. This means that the computational cost grows disproportionately with the dimension of the subspace. The current alternative is to use a Lanczos algorithm, where the subspace (19) is considered as a right Krylov subspace. In addition to it, and in parallel, the left Krylov subspace

$$K_k^l(A^*, \boldsymbol{w}) = span\{\boldsymbol{w}, A^* \cdot \boldsymbol{w}, \ldots, (A^*)^{k-1} \cdot \boldsymbol{w}\} \tag{21}$$

is also generated, where the vector $\boldsymbol{w}$ can be equal or not to vector $\boldsymbol{v}$, depending on the applications, and $A^*$ is the conjugate transpose of the matrix $A$.

The Lanczos algorithm produces a pair of biorthogonal bases for subspaces (19) and (21) contained in the matrices $X$ and $Y$ such that

$$Y^* \cdot X = I \tag{22}$$

and a Hessenberg matrix $H_L$ that is in tridiagonal form. This means that, for any iteration of the algorithm, it is necessary to deal with just two previously generated vectors. The Lanczos Hessenberg matrix is related to the original matrix as

$$Y^* \cdot A \cdot X = H_L \tag{23}$$

and can be considered to be an oblique projection of $A$ onto the subspace (19) while remaining perpendicular to subspace (21). Fig. 5 illustrates the orthogonal and oblique projections of a vector. Because the Lanczos algorithm is based on three-term recurrences, it is faster for large $k$. However, it is computationally less stable than the Arnoldi process: a typical trade-off of accuracy *vs.* efficiency. The Lanczos and Arnoldi algorithms are mathematically equivalent if the matrix $A$ is symmetric and the starting vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are same, in other words, when the Krylov subspaces (19) and (21) are equivalent.

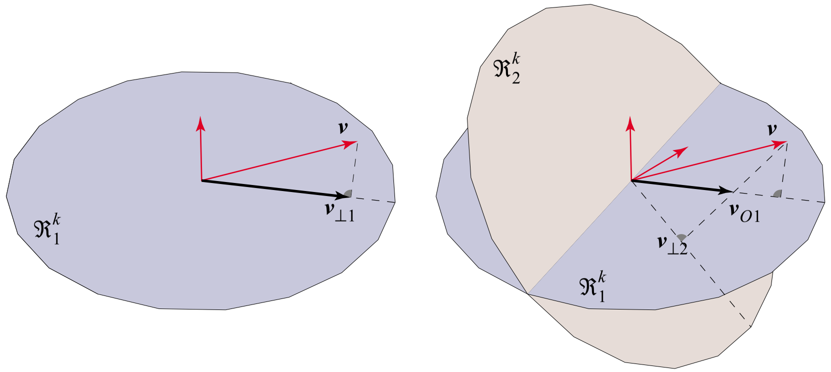

*Fig. 5 Example of an orthogonal and an oblique projection of a vector. The disks represent subspaces $\Re_i^k \subset \Re^n$ of the real linear space $\Re^n$ of a model reduction problem. A general vector $\boldsymbol{v} \in \Re^n$ is projected onto a subspace. The left figure illustrates an orthogonal projection $\boldsymbol{v}_{\perp 1} \in \Re_1^k$. The figure on the right demonstrates an oblique projection $\boldsymbol{v}_{O1} \in \Re_1^k$ determined by the "shadow" cast by an orthogonal projection of $\boldsymbol{v}$ onto a second subspace $\boldsymbol{v}_{\perp 2} \in \Re_2^k$.*

Instead of just one starting vector $\boldsymbol{v}$, one can take a number of starting vectors expressed by the matrix $B$. This leads to a generalization of the Arnoldi and Lanzcos algorithms to the so-called block-Arnoldi and block-Lanzcos algorithms [24][25]. Here we define the appropriate right and left Krylov subspaces as

$$K_k^r(A, B) = \mathrm{span}\{B, A \cdot B, \ldots, A^q \cdot B\} \tag{24}$$

$$K_k^l(A^*, C) = \mathrm{span}\{C, A^* \cdot C, \ldots, (A^*)^q \cdot C\} \tag{25}$$

One difficulty with the block-Krylov subspaces is that it is more difficult to predict the number of multiplications in Eq (24) and (25) in order to generate a $k$-dimensional subspace. Typically, $q$ is equal to the quotient of $k$ by the number of columns of the matrices $B$ or $C$, but the exact answer depends on the existence of linearly dependent vectors in (24) and (25).

### 4.2 Computing the inverse of the system matrix

One computational advantage of all Krylov subspace methods is in their iterative nature, *i.e.*, to perform them a user only needs to provide consecutive matrix by vector multiplications. This allows us to exploit the sparse form of the matrices, and to create fast application-specific implementations for the required $A \cdot \boldsymbol{v}$ product. The driver algorithms do not have to know the details how matrix $A$ is stored in the computer system.

For model reduction problems, the Krylov subspace (19) is actually based on the inverse of the system matrix $A$. Recalling Eq (5), this means that for both the Arnoldi and Lanzcos processes it is necessary to compute the product

$$F^{-1} \cdot E \cdot \boldsymbol{v} \tag{26}$$

We now discuss, using this example, the advantage of the iterative structure of the Krylov subspace methods.

$F$ and $E$ are large-dimensional sparse matrices, but the product $F^{-1} \cdot E$ might be a dense matrix, and the computational cost to compute this product is very high due to the presence of the matrix inverse. Hence its direct computation is unwise. It is much more efficient to compute the product $F^{-1} \cdot E \cdot \boldsymbol{v}$. First, before the procedure, one performs an LU-decomposition of $F$ (or equivalently, a Cholesky decomposition for a positive definite matrix, see [23]), which can take into account the sparse structure of $F$:

$$F = L \cdot U \tag{27}$$

where $L$ and $U$ are lower and upper triangular matrices, respectively. This is costly, but we will require $L$ and $U$ many times. Then, each multiplication $F^{-1} \cdot E \cdot \boldsymbol{v}$ is performed in three steps:

1) $E$ is multiplied by $\boldsymbol{v}$, $\boldsymbol{a} = E \cdot \boldsymbol{v}$. $E$ is sparse and so this is a potentially a fast operation.

2) a) The linear equations $L \cdot \boldsymbol{b} = \boldsymbol{a}$ are forward solved, so that $\boldsymbol{b} = L^{-1} \cdot E \cdot \boldsymbol{v}$. Since $L$ is lower triangular this is again a fast operation.
b) The backward solution of the linear equations $U \cdot \boldsymbol{c} = \boldsymbol{b}$ then gives us the desired product $U^{-1} \cdot L^{-1} \cdot E \cdot \boldsymbol{v}$. Again, since $U$ is upper triangular, this is a fast operation.

Once again, the above speedup is possible only because higher level algorithms do not need to have access to the full matrix $A^{-1}$; otherwise we would have no option but to compute it.

When the dimension of $A$ grows large enough LU-decomposition is no longer useful because it takes too much time. Hence, the second step above changes to

2) The linear equations $F \cdot \boldsymbol{b} = \boldsymbol{a}$ are solved by an iterative method, $\boldsymbol{b} = F^{-1} \cdot E \cdot \boldsymbol{v}$. If lucky, an iterative method can be reasonably fast for a particular $F$ matrix.

Iterative methods for the solution of a system of linear equations are also based on Krylov subspaces, and it is important not to confuse them with those reviewed in the present paper. The modified step above implies that, for any computation of the subsequent Krylov vector, it is necessary to use second level iterations to solve the linear system of equations. In addition to books [22] and [23], an excellent object oriented template-oriented review of Krylov-based methods for the solution of linear systems can be found in [26]. It should be noted that the success of iterative Krylov methods for a linear solve step depends on the structure of the matrix, and for the general case, their effective use requires finding a preconditioner, another matrix $P$, which transforms the original linear system to an equivalent $P \cdot F \cdot \boldsymbol{b} = P \cdot \boldsymbol{a}$, but which has superior convergence properties. For a discussion on the importance of preconditioning for solving linear systems that are generated through the discretization of PDEs by the finite element method, refer to Ref. [27].

## 5. Large Linear Systems

As was already mentioned, algorithm time complexity limitations do not allow us to directly employ control theory algorithms for large-scale systems. As a result, most of the practical work in model reduction of large linear dynamic systems have been tied to Padé approximants of the transfer function (14), and we start the present section with them. These methods are computationally feasible but, on the other hand, they do not provide a global error estimate. Recently, there have been considerable efforts to find computationally effective strategies in order to apply methods based on Hankel singular values to large-scale systems, and we briefly review them in the second part of this section.

### 5.1 *Approximating a Transfer Function by Padé and Padé-type Approximants*

For the case of Single-Input-Single-Output (SISO) systems, when matrices $A$ and $C$ both are composed of a single column and row accordingly, the transfer matrix (14) is a scalar rational function which can always be expressed in the factored form as

$$G(s) = \frac{a(s-z_1)\dots(s-z_{n-1})}{(s-p_1)\dots(s-p_n)} \tag{28}$$

where $z_i$ and $p_i$ are zeros and poles of the transfer function and $a$ is a constant. In the Multiple-Input-Multiple-Output (MIMO) case a transfer matrix is of dimension $p$ by $m$, each element of which being a function of the form of Eq (28).

The idea of Padé [28] and Padé-type [29] approximants is to find a rational function of smaller dimension $k$, $\ddot{G}(s)$, which retains the essential behavior of the large-dimensional original rational function. This is formulated in terms of moment matching in the expansion of the transfer functions around some given number $s_o$ (in most applications $s_o = 0$),

$$G(s) = \sum_{i=0}^{\infty} m_i(s-s_o)^i \tag{29}$$

that is,

$$m_i = \hat{m}_i \text{ for } i = 0, \dots, q \tag{30}$$

Padé approximants match the maximum number of moments, $q = 2k$, while Padé-type approximants match first $q < 2k$ moments. This is easily generalized to the multiple input - multiple output (MIMO) system, where all moments will be $p \times m$ matrices.

It happens that the Arnoldi process for the right Krylov subspace

$$K_k^r\{(A-s_oI)^{-1}, (A-s_oI)^{-1} \cdot B\} \tag{31}$$

produces such matrices $H_A$ and $X$ such that the reduced system

$$\hat{A} = H_A^{-1} \cdot (I + s_oH_A)\ ,\ \hat{B} = H_A^{-1} \cdot X^* \cdot (A-s_oI)^{-1} \cdot B\ ,\ \ddot{C} = C \cdot X \tag{32}$$

implicitly matches the first $k$ moments in Eq (29), that is, the Arnoldi process implicitly produces a Padé-type approximant of the original transfer function (14). On the other hand, if one performs Lanczos algorithms for the right (31) and left Krylov subspaces

$$K_k^l\left\{[(A-s_oI)^{-1}]^*, [(A-s_oI)^{-1}]^* \cdot C^*\right\} \tag{33}$$

matrices $H_L$, $X$ and $Y$ produce the reduce system

$$\hat{A} = H_L^{-1} \cdot (I + s_oH_L)\ ,\ \hat{B} = H_L^{-1} \cdot Y^* \cdot (A-s_oI)^{-1} \cdot B\ ,\ \ddot{C} = C \cdot X \tag{34}$$

which will match $2k$ moments of the original dynamic system [24][25][30][31]. Note that Eqs (32) and (34) greatly simplify for the case of an expansion about $s_o = 0$, and that there are computationally more effective formulas for producing the reduced matrices $\hat{B}$ and $\ddot{C}$.

The Lanczos algorithm produces a reduced system closer to the original one, because the number of moments matched here is twice that of the Arnoldi process. This has a simple explanation. Model reduction by the Arnoldi process does not take into account matrix $C$ at all, while model reduction by means of the Lanczos algorithm is made by an oblique projection on the right Krylov subspace (31) that takes into account the left Krylov subspace (33).

Still, both approaches are based on moment matching and they are by nature local, in the sense that, in general, they make a good approximation of the transfer function (14) near the expansion point $s_o$ only. This can be improved by multi-point expansion, i.e., expanding the transfer function (14) about several points $s_i$ and requiring the reduced transfer function to match the first moments at all expansion points. This idea was implemented in the so-called Rational Krylov method [31][32], where the Arnoldi or Lanczos algorithms were applied to the union of the Krylov subspaces (31) and (33) for different values of $s_i$. The main methodological challenge here is to decide how to choose the expansion points, and to determine how many are needed. Computationally this adds an additional load. If one uses LU-decomposition for the inverse of the system matrices (see Section 4.2), then in this case it is necessary to perform an LU-decomposition for each value of $s_i$.

The original dynamic system can be stable, that is, when time goes to infinity the values of $\boldsymbol{x}$ remain finite (bounded), and passive, which is to say, the system does not generate energy. If so, then it is important, especially in electrical circuit simulations, that the reduced system also possesses these properties. Unfortunately, both the “out-of-the-box” Arnoldi and Lanczos algorithms do not guarantee this, and special attention should be paid to preserve the properties of the original dynamical system. It happens that the Arnoldi process is mathematically more simple than Lanczos algorithm (this is stressed by their names, process and algorithms, respectively). Probably for this reason, engineers often choose the Arnoldi process: the coordinate transformed Arnoldi [33] for stable model reduction, and the provably passive model reduction method “block Arnoldi plus congruent transform” or (PRIMA) of Ref. [34]. On the other hand, mathematicians still bet on the Lanczos algorithms [24][25][35][36], because, as was mentioned above, it takes into account the observability matrix $C$ and it matches twice the number of moments of the Arnoldi process. They seem to prefer, while preserving the properties of the original dynamic system, to match as many moments as possible so as to obtain the most accurate representation for the same dimension $r$ of the reduced model. It is also worth noting that, even though when some algorithm provably produces a passive reduced model, this does not mean that its computer implementation will really produce a passive model in practice, mainly because of the inevitable numerical rounding errors [24].

Now let us return to the original case of model reduction for systems (4) to (8). From a control theory viewpoint we term it Single-Input-Complete-Output or SICO. It so happens that if system (4) is obtained during the discretization of a diffusion-convection partial differential equation, then the Krylov subspace (31) with $s_o = 0$ is a very good choice for the lower dimensional subspace in Eq (6) [8][9][37]-[43]. In this case, the model reduction step (8) can be viewed as an approximate solution of the original system (4), because it is possible to recover the solution for all of the original unknowns by means of Eq (6). This work has been superseded by the use of a Krylov subspace (31) to approximate the matrix exponential [10][23], but mathematically this is identical with a Padé-type approximant (32) when the matrix $C$ is just discarded.

We next list examples of the papers in which Padé and Padé-type approximants via Krylov subspace methods have been used for the model reduction of a linear system (9). The papers come from several distinct communities:

- The largest community comes from electrical engineering where model reduction is mostly employed to deal with the so-called microchip interconnect problem [44][45][46]: Mixed surface volume for 3D interconnect [47], Lossy multiconductor transmission lines [48], 3D interconnect and packaging based on an alternate Partial Element Equivalent Circuit (PEEC) formulation [49], Coupled lossy transmission lines [50], Magetoquasistatic analysis for packaging parasitics with skin effect [51], PEEC model of an electromagnetic problem [52], Electromagnetic devices modeled by linearized Maxwell equations [53], Full-wave electromagnetic analyses [54], and Electromagnetic wave propagation by the finite element method (FEM) [55]. The actual number of publications on model reduction here is much greater.
- The ideas from electrical engineers have been used for the model reduction of wave-propagation-like problems: The Helmholtz equation for exterior structural acoustics by FEM [56], Neutron noise for nuclear reactor by the finite difference method [57], Aeroelastic analyses of turbomachines [58].
- Another community solves the advection-diffusion PDE, which arises in a variety of engineering disciplines. They mostly deal with the single-input-complete-output (SICO) case discussed above. Here, model reduction is at the beginning stage if we compare the number of papers in which model reduction is used to the total number of papers on the solution of advection-diffusion PDEs: Advection dispersion equation for groundwater flow [37][38], Mass transport in hydrogeologic environments [39], Photon diffusion (optical tomography) problem [40], Radionuclide decay-chain transport in porous media [41], Groundwater flow in dual-porosity media [42], Radionuclide decay chain transport in dual-porosity media [43], Groundwater flow in discretely fractured porous media [8], and Diffusion and convection dominated flow [9].
- Finally, we have the MEMS community which has just recently started to exploit the modern opportunities of model reduction: Electrostatic gap-closing actuator [59], Linearized model for a micromirror [60], and the Comb-drive resonator [61]. It is interesting to note that the MEMS community appears to have learned about model reduction from the electrical engineers and is not aware of work on the model reduction of advection-diffusion PDEs, even though this body of work is much closer to typical MEMS simulations.

### 5.2 *Approximating Lyapunov Equations*

Unfortunately, Padé and Padé-type approximants do not have global error estimates, similar to Eq (18), and this drives mathematicians to develop computationally effective strategies for large dimensional systems based on the methods described in Section 3. In [7] these approaches are referred to as SVD-Krylov, and in [62] there is a good overview of existing strategies.

The optimal minimal reduction methods for linear systems comprise two computationally expensive steps: solution of Lyapunov equations (15) and (16) for the controllability and observability grammians, and then eigenvalue-type decomposition of the product of two grammians, Eq (17). The computational time for both steps, even using the most advanced computational methods [63], grows as the cube of the system dimension.

A general idea to decrease the computational time is to change the exact grammians to their low-rank approximations. It happens that it is possible if the number of inputs and outputs are much less than the dimension of the state vector, $m \ll n$ and $p \ll n$, and this is the case for the most important practical applications. As a result, it is possible to solve Lyapunov equations for low-rank grammian approximations much faster than for exact grammians [64][65][66][67]. For the case of a dense matrix $A$, the computational time is already proportional to the square of the system dimension $n$, and it may be linearly proportional to $n$ for the case of a sparse matrix $A$. Also, the advantage of these methods is that they can be formulated in terms of matrix-vector products only, as for the Krylov subspace methods. The second step, balancing, with the use of low-rank grammians, is also much faster because there are special algorithms that can take this into account [62][67].

A very simple case of model reduction arises when the inputs are the same as the outputs, and matrix $A$ is symmetric. Note, that if matrices $E$ and $F$ in Eq (1) are symmetric and $E$ is positive definite, then by an appropriate coordinate transformation one can obtain Eq (4) with a symmetric matrix $A$ [33]. In this case, the grammians are equal to each other because Eqs (15) and (16) become the same: then it is necessary to solve just a single Lyapunov equation and there is no need to perform balancing. Another approach is to use, instead of two Lyapunov equations (15) and (16), the Sylvester equation [68]

$$A \cdot R + R \cdot A = -B \cdot C \tag{35}$$

to find the so-called cross-grammian $R$. It happens that in the case of a linear dynamic system with a symmetric transfer function, the Hankel singular values are equal to the eigenvalues of the cross-grammian, and here there is also no need for balancing. This is always true for any SISO system, because in this case the transfer function is a scalar. In the MIMO case, one can use a transformation described in Ref [68] in order to convert any linear dynamic system to one with a symmetric transfer function.

Some methods for model reduction based on solving large dimensional Lyapunov equations are implemented in the library LYPACK [69] (it requires MATLAB). As mentioned in Ref [69], Lyapunov equations of order more than 12000 were solved by LYPACK within a few hours on a regular workstation.

## 6. Nonlinear Systems

Now let us allow the elements of the system matrices to depend on the state vector $\boldsymbol{x}$ and on the time. If they depend explicitly on time only, then we have a special case of a time-varying system, and there are examples of extending Krylov subspace model reduction methods to this case [54][70].

Note that, even when system matrices depend on $\boldsymbol{x}$, Eq (4) is a special case of a general non-linear system

$$\frac{d\boldsymbol{x}}{dt} = f[\boldsymbol{x}(t), \boldsymbol{u}(t)] \tag{36}$$

An evident solution for model reduction is to split the whole system into nonlinear and linear parts and then to apply the model reduction to the linear subparts [24], thus reducing the total number of unknowns in the state vector. Another popular alternative is to linearize the non-linear system around an operating point and then to make model reduction for the resulting linear system. Definitely, the answer as to whether this is possible depends on the application in question. There is an interesting example in Refs [71][72], where, in order to improve the precision of the linearization process, the authors have included quadratic terms in the expansion.

There are some methods for model reduction of nonlinear systems applicable to small-dimensional problems [73][74], and some special cases where it is possible to find particular approaches which allow us to use ideas from the previous section [59][75][76]. Nevertheless, to our knowledge, for the general case of model reduction of large nonlinear systems, there appears to be one approach only, which we consider in the next section.

### 6.1 *Proper Orthogonal Decomposition*

For systems with strong nonlinear effects, linearization is impossible because a linearized system cannot capture the complexity of the original phenomena. We remind ourselves that nonlinear systems may show instabilities such as snap-through, and bifurcations, and ultimately the onset of chaotic behaviour, all of which should be represented in the reduced system. In this case, in order to find an appropriate low-dimensional subspace (6), one can use results of the full order simulation of the original dynamic system (4), and this is implemented within the proper orthogonal decomposition (POD; another popular name is Karhunen-

Loève decomposition) [77]. This is the main difference w. r. t. linear systems, where the model reduction process can be based on the system matrices without performing a full order simulation.

Let us consider a slightly simplified procedure for a finite-dimensional system. The first step is to perform one or more simulations and to collect a series of so-called “snapshots”

$$W = \{\boldsymbol{x}_i\}\,,\ W \in \Re^n \times \Re^s \tag{37}$$

where matrix $W$ is composed from $s$ state vectors $\boldsymbol{x}_i$, corresponding to different times of simulations of Eq (4). This is the most crucial step during POD because the reduced basis will be obtained from matrix $W$ only, and if it does not give a good representation of the whole ensemble of possible values of $\boldsymbol{x}$, then the generated low-dimensional basis will lead to a poor quality of approximation. If, for linear systems, it was possible to perform model reduction for any input functions, for non-linear systems it is necessary to choose the most typical input functions, and to perform simulations with them. Unfortunately, there exist almost no formal rules as to how to choose the number “snapshots” to collect and at what times they should be taken. Hence, POD is more of an “art”, and typically, for any new nonlinear system, it is necessary to make a special investigation in this respect.

Nevertheless, the following POD steps are completely formal. For a given “snapshot” matrix $W$ it is formally possible to find a low rank approximation within a given error margin by means of a Singular Value Decomposition (SVD) [7][23]

$$W = U \cdot \Sigma \cdot V^T = \sum_{i=1}^{s} \sigma_i(\boldsymbol{u}_i \cdot \boldsymbol{v}_i^T)\,,\ \Sigma \in \Re^n \times \Re^s\,,\ U \in \Re^n \times \Re^n\,,\ S \in \Re^s \times \Re^s \tag{38}$$

where $\Sigma = diag\{\sigma_i\}$ is a diagonal matrix of singular values, $U = \{\boldsymbol{u}_i\}$ is a matrix of left singular vectors, and $V = \{\boldsymbol{v}_i\}$ is a matrix of right singular vectors. Provided the singular values of $W$ rapidly decay we can take only a small number singular vectors, $k \ll s$, corresponding to the largest singular values, and this gives us a low-rank approximation of matrix $W$ of the form

$$\hat{W} = \hat{U} \cdot \hat{\Sigma} \cdot \hat{V}^T = \sum_{i=1}^{k} \sigma_i(\boldsymbol{u}_i \cdot \boldsymbol{v}_i^T) \tag{39}$$

where the reduced matrices are formed from the full matrix by leaving only $k$ dominant vectors. Eq (39) shows that all observations are effectively described by a small number of vectors $\boldsymbol{u}_i$, which gives a reduced basis on which to project the original differential equation:

$$X = \ddot{U} \tag{40}$$

The transition from Eq (37) to (39) can be made completely automatic because according to SVD-theory there is an error estimate based on singular values with the norm

$$\|W - \ddot{W}\| \tag{41}$$

and Eq (39) actually reduces this norm to a minimum. The problem is that it is difficult to predict, a priori, whether this error estimate can be used for the transition from Eq (4) to (8), because this already strongly depends on the quality of the generated “snapshots”, that is, whether they are representative or not.

The final step is to project original non-linear equation onto the low-dimensional basis. For Eq (4), when the elements of $A$ and $\boldsymbol{b}$ depend on $\boldsymbol{x}$, we can write

$$\hat{A} = X^* \cdot A \cdot X \text{ and } \hat{\boldsymbol{b}} = X^* \cdot \boldsymbol{b} \tag{42}$$

and, for the general case of Eq (36), the reduced model becomes

$$\frac{dz}{dt} = X^* \cdot \boldsymbol{f}[X \cdot \boldsymbol{z}(t), \boldsymbol{u}(t)] \tag{43}$$

There is a hidden computational problem with Eq (42), momentarily ignoring Eq (43), which is, how to compute the reduced system matrices. Matrix $A$ contains some functions of $\boldsymbol{x}$ and hence Eq (42) should be computed by means of symbolic manipulations. This is practically unfeasible. In the general case one may only compute the right sides in (42) for each time step during the simulation of the reduced model and this then constitutes the main computational cost. For example, in Ref. [78], the size of the state vector has been reduced from 21540 to 15 (about 1500 times) but, because of the above reason, the time of the simulation was reduced only by a factor of six.

POD has been used extensively in fluid dynamics in order to model turbulence [77]. Recently, it has been employed in a variety of disciplines tied with nonlinear dynamics: Rapid thermal processing systems [79], Control of a solid fuel ignition [80], Chemical vapor deposition [81][82] and other Distributed reacting systems [83]-[86], Cascading failures in power systems [87], Feedback control of systems governed by a nonlinear PDE [88]-[90], and various Mechanical engineering problems [78],[91]-[93]. The MEMS community has also started to employ this technique [71][94][95].

The SVD decomposition of a matrix is a computationally demanding method: the time grows as the cube of the matrix dimension. This means that when the dimension of matrix $W$ grows we might not have enough computational resources in order to make the decomposition (38). It happens that again iterative methods based on the Krylov subspaces can help to find the dominant singular vectors without performing the full SVD decomposition [96][97] and thus keeping computational time within reasonable limits.

The original POD procedure does not take into account the information about required system inputs and outputs, and this limits its applicability in system simulation. Recently, the method has been generalized [98][99] in order to take into account ideas from the linear control theory. The generalization is based on the introduction of “empirical grammians” which are computed based on “empirical snapshots”. This opens new perspectives for applications of POD to nonlinear model reduction and hopefully in the future we will see further development of these ideas.

## 7. Conclusion

Let us summarize the current status of automatic model order reduction. The situation is reasonably good for large-scale linear dynamic time-invariant systems. The moment matching methods for model reduction based on the Arnoldi and Lanczos algorithms are in a mature state. They scale well with the size of the system, their behavior is fairly predictable, and they are easily implemented in almost any computational environment. As was already mentioned, the Arnoldi process is more computationally stable and one can implement it much easier than the Lanczos algorithm. On the other hand, the latter can match more moments and thus provides a better approximation of the original system. As a result, the Arnoldi process is the best choice for those who would like to implement moment matching methods fast and from scratch, and it is better to obtain the implementation of the Lanczos algorithm from professional sources.

A typical question with moment matching techniques is: when to stop model reduction. A good strategy is provided in Ref. [52], where a local error estimate has been suggested for model reduction based on the Lanczos algorithm. First, it is necessary to estimate a range of frequencies in which the approximation of the transfer function is required. It is possible to set $s_o$ to an expansion point in the middle of this range and then to use the local error estimate on the border of this range as a monitor as to when to stop the model reduction process, because the approximation error typically increases faster the further on is from the expansion point. This procedure still does not give a global error estimate as the balanced truncation approximation does, but for most engineering purposes this should be good enough.

Another problem is that Padé and Padé-type approximants are local by their nature, and they might be not optimal if one would like to obtain a good approximation of the transfer function over a wide range of $s$, that is, the dimension of the reduced model might then be too large. Here one can think of a Rational Krylov approximation or to employ a two-step strategy suggested in Ref. [51]. First one computes a medium order model by means of moment matching techniques around a chosen $s_o$ (for many cases outside of electrical engineering $s_o = 0$ seems to be a satisfying choice) and then to employ a truncated balanced approximation to reduce the intermediate model as much as possible.

The development of model reduction based on the solution of Lyapunov equations is the next logical step for a large linear system. It is quite evident that in the few next years we will see more practical examples in this area, and as the experience of mathematicians grows one can expect more practical outcomes for engineers. This will bring us truly automatic model reduction, just as we have for the case of small linear systems right now, provided the minimum over the norm (18) is enough for the application. Let us stress this with the example from Ref. [100]. The norm (18) measures the absolute error over the whole frequency range, and if the transfer function changes by many orders of magnitude, then the balanced truncation approximation could describe the transfer function behavior quite well if we consider it from the viewpoint of the absolute error, but not that well if we consider the relative error.

The situation with nonlinear systems is quite different, and human intervention in some form appears to be inevitable here. First, it is necessary to see if a problem in question can be handled by

- linearization,
- splitting to linear and non-linear subparts,
- some special effective case for a particular nonlinear dynamic system.

If not then the choice is clearly POD, where the main questions are: how many “snapshots” should be generated, and how often. Alternatives here are to follow the example of a similar nonlinear system, or to make a special investigation in order to learn the special behaviour and requirements of the system. Nevertheless, the POD suggests quite an appropriate framework for general nonlinear model reduction because it is possible to state that human intervention here is limited to decision making. After a researcher has decided on how to obtain matrix (37), all other POD steps can be made fully automatic. POD is especially attractive for those applications where it is possible to obtain a reduced system matrix (42) in a closed form, that is, when the governing equations can be directly projected to the reduced basis.

## 8. On-line resources

The advent of the internet has made accessible a wide variety of informational resources. There are good slide shows on model reduction with illustrations and examples [101][102]. Below there are homepages of scientists involved in model reduction, in which one can find additional resources:

- A. C. Antoulas - http://www-ece.rice.edu/~aca/
- P. Benner - http://www.math.uni-bremen.de/~benner/
- D. Boley - http://www-users.cs.umn.edu/~boley/
- R. W. Freund - http://cm.bell-labs.com/who/freund/
- B. B. King - http://www.math.vt.edu/people/bbking/
- J. E. Marsden - http://www.cds.caltech.edu/~marsden/
- S. Lall - http://element.stanford.edu/~lall/
- T. Penzl - http://www.mathematik.tu-chemnitz.de/in_memoriam/penzl/
- Y. Saad - http://www-users.cs.umn.edu/~saad/
- P. Van Dooren - http://www.auto.ucl.ac.be/~vdooren/
- A. Varga - http://www-er.df.op.dlr.de/staff/varga/

Also, there are a number of theses, available on the internet, which provide a good introduction in a particular field: Adaptive meshing [103], Krylov subspaces [104], Control theory [105], Moment matching model reduction [106][107], SVD-Krylov model reduction [108], and POD [109][110].

## 9. Acknowledgments

The research described in this publication was partly made possible by Grant IST - 1999 - 29047 from the European Commission Community Research, Information Society Technologies Program, and by Grant partly by Grant DFG 1883/3-1 from the Deutsche Forschungsgemeinschaft, and partly by an operating grant from the University of Freiburg. Part of the manuscript was completed while Korvink was on sabbatical at Ritsumeikan University in Kusatsu, Japan. Authors wish thank Tamara Bechtold for invaluable help during the writing of the review, and to Prof. Dr. Boris Lohmann for critically reading the manuscript.

## 10. References

[1] T. Mukherjee, G. K. Fedder, D. Ramaswany, J. White, Emerging simulation approaches for micromachined devices, *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst.* 2000, 19, 1572-1589.

[2] *The NCAC Animation Archives of Impact Simulations*, http://www.ncac.gwu.edu/archives/animation/ .

[3] *LS-DYNA performance for car-to-car and car-to-barrier benchmarks on IBM SP and Compaq AlphaServer SC*, September, 2000, http://www.epm.ornl.gov/evaluation/CRASH/index.html

[4] M. Emmenegger, J. G. Korvink, M. Bächtold, M. von Arx, O. Paul and H. Baltes., Application of Harmonic Finite Element Analysis to a CMOS Heat Capacity Measurement Structure, *Sensors and Materials*,1998, 10, 405–412

[5] N. Bouhaddi, R. Fillod, Substructuring by a two level dynamic condensation method, *Computers and Structures*, 1996, 60, 403-409.

[6] J. Guyan, Reduction of stiffness and mass matrices, *AIAA J.* 1965, 3, 380.

[7] A. C. Antoulas, D. C. Sorensen, *Approximation of large-scale dynamical systems: An overview*, Technical Report, Rice University, Houston, 2001, http://www-ece.rice.edu/~aca/mtns00.pdf.

[8] A. Woodbury, K. N. Zhang, Lanczos method for the solution of groundwater flow in discretely fractured porous media, *Adv. Water Resour.* 2001, 24, 621-630.

[9] G. Pini, G. Gambolati, Arnoldi and Crank-Nicholson methods for integration in time of the transport equation, *Int. J. Numer. Methods Fluids*, 2001, 35, 25-38.

[10] M. Hochbruck, C. Lubich, H. Selhofer, Exponential integrators for large systems of differential equations, *SIAM J. Sci. Comput.* 1998, 19, 1552-1574.

[11] H. A. van der Vorst, Krylov subspace iteration, *Comput. Sci. Eng.* 2000, 2, 32-37.

[12] E. Hairer, S. P. Norsett, G. Wanner, *Solving ordinary differential equation, I, Nonstiff problem*, Springer-Verlag, 1986.

[13] Antoulas, Frequency domain representation and singular value decomposition, A.C. Antoulas, *UNESCO EOLSS (Encyclopedia for the Life Sciences)*, Contribution 6.43.13.4, June 2001.

[14] O. C. Zienkiewicz, R. L. Taylor, *The Finite Element Method*, McGraw-Hill, 1989.

[15] J. Müller, J. G. Korvink, Automatic Mesh Adaptivity for Finite Element Simulation of Multi-Layer MEMS, in *Materials and Device Characterization in Micromachining III*, Y. Vladimirski, P.J. Coane, eds, Proc. SPIE 4175, Santa Clara, 2000, 82-93.

[16] J. W. Polderman, J. C. Willems, *Introduction to mathematical systems and control: A behavioral approach*, Text in Applied Mathematics, 26, Springer Verlag, 1998.

[17] Antoulas, Approximation of linear dynamical systems, *Wiley Encyclopedia of Electrical and Electronics Engineering*, (Ed.: J.G. Webster), 1999, 11, 403-422.

[18] T. Penzl, Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case, *Syst. Control Lett.* 2000, 40, 139-144.

[19] A. C. Antoulas, D. C. Sorensen, Y. Zhou, *On the decay rate of the Hankel singular values and related issues*, Technical Report, Rice University, Houston, 2001, http://www-ece.rice.edu/~aca/decay_fin.ps.

[20] A. Varga, Model reduction software in the SLICOT library. In *Applied and Computational Control, Signals, and Circuits*, (Ed.: B.N. Datta), Kluwer Academic Publishers, Boston, 2001, 2, 239-282.

[21] A. Varga, Model reduction routines for SLICOT, *Niconet Report 1999-8*, 1999, ftp://wgs.esat.kuleuven.ac.be/pub/WGS/REPORTS/NIC1999-8.ps.Z.

[22] Y. Saad, *Iterative methods for sparse linear systems*, The PWS series in computer science, Boston, PWS Pub. Co, 1996, p. 447.

[23] G. H. Golub, C. F. Van Loan, *Matrix computations*, 3rd ed., Johns Hopkins studies in the mathematical sciences, Johns Hopkins University Press, 1996, p. 694.

[24] R. W. Freund, Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation, in *Applied and Computational Control Signals and Circuits*, (Ed: B.N. Datta), Birkhauser, Boston, 1999, 435-498.

[25] R. W. Freund, Krylov-subspace methods for reduced-order modeling in circuit simulation, *J. Comput. Appl. Math*. 2000, 123, 395-421.

[26] R. Barrett, M. Berry, Y. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Poso, C. Romine, H. van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*, Society for Industrial and Applied Mathematics, Philadelphia, 1994, p. 112. See also http://www.netlib.org/templates/templates.ps.

[27] R. M. Ferencz, T. J. R. Hughes, Iterative finite element solutions in nonlinear solid mechanics, in *Handbook of numerical analysis* (Ed: P. G. Ciarlet, J. L. Lions), Elsevier Science, 1998, 6, 3-174.

[28] G. A. Baker, P. R. Graves-Morris, *Padé approximant*,. 2nd ed., Encyclopedia of mathematics and its applications, Cambridge University Press, 1996, 59, p. 746.

[29] C. Brezinski, *Padé-type approximation and general orthogonal polynomials*, International series of numerical mathematics, Birkhäuser, 1980, v. 50, p. 250.

[30] E. J. Grimme, D. C. Sorensen, P. Van Dooren, Model reduction of state space systems via an implicitly restarted Lanczos method, *Numerical Algorithms*, 1996, 12, 1-31.

[31] K. Gallivan, E. Grimme, P. Van Dooren, A rational Lanczos algorithm for model reduction, *Numerical Algorithms*, 1996, 12, 33-63.

[32] A. Ruhe, D. Skoogh, Rational Krylov algorithms for eigenvalue computation and model reduction, in *Applied Parallel Computing, Lecture Notes in Computer Science*, 1998, 1541, 491-502.

[33] L. M. Silveira, M. Kamon, I. Elfadel, J. White, A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits, *Comput. Meth. Appl. Mech. Eng*. 1999, 169, 377-389.

[34] A. Odabasioglu, M. Celik, L. T. Pileggi, PRIMA: Passive reduced-order interconnect macromodeling algorithm, *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst*. 1998, 17, 645-654.

[35] Z. J. Bai, R. W. Freund, A partial Padé-via-Lanczos method for reduced-order modeling, *Linear Alg. Appl*. 2001, 332, 139-164.

[36] K. Gallivan, E. Grimme, D. Sorensen, P. Van Dooren, On some modifications of the Lanczos algorithm and the relation with padé approximations, *Mathematical Research Series*, 1996, 7, 87-116.

[37] H. Li, A. Woodbury, P. Aitchison, Eigenvalue translation method for stabilizing an Unsymmetric Lanczos Reduction process, *Int. J. Numer. Methods Eng*. 1998, 43, 221-239.

[38] H. N. Li, P. Aitchison, A. Woodbury, Methods for overcoming breakdown problems in the Unsymmetric Lanczos Reduction method, *Int. J. Numer. Methods Eng*. 1998, 42, 389-408.

[39] D. A. Farrell, A. D. Woodbury, E. A. Sudicky, Numerical modeling of mass transport in hydrogeologic environments: performance comparison of the Laplace Transform Galerkin and Arnoldi modal reduction schemes, *Adv. Water Resour*. 1998, 21, 217-235.

[40] Q. Su, V. L. Syrmos, D. Y. Y. Yun, A numerical algorithm for the diffusion equation using 3D FEM and the Arnoldi method, *Circuits Syst. Signal Process*. 1999, 18, 291-314.

[41] H. N. Li, A. Woodbury, P. Aitchison, Application of the unsymmetric Lanczos method to radionuclide decay-chain transport in porous media, *Int. J. Numer. Methods Eng*. 1999, 44, 355-372.

[42] K. Zhang, A. D. Woodbury, W. S. Dunbar, Application of the Lanczos algorithm to the simulation of groundwater flow in dual-porosity media, *Adv. Water Resour*. 2000, 23, 579-589.

[43] K. Zhang, A. D. Woodbury, The Arnoldi reduction technique for efficient direct solution of radionuclide decay chain transport in dual-porosity media, *J. Contam. Hydrol*. 2000, 44, 387-416.

[44] C.-K. Cheng, J. Lillis, S. Lin, N. Chang, *Interconnect Analysis and Synthesis*, John Wiley & Sons, Inc. 2000.

[45] R. Achar, M. S. Nakhla, Simulation of high-speed interconnects, *Proc. IEEE*, 2001, 89, 693-728.

[46] A. E. Ruehli, A. C. Cangellaris, Progress in the methodologies for the electrical modeling of interconnects and electronic packages, *Proc. IEEE*, 2001, 89, 740-771.

[47] M. Chou, J. K. White, Efficient formulation and model-order reduction for the transient simulation of three-dimensional VLSI interconnect, *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst*. 1997, 16, 1454-1476.

[48] M. Celik, A. C. Cangellaris, Simulation of multiconductor transmission lines using Krylov subspace order-reduction techniques, *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst*. 1997, 16, 485-496.

[49] M. Kamon, A. Marques, L. M. Silveira, J. White, Automatic generation of accurate circuit models of 3-D interconnect, *IEEE Trans. Compon. Packag. Manuf. Technol. Part B-Adv. Packag*. 1998, 21, 225-240.

[50] L. Knockaert, D. De Zutter, Passive reduced order multiport modeling: The Pade-Laguerre, Krylov-Arnoldi-SVD connection, *AEU-Int. J. Electron. Commun*. 1999, 53, 254-260.

[51] M. Kamon, F. Wang, J. White, Generating nearly optimally compact models from Krylov-subspace based reduced-order models, *IEEE Trans. Circuits Syst. II-Analog Digit. Signal Process*. 2000, 47, 239-248.

[52] Z. J. Bai, R. D. Slone, W. T. Smith, Error bound for reduced system model by Pade approximation via the Lanczos process, *IEEE Trans. Comput-Aided Des. Integr. Circuits Syst*. 1999, 18, 133-141.

[53] J. E. Bracken, D. K. Sun, Z. Cendes, Characterization of electromagnetic devices via reduced-order models, *Comput. Meth. Appl. Mech. Eng*. 1999, 169, 311-330.

[54] I. Balk, On a passivity of the Arnoldi based model order reduction for full-wave electromagnetic modeling, *IEEE Trans. Adv. Packag*. 2001, 24, 304-308.

[55] R. D. Slone, R. Lee, Applying Pade via Lanczos to the finite element method for electromagnetic radiation problems, *Radio Sci*., 2000, 35, 331-340.

[56] M. Malhotra, P. M. Pinsky, Efficient computation of multi-frequency far-field solutions of the Helmholtz equation using Pade approximation, *J. Comput. Acoust*., 2000, 8, 223-240.

[57] Z. F. Kuang, I. Pazsit, A. Ruhe, Calculating reactor transfer functions by Pade approximation via Lanczos algorithm, *Ann. Nucl. Energy*, 2001, 28, 1595-1611.

[58] K. Willcox, J. Peraire, J. White, An Arnoldi approach for generation of reduced-order models for turbomachinery, *Comput. Fluids*, 2001, 31, 369-389.

[59] Z. Bai, D. Bindel, J. Clark, J. Demmel, K. S. J. Pister, N. Zhou, New Numerical Techniques and Tools in SUGAR for 3D MEMS Simulation, in *Technical Procedings of the Forth International Conference on Modeling and Simulation of Microsystems*, 2001.

[60] D. Ramaswamy, J. White, Automatic Generation of Small-Signal Dynamic Macromodels from 3-D Simulation, in *Technical Procedings of the Forth International Conference on Modeling and Simulation of Microsystems*, 2001.

[61] V. Srinivasan, A. Jog, R. B. Fair, Scalable Macromodels for Microelectromechanical System, in *Technical Procedings of the Forth International Conference on Modeling and Simulation of Microsystems*, 2001.

[62] T. Penzl, Algorithms for Model Reduction of Large Dynamical Systems, *Preprint SFB393/99-40*, 1999, http://www.mathematik.tu-chemnitz.de/preprint/quellen/1999/SFB393_40.ps.gz.

[63] T. Penzl, Numerical solution of generalized Lyapunov equations, *Adv. Comput. Math*. 1998, 8, 33-48.

[64] A. Hodel, K. Poolla, B. Tenison, Numerical solution of the Lyapunov equation by approximate power iteration, *Linear Algebra Appl*. 1996, 236, 205-230.

[65] T. Penzl, A cyclic low-rank smith method for large sparse Lyapunov equations, *SIAM J. Sci. Comput*. 2000, 21, 1401-1418.

[66] A. C. Antoulas, D. C. Sorensen, S. Gugercin, *A modified low-rank Smith method for large-scale Lyapunov equations*, Technical report, Rice University, Houston, 2001, http://www.caam.rice.edu/caam/trs/2001/TR01-10.ps.

[67] J. Li, J. White, Low rank solution of Lyapunov equations, Submitted to *SIAM J. Matrix Anal. Appl*., 2001.

[68] D. C. Sorensen, A. C. Antoulas, *The Sylvester equation and approximate balanced reduction*, Technical report, 2002, Rice University, Houston, http://www-ece.rice.edu/~aca/crossgram.pdf.

[69] T. Penzl, *LYAPACK: A MATLAB Toolbox for Large Lyapunov and Riccati Equations, Model Reduction Problems, and Linear-Quadratic Optimal Control Problems*, version 1, 2000, http://www.netlib.org/lyapack/.

[70] J. Roychowdhury, Reduced-order modeling of time-varying systems, *IEEE Trans. Circuits Syst. II-Analog Digit. Signal Process.* 1999, 46, 1273-1288.

[71] J. Chen, S.-M. Kang. Techniques for Coupled Circuit and Micromechanical Simulation. in *Technical Proceedings of Third International Conference on Modeling and Simulation of Microsystems*, 2000, 213-216.

[72] Y. Chen, J. White. A Quadratic Method for Nonlinear Model Order Reduction, *Technical Procedings of the Third International Conference on Modeling and Simulation of Microsystems*, 2000, 477-480.

[73] B. Lohmann, Application of Model Order Reduction to a Hydropneumatic Vehicle Suspension, *IEEE Trans. Control Sys. Tech.*, 1995, 3, 102-109.

[74] B. Lohmann, Order Reduction and Determination of Dominant State Variables of Non-linear Systems, *Math. Mod. Systems*, 1995, 1, 77-90.

[75] J. R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. *Proceedings of DAC*, 2000.

[76] P. K. Gunupudi, M. S. Nakhla, Nonlinear circuit-reduction of high-speed interconnect networks using congruent transformation techniques, *IEEE Trans. Adv. Packag.* 2001, 24, 317-325.

[77] P. Holmes, J. L. Lumley, G. Berkooz, *Turbulence, coherent structures, dynamical systems, and symmetry*, Cambridge monographs on mechanics, Cambridge, New York: Cambridge University Press, 1996, pp. 420.

[78] P. Krysl, S. Lall, J. E. Marsden, Dimensional model reduction in non-linear finite element dynamics of solids and structures, *Int. J. Numer. Methods Eng.* 2001, 51, 479-504.

[79] S. Banerjee, J. V. Cole, K. F. Jensen, Nonlinear model reduction strategies for rapid thermal processing systems, *IEEE Trans. Semicond. Manuf.* 1998, 11, 266-275.

[80] M. Hinze, A. Kauffmann, *Reduced order modeling and suboptimal control of a solid fuel ignition model*, Report 636, Technische Universität, Berlin, 1999, http://www.math.Tu-Dresden.de/~hinze/Psfiles/SFIM.ps.gz.

[81] G. M. Kepler, H. T. Tran, H. T. Banks, Reduced order model compensator control of species transport in a CVD reactor, *Optim. Control Appl. Methods*, 2000, 21, 143-160.

[82] G. M. Kepler, H. T. Tran, H. T. Banks, Compensator control for chemical vapor deposition film growth using reduced-order design models, *IEEE Trans. Semicond. Manuf.* 2001, 14, 231-241.

[83] S. Y. Shvartsman, I. G. Kevrekidis, Low-dimensional approximation and control of periodic solutions in spatially extended systems, *Phys. Rev. E*, 1998, 58, 361-368.

[84] S. Y. Shvartsman, C. Theodoropoulos, R. Rico-Martinez, I. G. Kevrekidis, E. S. Titi, T. J. Mountziaris, Order reduction for nonlinear dynamic models of distributed reacting systems, *J. Process Control*, 2000, 10, 177-184.

[85] E. Bendersky, P. D. Christofides, Optimization of transport-reaction processes using nonlinear model reduction, *Chem. Eng. Sci.* 2000, 55, 4349-4366.

[86] S. Raimondeau, D. G. Vlachos, Low-dimensional approximations of multiscale epitaxial growth models for microstructure control of materials, *J. Comput. Phys.* 2000, 160, 564-576.

[87] P. A. Parrilo, S. Lall, F. Paganini, G. C. Verghese, B. C. Lesieutre, J. E. Marsden. Model reduction for analysis of cascading failures in power systems, in *Proceedings of the American Control Conference*, 1999.

[88] J. Baker, P. D. Christofides, Finite-dimensional approximation and control of non-linear parabolic PDE systems, *Int. J. Control*, 2000, 73, 439-456.

[89] H. T. Banks, R. C. H. del Rosario, R. C. Smith, Reduced-order model feedback control design: Numerical implementation in a thin shell model, *IEEE Trans. Autom. Control*, 2000, 45, 1312-1324.

[90] J. A. Atwell, B. B. King, Proper orthogonal decomposition for reduced basis feedback controllers for parabolic equations, *Math. Comput. Model.* 2001, 33, 1-19.

[91] I. T. Georgiou, J. Sansour, Analyzing the Finite Element Dynamics of Nonlinear In-Plane Rods by the Method of Proper Orthogonal Decomposition, in *Computational Mechanics, New Trends and Applications*, (Ed.: S. Idelsohn, E. Onate, E. Dvorkin), @CIMNE, Barcelona, Spain, 1998.

[92] I. T. Georgiou, I. B. Schwartz, Dynamics of large scale coupled structural mechanical systems: A singular perturbation proper orthogonal decomposition approach, *SIAM J. Appl. Math.* 1999, 59, 1178-1207.

[93] A. Steindl, H. Troger, Methods for dimension reduction and their application in nonlinear dynamics, *Int. J. Solids Struct.* 2001, 38, 2131-2147.

[94] E. S. Hung, S. D. Senturia, Generating efficient dynamical models for microelectromechanical systems from a few finite-element simulation runs, *J. Microelectromech. Syst.* 1999, 8, 280-289.

[95] Y. C. Liang, W. Z. Lin, H. P. Lee, S. P. Lim, K. H. Lee, D. P. Feng, A neural-network-based method of model reduction for the dynamic simulation of MEMS, *J. Micromech. Microeng.* 2001, 11, 226-233.

[96] M. Fahl, Computation of POD basis functions for fluid flows with Lanczos methods, *Math. Comput. Model.* 2001, 34, 91-107.

[97] Y. Chahlaoui, K. Gallivan, P. Van Dooren, Recursive calculation of dominant singular subspaces, submitted to *SIAM J. Matrix Anal. Appl.*, 2001.

[98] S. Lall, J. E. Marsden, S. Glavaski. Empirical model reduction of controlled nonlinear systems, in *Proceedings of the IFAC World Congress*, 1999.

[99] S. Lall, J. E. Marsden, S. Glavaski, A subspace approach to balanced truncation for model reduction of nonlinear control systems, to appear in *Int. J. Robust Nonlin. Control*, 2001.

[100] K. Gallivan, E. Grimme, P. Van Dooren, Model reduction of large-scale systems : rational Krylov versus balancing techniques, in *Error Control and Adaptivity in Scientific Computing*, (Ed: B. Zenger), 1999, 177-190.

[101] T. Antoulas, P. Van Dooren, *Model Reduction of large-scale dynamical systems*, Course presented at the SIAM Annual Meeting, 2000 , http://www.auto.ucl.ac.be/~vdooren/siam.html.

[102] J. White, *Techniques for Model Order Reduction*, http://rle-vlsi.mit.edu/people/users_krylov/white/public_html/pubs/Mor.ppt.

[103] Jens Müller, *Accurate FE-Simulation of Three-Dimensional Microstructures*, Fakultät für Angewandte Wissenschaften, Albert-Ludwigs Universität Freiburg im Breisgau, 2001, http://www.freidok.uni-freiburg.de/volltexte/251/pdf/thesis.pdf.

[104] O. G. Ernst, *Minimal and Orthogonal Residual Methods and their Generalizations for Solving Linear Operator Equations*, Fakultät für Mathematik und Informatik, TU Bergakademie Freiberg, 2000, http://www.mathe.tu-freiberg.de/~ernst/publ/habil.pdf.

[105] P. M. R. Wortelboer, *Frequency-weighted Balanced Reduction of Closed-loop Mechanical Servo-systems: Theory and Tools*, Delft University of Technology, 1994, http://www.ocp.tudelft.nl/sr/Downloads/thesis/wortelboer/wortelb.zip.

[106] E. J. Grimme, *Krylov Projection Methods For Model Reduction*, Graduate College, University of Illinois, Urbana-Champaign, 1997, http://www.cs.fsu.edu/~gallivan/theses/grimmephd.ps.gz.

[107] D. Skoogh, *Krylov subspace methods for linear systems, eigenvalues and model reduction*, Departemnt of mathematics, Göteborg, 1998, http://www.md.chalmers.se/~skoogh/skooghPHD.html.

[108] J.-R. Li, *Model Reduction of Large Linear Systems via Low Rank System Gramians*, Department of Mathematics, MIT, Boston, 2000, http://www.cims.nyu.edu/~jingli/thesis.ps.

[109] A. J. Newman, *Modeling and Reduction with Applications to Semiconductor Processing*, Department of Electrical and Computer Engineering, University of Maryland, College Park, 1999, http://www.isr.umd.edu:80/TechReports/ISR/1999/PhD_99-5/PhD_99-5.phtml.

[110] J. A. Atwell, *Proper Orthogonal Decomposition for Reduced Order Control of Partial Differential Equations*, Virginia Polytechnic Institute, 2000, http://www.math.vt.edu/people/bbking/papers/atwell.pdf.