Introduction to Model Order Reduction

Wil Schilders[1,2]

[1] NXP Semiconductors, Eindhoven, The Netherlands
wil.schilders@nxp.com
[2] Eindhoven University of Technology, Faculty of Mathematics and Computer Science, Eindhoven, The Netherlands
w.h.a.schilders@tue.nl

1 Introduction

In this first section we present a high level discussion on computational science, and the need for compact models of phenomena observed in nature and industry. We argue that much more complex problems can be addressed by making use of current computing technology and advanced algorithms, but that there is a need for model order reduction in order to cope with even more complex problems. We also go into somewhat more detail about the question as to what model order reduction is.

1.1 Virtual Design Environments

Simulation or, more generally, computational science has become an important part of todays technological world, and it is now generally accepted as the third discipline, besides the classical disciplines of theory and (real) experiment. Physical (and other) experiments lead to theories that can be validated by performing additional experiments. Predictions based on the theory can be made by performing virtual experiments, as illustrated by Figure 1.

Computer simulations are now performed routinely for many physical, chemical and other processes, and virtual design environments have been set up for a variety of problem classes in order to ease the work of designers and engineers. In this way, new products can be designed faster, more reliably, and without having to make costly prototypes.

The ever increasing demand for realistic simulations of complex products places a heavy burden on the shoulders of mathematicians and, more generally, researchers working in the area of computational science and engineering (CSE). Realistic simulations imply that the errors of the virtual models should be small, and that different aspects of the product must be taken into account. The former implies that care must be taken in the numerical treatment and that, for example, a relatively fine adaptively determined mesh is necessary in the simulations. The latter explains the trend in coupled simulations, for example combined mechanical and thermal behaviour, or combined mechanical and electromagnetic behaviour.

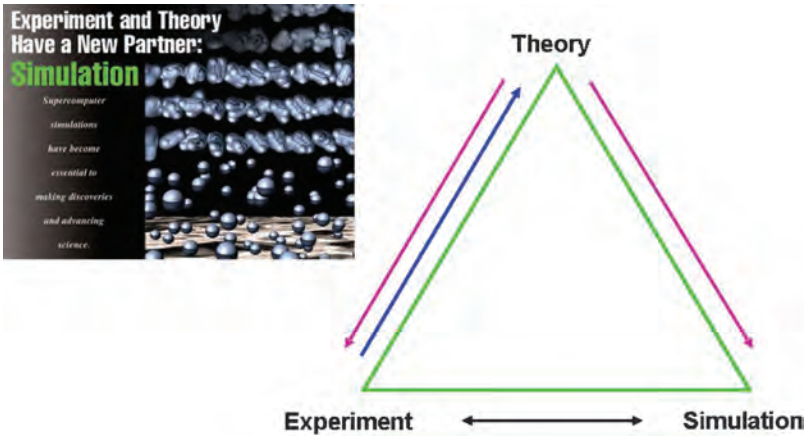

Fig. 1. Simulation is the third discipline.

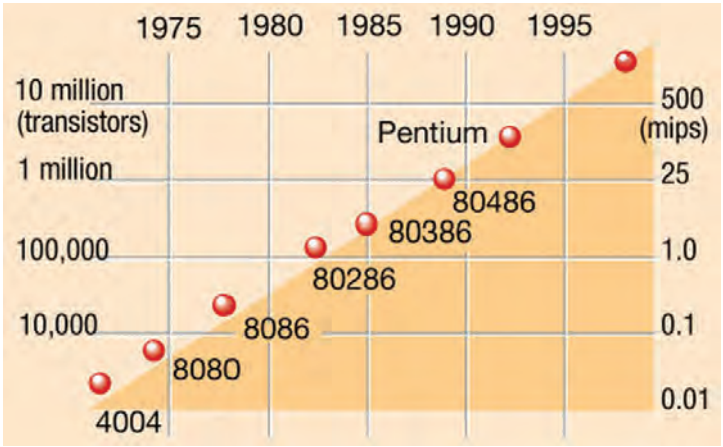

Fig. 2. Moore's law.

An important factor in enabling the complex simulations carried out today is the increase in computational power. Computers and chips are getting faster, Moores law predicting that the speed will double every 18 months (see Figure 2).

This increase in computational power appears to go hand-in-hand with developments in numerical algorithms. Iterative solution techniques for linear systems are mainly responsible for this speed-up in algorithms, as is shown in Figure 3. Important contributions in this area are the conjugate gradient method (Hestenes and Stiefel [22]), preconditioned conjugate gradient methods (ICCG [25], biCGstab [34]) and multigrid methods (Brandt [4] and [5]).

The combined speed-up achieved by computer chips and algorithms is enormous, and has enabled computational science to make big steps forward. Many problems that people did not dream of solving two decades ago are now solved routinely.

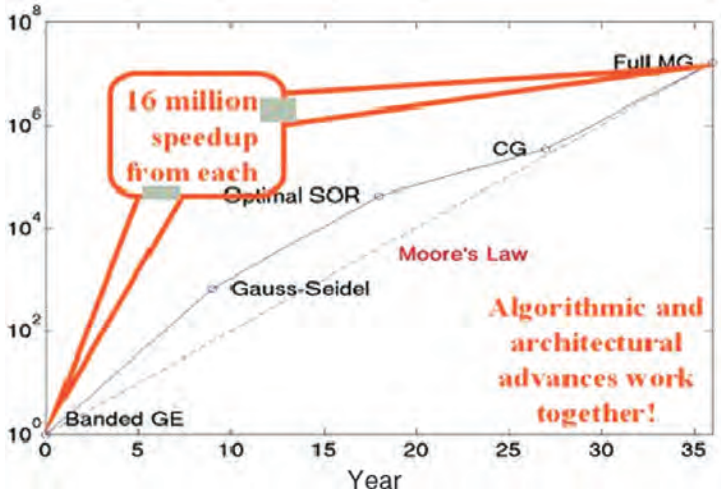

Fig. 3. Numerical version of Moore's law.

1.2 Compact Model Descriptions

The developments described in the previous section also have a counter side. The increased power of computers and algorithms reduces the need to develop smart, sophisticated solution methods that make use of properties of the underlying systems. For example, whereas in the 1960s and 1970s one often had to construct special basis functions to solve certain problems, this can be avoided nowadays by using brute force methods using grids that are refined in the right places.

The question arises whether we could use the knowledge generated by these very accurate, but time-consuming, simulations to generate the special basis functions that would have constituted the scientific approach a few decades ago. This is a promising idea, as many phenomena are described very well by a few dominant modes.

Example: electromagnetic behaviour of interconnect structures in chips
To give an example, consider the electromagnetic behaviour of interconnect structures in a computer chip, depicted in Figure 4. Such a chip consists of millions of devices, such as transistors, that need to be connected to each other for correct functioning of the chip. The individual devices are contained in the semiconductor material, their contacts being located in a two dimensional domain. Clearly, to connect these contacts in the way designers have prescribed, a three dimensional structure of wires is needed. This is the so-called interconnect structure of the chip, which nowadays consists of 7-10 layers in which metal wires are running, with so-called vias between the wires located in different metal layers.

In previous generations of chips, these interconnect structures occupied a relatively large area, and contained less wires, so that the distance between wires was large enough to justify discarding mutual influence. In recent years, however, chips have shrunk, and the number of devices has grown enormously. This means that for modern interconnect structures one needs to take into account mutual influence of wires, as this can lead to serious delay phenomena and other spurious effects. The problem is complicated even further by the use of higher frequencies.

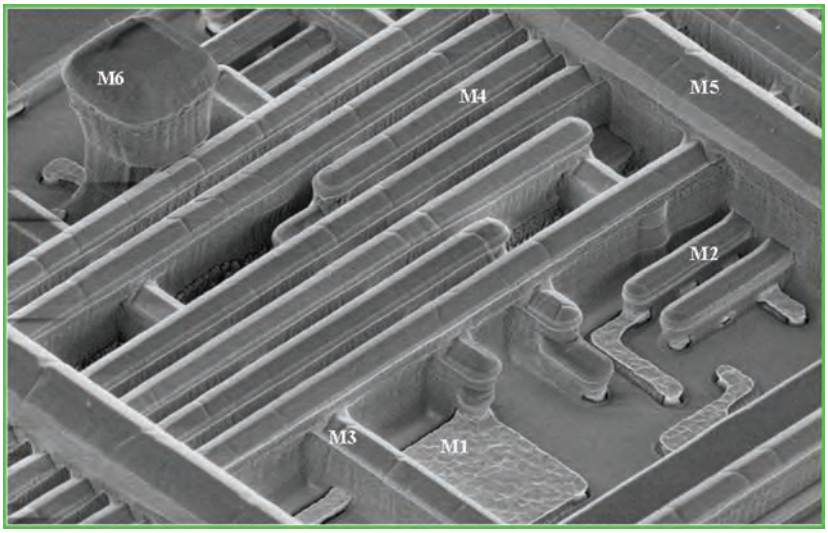

Fig. 4. Interconnect structure.

Clearly, the modelling of the mutual electromagnetic influence of interconnect wires is a gradual process. A decade ago, one did not have to take this influence into account, and could consider the wires as individual entities. Nowadays, resistive and capacitive effects are clearly noticeable, and will become more significant over the years. Because of the gradual character of this phenomenon, one can imagine that it is not necessary to include all minute detail of an electromagnetic simulation of interconnect structures. Such a simulation could easily involve millions of nodes, because of the complicated geometric structure. The simulation will probably reveal that crosstalk and signal integrity problems are quite localized, at a few places in the structure where wires are too close together.

Another point of view may be to consider the problem as an input-output model, where a time-dependent input signal is sent through the interconnect structure, and a resulting time-dependent output signal is registered. Again, to calculate the output resulting from the given input is a time-consuming exercise due to the excessive number of nodes necessary for this simulation, in the spatial and time domain. However, it is expected to be possible to delete superfluous detail, and calculate a very good approximation to the output in a much more efficient way.

The foregoing example clearly shows that it may not be necessary to calculate all details, and nevertheless obtain a good understanding of the phenomena taking place. There may be many reasons why such detail is not needed. There may be physical reasons that can be formulated beforehand, and therefore incorporated into the model before starting calculations. A very nice example is that of simulating the blood flow in the human body, as described in many publications by the group of Alfio Quarteroni (see [30], but also work of others). In his work, the blood flow in the body is split into different parts. In very small arteries, it is assumed that the flow is one dimensional. In somewhat larger arteries, two dimensional models are used, whereas in the heart, a three dimensional model is used as these effects

are very important and must be modelled in full detail. This approach does enable a simulation of the blood flow in the entire human body; clearly, such simulations would not be feasible if three dimensional models would be used throughout. This approach, which is also observed in different application areas, is also termed *operational model order reduction*. It uses physical (or other) insight to reduce the complexity of models.

Another example of operational model order reduction is the simulation of electromagnetic effects in special situations. As is well known, electromagnetic effects can be fully described by a system of Maxwell equations. Despite the power of current computers and algorithms, solving the Maxwell equations in 3-dimensional space and time is still an extremely demanding problem, so that simplifications are being made whenever possible. An assumption that is made quite often is that of quasi-statics, which holds whenever the frequencies playing a role are low to moderate. In this case, simpler models can be used, and techniques for solving these models have been developed (see [32]).

In special situations, the knowledge about the problem and solutions can be so detailed, that a further reduction of model complexity can be achieved. A prominent and very successful example is the *compact modelling* [19] of semiconductor devices. Integrated circuits nowadays consist of millions of semiconductor devices, such as resistors, capacitors, inductors, diodes and transistors. For resistors, capacitors and inductors, simple linear models are available, but diodes and especially transistors are much more complicated. Their behaviour is not easily described, but can be calculated accurately using software dedicated to semiconductor device simulation. However, it is impossible to perform a full simulation of the entire electronic circuit, by using the results of the device simulation software for each of the millions of transistors. This would imply coupling of the circuit simulation software to the device simulation software. Bearing in mind that device simulations are often quite time consuming (it is an extremely nonlinear problem, described by a system of three partial differential equations), this is an impossible task.

The solution to the aforementioned problem is to use accurate compact models for each of the transistors. Such models look quite complicated, and can easily occupy a number of pages of description, but consist of a set of algebraic relations that can be evaluated very quickly. The compact models are constructed using a large amount of measurements and simulations, and, above all, using much human insight. The models often depend on as many as 40-50 parameters, so that they are widely applicable for many different types and geometries of transistors. The most prominent model nowadays is the Penn-State-Philips (PSP) model for MOS transistors (see Figure 5), being chosen as the world standard in 2007 [15]. It is very accurate, including also derivatives up to several orders. Similar developments can be observed at Berkeley [6], where the BSIM suite of models is constructed.

Using these so-called compact models, it is possible to perform simulations of integrated circuits containing millions of components, both for steady-state and time-dependent situations. Compact modelling, therefore, plays an extremely important role in enabling such demanding simulations. The big advantage of this approach is that the compact models are formulated in a way that is very appealing to designers, as they are formulated in terms of components they are very familiar with.

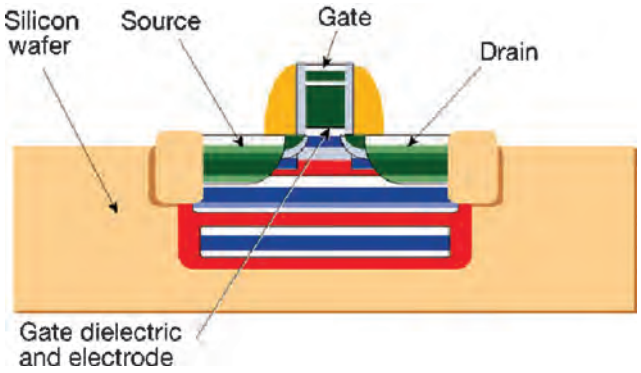

Fig. 5. MOS transistor.

Unfortunately, in many cases, it is not possible to a priori simplify the model describing the behaviour. In such cases, a procedure must be used, in which we rely on the automatic identification of potential simplifications. Designing such algorithms is, in essence, the task of the field of model order reduction. In the remainder of this chapter, we will describe it in more detail.

1.3 Model Order Reduction

There are several definitions of model order reduction, and it depends on the context which one is preferred. Originally, MOR was developed in the area of systems and control theory, which studies properties of dynamical systems in application for reducing their complexity, while preserving their input-output behavior as much as possible. The field has also been taken up by numerical mathematicians, especially after the publication of methods such as PVL [9]. Nowadays, model order reduction is a flourishing field of research, both in systems and control theory and in numerical analysis. This has a very healthy effect on MOR as a whole, bringing together different techniques and different points of view, pushing the field forward rapidly.

So what is model order reduction about? As was mentioned in the foregoing sections, we need to deal with the simplification of dynamical models that may contain many equations and/or variables ($10^5 - 10^9$). Such simplification is needed in order to perform simulations within an acceptable amount of time and limited storage capacity, but with reliable outcome. In some cases, we would even like to have on-line predictions of the behaviour with acceptable computational speed, in order to be able to perform optimizations of processes and products.

Model Order Reduction tries to quickly capture the essential features of a structure. This means that in an early stage of the process, the most basic properties of the original model must already be present in the smaller approximation. At a certain moment the process of reduction is stopped. At that point all necessary properties of the original model must be captured with sufficient precision. All of this has to be done automatically.

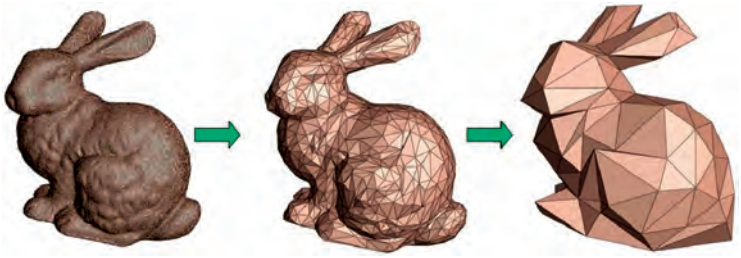

Fig. 6. Graphical illustration of model order reduction.

Figure 6 illustrates the concept in a graphical easy-to-understand way, demonstrating that sometimes very little information is needed to describe a model. This example with pictures of the Stanford Bunny shows that, even with only a few facets, the rabbit can still be recognized as such (Graphics credits: Harvard University, Microsoft Research). Although this example was constructed for an entirely different purpose, and does not contain any reference to the way model order reduction is performed mathematically, it can be used to explain (even to lay persons) what model order reduction is about.

In the history of mathematics we see the desire to approximate a complicated function with a simpler formulation already very early. In the year 1807 Fourier (1768-1830) published the idea to approximate a function with a few trigonometric terms. In linear algebra the first step in the direction of model order reduction came from Lanczos (1893-1974). He looked for a way to reduce a matrix in tridiagonal form [64, 65]. W.E. Arnoldi realized that a smaller matrix could be a good approximation of the original matrix [2]. He is less well-known, although his ideas are used by many numerical mathematicians. The ideas of Lanczos and Arnoldi were already based on the fact that a computer was available to do the computations. The question, therefore, was how the process of finding a smaller approximation could be automated.

The fundamental methods in the area of Model Order Reduction were published in the eighties and nineties of the last century. In 1981 Moore [71] published the method of Truncated Balanced Realization, in 1984 Glover published his famous paper on the Hankel-norm reduction [38]. In 1987 the Proper Orthogonal Decomposition method was proposed by Sirovich [94]. All these methods were developed in the field of systems and control theory. In 1990 the first method related to Krylov subspaces was born, in Asymptotic Waveform Evaluation [80]. However, the focus of this paper was more on finding Padé approximations rather than Krylov spaces. Then, in 1993, Freund and Feldmann proposed Padé Via Lanczos [28] and showed the relation between the Padé approximation and Krylov spaces. In 1995 another fundamental method was published. The authors of [73] introduced PRIMA, a method based on the ideas of Arnoldi, instead of those of Lanczos. This method will be considered in detail in Section 3.3, together with the Laguerre-SVD method [61].

In more recent years much research has been done in the area of the Model Order Reduction. Consequently a large variety of methods is available. Some are tailored

to specific applications, others are more general. In the second and third part of this book, many of these new developments are being discussed. In the remainder of this chapter, we will discuss some basic methods and properties, as this is essential knowledge required for the remainder of the book.

1.4 Dynamical Systems

To place model reduction in a mathematical context, we need to realize that many models developed in computational science consist of a system of partial and/or ordinary differential equations, supplemented with boundary conditions. Important examples are the Navier-Stokes equations in computational fluid dynamics (CFD), and the Maxwell equations in electromagnetics (EM). When partial differential equations are used to describe the behaviour, one often encounters the situation that the independent variables are space and time. Thus, after (semi-)discretising in space, a system of ordinary differential equations is obtained in time. Therefore, we limit the discussion to ODE's and consider the following explicit finite-dimensional dynamical system (following Antoulas, see [2]):

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} &= \mathbf{g}(\mathbf{x},\mathbf{u}). \end{aligned}$$

Here, $\mathbf{u}$ is the input of the system, $\mathbf{y}$ the output, and $\mathbf{x}$ the so-called *state variable*. The dynamical system can thus be viewed as an *input-output system*, as displayed in Figure 7.

The complexity of the system is characterized by the number of its state variables, i.e. the dimension n of the state space vector $\mathbf{x}$. It should be noted that similar dynamical systems can also be defined in terms of differential algebraic equations, in which case the first set of equations in (1) is replaced by $\mathbf{F}(\frac{d\mathbf{x}}{dt},\mathbf{x},\mathbf{u}) = 0$.

Model order reduction can now be viewed as the task of reducing the dimension of the state space vector, while preserving the character of the input-output relations. In other words, we should find a dynamical system of the form

$$\begin{aligned} \frac{d\hat{\mathbf{x}}}{dt} &= \hat{\mathbf{f}}(\hat{\mathbf{x}},\mathbf{u}), \\ \mathbf{y} &= \hat{\mathbf{g}}(\hat{\mathbf{x}},\mathbf{u}), \end{aligned}$$

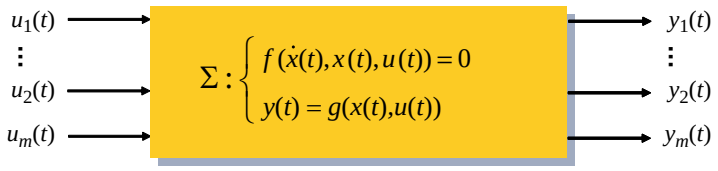

Fig. 7. Input-output system

where the dimension of $\hat{\mathbf{x}}$ is much smaller than n. In order to provide a good approximation of the original input-output system, a number of conditions should be satisfied:

- the approximation error is small,
- preservation of properties of the original system, such as stability and passivity (see Sections 2.4-2.6),
- the reduction procedure should be computationally efficient.

A special case is encountered if the functions $\mathbf{f}$ and $\mathbf{g}$ are linear, in which case the system reads

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= A\mathbf{x} + B\mathbf{u}, \\ \mathbf{y} &= C^T\mathbf{x} + D\mathbf{u}. \end{aligned}$$

Here, the matrices A, B, C, D can be time-dependent, in which case we have a *linear time-varying (LTV) system*, or time-independent, in which case we speak about a *linear time-invariant (LTI) system*. For linear dynamical systems, model order reduction is equivalent to reducing the matrix A, but retaining the number of columns of B and C.

1.5 Approximation by Projection

Although we will discuss in more detail ways of approximating input-output systems later in this chapter, there is a unifying feature of the approximation methods that is worthwhile discussing briefly: *projection*. Methods based on this concept truncate the solution of the original system in an appropriate basis. To illustrate the concept, consider a basis transformation T that maps the original n-dimensional state space vector $\mathbf{x}$ into a vector that we denote by

$$\bar{\mathbf{x}} = \begin{pmatrix} \hat{\mathbf{x}} \\ \tilde{\mathbf{x}} \end{pmatrix},$$

where $\hat{\mathbf{x}}$ is k-dimensional. The basis transformation T can then be written as

$$T = \begin{pmatrix} W^* \\ T_2^* \end{pmatrix},$$

and its inverse as

$$T^{-1} = (V \;\; T_1).$$

Since $W^*V = I_k$, we conclude that

$$\Pi = VW^*$$

is an oblique projection along the kernel of W^* onto the k-dimensional subspace that is spanned by the columns of the matrix V.

If we substitute the projection into the dynamical system (1), the first part of the set of equations obtained is

$$\begin{aligned}\frac{d\hat{\mathbf{x}}}{dt} &= W^*\hat{\mathbf{f}}(V\hat{\mathbf{x}} + T_1\tilde{\mathbf{x}}, \mathbf{u}),\\ \mathbf{y} &= \hat{\mathbf{g}}(V\hat{\mathbf{x}} + T_1\tilde{\mathbf{x}}, \mathbf{u}).\end{aligned}$$

Note that this is an exact expression. The approximation occurs when we would delete the terms involving $\tilde{\mathbf{x}}$, in which case we obtain the reduced system

$$\begin{aligned}\frac{d\hat{\mathbf{x}}}{dt} &= W^*\hat{\mathbf{f}}(V\hat{\mathbf{x}}, \mathbf{u}),\\ \mathbf{y} &= \hat{\mathbf{g}}(V\hat{\mathbf{x}}, \mathbf{u}).\end{aligned}$$

For this to produce a good approximation to the original system, the neglected term $T_1\tilde{\mathbf{x}}$ must be sufficiently small. This has implications for the choice of the projection Π. In the following sections, various ways of constructing this projection are discussed.

2 Transfer Function, Stability and Passivity

Before discussing methods that have been developed in the area of model order reduction, it is necessary to shed light on several concepts that are being used frequently in the field. Often, model order reduction does not address the reduction of the entire problem or solution, but merely a number of characteristic functions that are important for designers and engineers. In addition, it is important to consider a number of specific aspects of the underlying problem, and preserve these when reducing. Therefore, this section is dedicated to a discussion of these important concepts.

2.1 Transfer Function

In order to illustrate the various concepts related to model order reduction of input-output systems as described in the previous section, we consider the linear time-invariant system

$$\begin{aligned}\frac{d\mathbf{x}}{dt} &= A\mathbf{x} + B\mathbf{u},\\ \mathbf{y} &= C^T\mathbf{x}.\end{aligned}$$

The general solution of this problem is

$$\mathbf{x}(t) = \exp\left(A(t-t_0)\right)\mathbf{x}_0 + \int_{t_0}^{t} \exp\left(A(t-\tau)\right)B\mathbf{u}(\tau)d\tau. \tag{1}$$

A common way to solve the differential equation is by transforming it from the time domain to the frequency domain, by means of a Laplace transform defined as

$$\mathcal{L}(f)(s) \equiv \int_0^\infty f(t) \exp(-st) dt.$$

If we apply this transform to the system, assuming that $\mathbf{x}(0) = 0$, the system is transformed to a purely algebraic system of equations:

$$\begin{aligned}(I_n - sA)\mathbf{X} &= B\mathbf{U}, \\ \mathbf{Y} &= C^T\mathbf{X},\end{aligned}$$

where the capital letters indicate the Laplace transforms of the respective lower case quantities. This immediately leads to the following relation:

$$\mathbf{Y}(s) = C^T(I_n - sA)B\mathbf{X}(s). \tag{2}$$

Now define the *transfer function* $H(s)$ as

$$H(s) = C^T(I_n - sA)B. \tag{3}$$

This transfer function represents the direct relation between input and output in the frequency domain, and therefore the behavior of the system in frequency domain. For example, in the case of electronic circuits this function may describe the transfer from currents to voltages, and is then termed impedance. If the transfer is from voltages to currents, then the transfer function corresponds to the admittance.

Note that if the system has more than one input or more than one output, then B and C have more than one column. This makes $H(s)$ a matrix function. The i, j entry in $H(s)$ then denotes the transfer from input i to output j.

2.2 Moments

The transfer function is a function in s, and can therefore be expanded into a moment expansion around $s = 0$:

$$H(s) = M_0 + M_1 s + M_2 s^2 + \ldots,$$

where M_0, M_1, M_2, ... are the *moments* of the transfer function. In electronics, M_0 corresponds to the DC solution.In that case the inductors are considered as short circuits, and capacitors as open circuits. The moment M_1 then corresponds to the so-called Elmore delay, which represents the time for a signal at the input port to reach the output port. The Elmore delay is defined as

$$t_{elm} \equiv \int_0^\infty t h(t) dt,$$

where $h(t)$ is the *impulse response function*, which is the response of the system to the Dirac delta input. The transfer function in the frequency domain is the Laplace transform of the impulse response function:

$$H(s) = \int_0^\infty h(t) \exp(-st) dt.$$

Expanding the exponential function in a power series, it is seen that the Elmore delay indeed corresponds to the first order moment of the transfer function.

Of course, the transfer function can also be expanded around some non-zero s_0. We then obtain a similar expansion in terms of moments. This may be advantageous in some cases, and truncation of that alternative moment expansion may lead to better approximations.

2.3 Poles and Residues

The transfer function can also be expanded as follows:

$$H(s) = \sum_{j=1}^{n} \frac{R_j}{s - p_j}, \tag{4}$$

where the p_j are the poles, and R_j are the corresponding residues. The poles are exactly the eigenvalues of the matrix $-A^{-1}$. In fact, if the matrix E of eigenvectors is non-singular, we can write

$$-A^{-1} = E \Lambda E^{-1},$$

where the diagonal matrix Λ contains the eigenvalues λ_j. Substituting this into the expression for the transfer function, we obtain:

$$H(s) = -C^T E(I + s\Lambda)^{-1} E^{-1} A^{-1} B.$$

Hence, if B and C contain only one column (which corresponds to the single input, single output or SISO case), then

$$H(s) = \sum_{j=1}^{n} \frac{l_j^T r_j}{1 + s\lambda_j},$$

where the l_j and r_j are the left and right eigenvectors, respectively.

We see that there is a one-to-one relation between the poles and the eigenvalues of the system. If the original dynamical system originates from a differential algebraic system, then a generalized eigenvalue problem needs to be solved. Since the poles appear directly in the pole-residue formulation of the transfer function, there is also a strong relation between the transfer function and the poles or, stated differently, between the behavior of the system and the poles. If one approximates the system, one should take care to approximate the most important poles. There are several

methods that do this, which are discussed in later chapters of this book. In general, we can say that, since the transfer function is usually plotted for imaginary points $s = \omega i$, the poles that have a small imaginary part dictate the behavior of the transfer function for small values of the frequency ω. Consequently, the poles with a large imaginary part are needed for a good approximation at higher frequencies. Therefore, a successful reduction method aims at capturing the poles with small imaginary part rather, and leaves out poles with a small residue.

2.4 Stability

Poles and eigenvalues of a system are strongly related to the stability of the system. Stability is the property of a system that ensures that the output signal of a system is limited (in the time domain).

Consider again the system (1). The system is *stable* if and only if, for all eigenvalues λ_j, we have that $Re(\lambda_j) \leq 0$, and all eigenvalues with $Re(\lambda_j) = 0$ are simple. In that case, the corresponding matrix A is termed stable.

There are several properties associated with stability. Clearly, if A is stable, then also A^{-1} is stable. Stability of A also implies stability of A^T and stability of A^*. Finally, if the product of matrices AB is stable, then also BA can be shown to be stable. It is also clear that, due to the relation between eigenvalues of A and poles of the transfer function, stability can also be formulated in terms of the poles of $H(s)$.

The more general linear dynamical system

$$\begin{aligned} Q\frac{d\mathbf{x}}{dt} &= A\mathbf{x} + B\mathbf{u}, \\ \mathbf{y} &= C^T\mathbf{x}, \end{aligned}$$

is stable if and only if for all generalized eigenvalues we have that $Re(\lambda_j(Q, A)) \leq 0$, and all generalized eigenvalues for which $Re(\lambda_j(Q, A)) = 0$ are simple. The set of generalized eigenvalues $\sigma(Q, A)$ is defined as the collection of eigenvalues of the generalized eigenvalue problem

$$Q\mathbf{x} = \lambda A\mathbf{x}.$$

In this case, the pair of matrices (Q, A) is termed a *matrix pencil*. This pencil is said to be regular if there exists at least one eigenvalue λ for which $Q + \lambda A$ is regular. Just as for the simpler system discussed in the above, stability can also be formulated in terms of the poles of the corresponding transfer function.

2.5 Positive Real Matrices

The concept of stability explained in the previous subsection leads us to consider other properties of matrices. First we have the following theorem.

Theorem 1. *If $Re(\mathbf{x}^* A\mathbf{x} > 0$ for all $x \in C^n$, then all eigenvalues of A have a positive real part.*

The converse of this theorem is not true, as can be seen when we take

$$A = \begin{pmatrix} -\frac{1}{3} & -1 \\ 1 & 2 \end{pmatrix},$$

and

$$\mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Matrices with the property that $Re(\mathbf{x}^* A\mathbf{x} > 0$ for all $x \in C^n$ are termed *positive real*. The counter example shows that the class of positive real matrices is smaller than the class of matrices for which all eigenvalues have positive real part. In the next section, this new and restricted class will be discussed in more detail. For now, we remark that a number of properties of positive real matrices are easily derived. If A is positive real, then this also holds for A^{-1} (if it exists). Furthermore, A is positive real if and only if A^* is positive real. If two matrices A and B are both positive real, then any linear combination $\alpha A + \beta B$ is also positive real provided $Re(\alpha) > 0$ and $Re(\beta) > 0$.

There is an interesting relation between positive real and positive definite matrices. Evidently, the class of positive definite matrices is a subclass of the set of positive real matrices. But we also have:

Theorem 2. *A matrix $A \in C^{n\times n}$ is positive real if and only if the Hermitian part of A (i.e. $\frac{1}{2}(A + A^*)$) is symmetric positive definite.*

Similarly one can prove that a matrix is non-negative real if and only if its Hermitian part is symmetric positive semi-definite.

2.6 Passivity

Stability is a very natural property of physical structures. However, stability is not strong enough for electronic structures that contain no sources. A stable structure can become unstable if non-linear components are connected to it. Therefore, another property of systems should be defined that is stronger than stability. This property is called *passivity*. Being passive means being incapable of generating energy. If a system is passive and stable, we would like a reduction method to preserve these properties during reduction. In this section the principle of passivity is discussed and what is needed to preserve passivity.

To define the concept, we consider a system that has N so-called ports. The total instantaneous power absorbed by this real N-port is defined by:

$$w_{inst}(t) \equiv \sum_{j=1}^{N} v_j(t) i_j(t),$$

where $v_j(t)$ and $i_j(t)$ are the real instanteneous voltage and current at the j-th port. An N-port contains stored energy, say $E(t)$. If the system dissipates energy at rate

$w_d(t)$, and contains sources which provide energy at rate $w_s(t)$, then the energy balance during a time interval $[t_1, t_2]$ looks like:

$$\int_{t_1}^{t_2} (w_{inst} + w_s - w_d)dt = E(t_2) - E(t_1). \tag{5}$$

An N-port is termed *passive* if we have

$$\int_{t_1}^{t_2} w_{inst}dt \geq E(t_2) - E(t_1), \tag{6}$$

over any time interval $[t_1, t_2]$. This means that the increase in stored energy must be less than or equal to the energy delivered through the ports. The N-port is called lossless if (6) holds with equality over any interval. Assume that the port quantities exhibit purely exponential time-dependence, at a single complex frequency s. We may then write:

$$v(t) = \hat{v}\exp(it), i(t) = \hat{i}\exp((it)),$$

where $\hat{v}$ and $\hat{i}$ are the complex amplitudes. We define the total complex power absorbed to be the inner product of $\hat{i}$ and $\hat{v}$,

$$w = \hat{i}^*\hat{v},$$

and the average or active power as:

$$< w >= Re(w).$$

For an N-port defined by an impedance relationship, we may immediately write $< w >$ in terms of the voltage and current amplitudes:

$$< w >= \frac{1}{2}(\hat{i}^*\hat{v} + \hat{v}^*\hat{i}) = \frac{1}{2}(\hat{i}^*Z\hat{i} + \hat{i}^*Z^*\hat{i}) = \frac{1}{2}(\hat{i}^*(Z + Z^*)\hat{i}).$$

For such a real linear time invariant (LTI) N-port, passivity may be defined in the following way. If the total active power absorbed by an N-port is always greater than or equal to zero for frequencies s such that $Re(s) \geq 0$, then it is called passive. This implies that $Z + Z^* \geq 0$ for $Re(s) \geq 0$. Hence, the matrix Z must be positive real.

Given our discussion and the definition of passivity based on an energy argument, we can formulate the following theorem.

Theorem 3. *The transfer function $H(s)$ of a passive system is positive real, i.e.*

$$H^*(s) + H(s) \geq 0$$

for all s with $Re(s) \geq 0$.

Sometimes another definition of passivity is used, for instance in [35]. Under certain assumptions these definitions are equal.

3 A Short Account of Techniques for Model Order Reduction

Having discussed the need for model order reduction, and several essential prerequisites, this section will now give an overview of the field by discussing the most important techniques. The methods and new developments discussed in subsequent chapters of this book build on these basic algorithms. Adequate referencing is provided, so that readers can go into more detail when desired.

3.1 Asymptotic Waveform Evaluation

One of the basic and earliest methods in Model Order Reduction is Asymptotic Waveform Evaluation (AWE), proposed by Pillage and Rohrer in 1990 [7, 29]. The underlying idea of the method is that the transfer function can be well approximated by a Padé approximation. A Padé approximation is a ratio of two polynomials $P(s)$ and $Q(s)$. AWE calculates a Padé approximation of finite degree, so the degree of $P(s)$ and $Q(s)$ is finite and $deg(Q(s)) \geq deg(P(s))$. There is a close relation between the Padé approximations and Krylov subspace methods (see Chapter 2). To explain this fundamental property, consider the general system:

$$(sI_n - A)\mathbf{X}(s) = B\mathbf{U}(s).$$

Expanding $\mathbf{X}(s)$ around some expansion point $s_0 \in C$ we obtain:

$$(s_0 I_n - A + (s - s_0)I_n)(\mathbf{X}_0 + (s - s_0)\mathbf{X}_1 + \ldots) = B\mathbf{U}(s).$$

Here, the $\mathbf{X}_i(s)$ are the moments. Assuming $\mathbf{U}(s) = 1$, and equating like powers of $(s - s_0)^i$, we find:

$$(s_0 I_n - A)\mathbf{X}_0 = B,$$

for the term corresponding to $i = 0$, and for $i \geq 1$

$$(s_0 I_n - A)\mathbf{X}_i = -X_{i-1}.$$

We conclude that, in fact, a Krylov space is built up (see Chapter 2):

$$\mathcal{K}((s_0 I_n - A)^{-1}B, (s_0 I_n - A)^{-1}).$$

The process can be terminated after finding a sufficient number of moments, and the hope is that then a good approximation has been found for the transfer function. Clearly, this approximation is of the form

$$\tilde{H}(s) = \sum_{k=0}^{n} m_k (s - s_0)^k,$$

for some finite n.

Once the moments have been calculated, a Padé approximation $\hat{H}(s)$ of the transfer function $H(s)$ can be determined:

$$\hat{H}(s) = \frac{P(s-s_0)}{Q(s-s_0)}.$$

Letting

$$P(s) = \sum_{k=0}^{p} a_k (s-s_0)^k, \qquad Q(s) = \sum_{k=0}^{p+1} b_k (s-s_0)^k,$$

we find that the following relation must hold:

$$\sum_{k=0}^{p} a_k (s-s_0)^k = \left(\sum_{k=0}^{n} m_k (s-s_0)^k \right) \left(\sum_{k=0}^{p+1} b_k (s-s_0)^k \right).$$

Equating like powers of $s - s_0$ (for the higher powers), and setting $b_0 = 1$, we find the following system to be solved:

$$\begin{pmatrix} m_0 & m_1 & \dots & m_p \\ m_1 & m_2 & \dots & m_{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ m_p & m_{p+1} & \dots & m_{2p} \end{pmatrix} \begin{pmatrix} b_{p+1} \\ b_p \\ \vdots \\ b_1 \end{pmatrix} = - \begin{pmatrix} m_{p+1} \\ m_{p+2} \\ \vdots \\ m_{2p+1} \end{pmatrix}, \tag{1}$$

from which we can extract the coefficients $b_i, i = 1, \dots, p+1$ of Q. A similar step can be used to subsequently find the coefficients of the polynomial P.

The problem with the above method, and with AWE in general, is that the coefficient matrix in (1) quickly becomes ill-conditioned as the number of moments used goes up. In fact, practical experience indicates that applicability of the method stops once 8 or more moments are used. The method can be made more robust by using *Complex Frequency Hopping* [29], meaning that more than one expansion point is used. However, the method remains computationally demanding and, for that reason, alternatives as described in the next subsections are much more popular nowadays.

3.2 The PVL Method

Although AWE was initially considered an attractive method for approximating the transfer function, soon the disadvantages of the method were recognized. Then, in 1993, Roland Freund and Peter Feldmann [9] published their method named *Padé-via-Lanczos* or *PVL*. In this method, the Padé approximation based on the moments is calculated by means of a two-sided Lanczos algorithm (see also Chapter 2). The algorithm requires approximately the same computational effort as AWE, but it generates more poles and is much more robust.

To explain the method, consider the transfer function of a SISO system:

$$H(s) = \mathbf{c}^T (sI_n - A)^{-1} \mathbf{b}.$$

Let $s_0 \in C$ be the expansion point for the approximation. Then the transfer function can be cast into the form

$$H(s) = \mathbf{c}^T (I_n - (s-s_0)\hat{A})^{-1} \mathbf{r},$$

where

$$\hat{A} = -(s_0 I_n - A)^{-1},$$

and

$$\mathbf{r} = (s_0 I_n - A)^{-1}\mathbf{b}.$$

This transfer function can, just as in the case of AWE, be approximated well by a rational function in the form of a Padé approximation. In PVL this approximation is found via the Lanczos algorithm. By running q steps of this algorithm (see Chapter 2 for details on the Lanczos method), an approximation of $\hat{A}$ is found in the form of a tridiagonal matrix T_q, and the approximate transfer function is of the form:

$$H_q(s) = \mathbf{c}^T\mathbf{r} \cdot \mathbf{e}_1^T(I_n - (s - s_0)T_q)^{-1}\mathbf{e}_1, \tag{2}$$

where $\mathbf{e}_1$ is the first unit vector. The moments can also be found from this expression:

$$\mathbf{c}^T\hat{A}^k\mathbf{r} = \mathbf{c}^T\mathbf{r} \cdot \mathbf{e}_1^T T_q^k \mathbf{e}_1.$$

A proof of these facts can be found in the original paper [9].

Every iteration leads to the preservation of two extra moments. This makes PVL a very powerful and efficient algorithm. Unfortunately, there are also disadvantages associated with the algorithm. For example, it is known that PVL does not always preserve stability. The reason for this is that PVL is based on a two-sided Lanczos algorithm, and uses non-orthogonal or skew projections. The problem has been studied by several authors; in [9], it is suggested to simply delete the poles which have a positive real part. However, such "remedies" are not based upon theory, and should probably be avoided. In the case of symmetric matrices, the problem can be resolved by using a one-sided Lanczos process, which would preserve stability.

Another problem associated with PVL is that the inner products $\mathbf{w}_{n+1}^T\mathbf{v}_{n+1}$ in the bi-orthogonal sequence may be zero or near to zero, in which case the Lanczos algorithm breaks down. To avoid this, a look-ahead version of the Lanczos process has been suggested in [10].

The PVL method has been very successful since 1993, and many new developments are based upon it. We mention the *matrix PVL* method, published by the inventors of PVL [10]. In [1] a more extensive version of MPVL with look-ahead and deflation is described. Another method presented by Freund and Feldmann is *SymPVL* [12–14], which is an efficient version of PVL for the case of symmetric matrices. Te method cures the stability problem observed for PVL. A similar (and earlier) development is SyPVL [11] The main idea of all these methods is to make use of the fact that the matrix is symmetric, so that it can be decomposed using a Cholesky decomposition. This then automatically leads to stability of the associated approximate models.

Another nice development worth mentioning is the *two-step Lanczos algorithm*. It splits the problem of reducing the original system into two separate phases. First a Lanczos process is performed using a Krylov subspace based upon the matrix itself, rather than its inverse. Clearly, this is much more efficient, and so it is easy to perform many steps of this procedure. This then leads to a reduction of the original problem

to a system of size, say, a few thousand. In the second phase, the 'ordinary' Lanczos procedure is used to reduce the problem much further, now using the inverse of the coefficient matrix. For more details, see [35, 36].

3.3 Arnoldi and PRIMA Method

The Lanczos process is most suitable for symmetric matrices (see Chapter 2), but for general matrices the Arnoldi method is a better choice. It can also be used as the basis for a model order reduction method like PVL. Similar to PVL, one can define an expansion point s_0, and work with the shift-and-invert transfer function:

$$H(s) = C^T(sI_n - A)^{-1}B = C^T(I_n + (s - s_0)\hat{A})^{-1}R,$$

with

$$\hat{A} = (s_0 I_n - A)^{-1},$$

and

$$R = (s_0 I_n - A)^{-1}B.$$

Then, in the Arnoldi process, a Krylov space associated with the matrices $\hat{A}$ and R is generated:

$$\mathcal{K}_q(R, \hat{A}) = span\{R, \hat{A}R, \ldots, \hat{A}^q R\}.$$

The main differences with PVL are that only one Krylov space is generated, namely with the (block) *Arnoldi process* [31], and that the projections are performed with orthogonal operators.

The expansion point in the above can be chosen either real or complex, leading to different approximations of the poles of the system. A real shift may be favorable over a complex shift, as the convergence of the reduced transfer function towards the original transfer function is more global.

A very important new development was published in 1998, with a method now known as *PRIMA* [26]. Up till then, the methods developed suffered from non-passivity. Odabasioglu and Celik realized that the Arnoldi method had the potential to resolve these problems with passivity. The PRIMA method, in full passive reduced-order interconnect macromodeling algorithm, builds upon the same Krylov space as in the Arnoldi method and PVL, using the Arnoldi method to generate an orthogonal basis for the Krylov space. The fundamental difference with preceding methods is, however, that the projection of the matrices is done explicitly. This is in contrast with PVL and Arnoldi, where the tridiagonal or the Hessenberg matrix is used for this purpose. In other words, the following matrix is formed:

$$A_q = V_q^T A V_q,$$

where V_q is the matrix containing an orthonormal basis for the Krylov space.

Although more expensive, the explicit projection onto the Krylov space has strong advantages. It makes PRIMA more accurate than the Arnoldi method and it ensures preservation of stability and passivity. As such, it was the first method to achieve this in a provable way. A slight disadvantage of the method, as compared to PVL, is that only one moment per iteration is preserved additionally. This is only a minor disadvantage, if one realizes that PVL requires iterating both with A and its transpose.

3.4 Laguerre Methods

The search for provable passive model order reduction techniques continued after the publication of PRIMA. A new development was the construction of approximations using the framework of Laguerre functions, as proposed in [23,24]. In these methods, the transfer function is not shifted-and-inverted, as is the case in the PVL and Arnoldi methods. Instead, it is expanded in terms of Laguerre functions that are defined as

$$\phi_k^\alpha(t) \equiv \sqrt{2\alpha}\exp(-\alpha t)\updownarrow_k(2\alpha t),$$

where α is a positive scaling parameter, and $\updownarrow_k(t)$ is the Laguerre polynomial

$$\updownarrow_k(t) \equiv \frac{\exp(t)}{k!}\frac{d^k}{dt^k}(\exp(-t)t^k).$$

The Laplace transform of $\phi_k^\alpha(t)$ is

$$\Phi_k^\alpha(s) = \frac{\sqrt{2\alpha}}{s+\alpha}\left(\frac{s-\alpha}{s+\alpha}\right)^k.$$

Furthermore, it can be shown (see [20]) that the Laguerre expansion of the transfer function is

$$H(s) = \frac{2\alpha}{s+\alpha}C^T\sum_{k=0}^{\infty}\left((\alpha I_n - A)^{-1}(-\alpha I_n - A)\right)^k(\alpha I_n - A)^{-1}B\left(\frac{s-\alpha}{s+\alpha}\right)^k.$$

Clearly, this expansion gives rise to a Krylov space again. The number of linear systems that needs to be solved is equivalent to that in PRIMA, so the method is computationally competitive.

The algorithm presented in [23] then reads:

1. select a value for α and q
2. solve $(\alpha I_n - A)R_1 = B$
3. for k=2,...,q, solve $(\alpha I_n - A)R_k = (-\alpha I_n - A)R_{k-1}$
4. define $R = [R_1, ..., R_q]$ and calculate the SVD of R: $R = V\Sigma W^T$
5. $\tilde{A} = V^T A V$
6. $\tilde{C} = V^T C V$
7. $\tilde{B} = V^T B V$

In [23] it is argued that the best choice for α is to take it equal to $2\pi f_{\max}$, where $f_{\max}$ is the maximum frequency for which the reduced order model is to be valid.

As can be seen from the algorithm, the Krylov space is built without intermediate orthogonalisation. Instead, a singular value decomposition (SVD) is performed after the process has terminated. Consequently, V is an orthonormal basis of R. SVD is known to be a very stable and accurate way to perform this orthogonalisation, on the other hand it is computationally expensive. There are good alternatives, such as the QR method or Modified Gram-Schmidt. In [21], an alternative to the Laguerre-SVD

method is presented that makes use of intermediate orthogonalisation, and has been shown to have certain advantages over using an SVD.

Just as in the PRIMA method, the Laguerre-based methods make use of explicit projection of the system matrices. Consequently, these methods preserve stability and passivity. Since α is a real number, the matrices in the Laguerre algorithm remain real during projection, thereby making it suitable for circuit synthesis (see [21, 24]).

3.5 Truncation Methods

As mentioned before, model order reduction has its roots in the area of systems and control theory. Within this area, methods have been developed that differ considerably from the Krylov based methods as discussed in subsections 3.1-3.5. The basic idea is to truncate the dynamical system studied at some point. To illustrate how it works, consider again the linear dynamical system (1):

$$\begin{aligned} \frac{d\mathbf{x}}{dt} &= A\mathbf{x} + B\mathbf{u}, \\ \mathbf{y} &= C^T\mathbf{x} + D\mathbf{u}. \end{aligned}$$

Applying a state space transformation

$$T\tilde{\mathbf{x}} = \mathbf{x},$$

does not affect the input-output behavior of the system. This transformation could be chosen to be based on the eigenvalue decomposition of the matrix A:

$$AT = T\Lambda.$$

When T is non-singular, $T^{-1}AT$ is a diagonal matrix consisting of the eigenvalues of A, and we could use an ordering such that the eigenvalues on the diagonal occur in order of decreasing magnitude. The system can then be truncated by restricting the matrix T to the dominant eigenvalues. This process is termed *modal truncation*.

Another truncation method is that of balanced truncation, usually known as *Truncated Balanced Realization* (TBR). This method is based upon the observation that only the largest singular values of a system are important. As there is a very good reference to this method, containing all details, we will only summarize the main concepts. The reader interested in the details of the method is referred to the book by Antoulas [2].

The controllability Gramian and the observability Gramian associated to the linear time-invariant system (A, B, C, D) are defined as follows:

$$P = \int_0^\infty e^{At} B B^* e^{A^* t} dt, \tag{3a}$$

and

$$Q = \int_0^\infty e^{At} C^* C e^{A^* t} dt, \tag{3b}$$

respectively.

The matrices P and Q are the unique solutions of two Lyapunov equations:

$$AP + PA^* + BB^* = 0, \tag{4a}$$

$$A^*Q + QA + C^*C = 0. \tag{4b}$$

The Lyapunov equations for the Gramians arise from a stability assumption of A. Stability in the matrix A implies that the infinite integral defined in (3) is bounded. Finding the solution of the Lyapunov equation is quite expensive. There are direct ways and iterative ways to do this. One of the interesting iterative methods to find a solution is vector ADI [8,33]. New developments are also in the work of Benner [3].

After finding the Gramians, we look for a state space transformation which balances the system. A system is called *balanced* if $P = Q = \Sigma = \text{diag}(\sigma_i)$. The transformation will be applied to the system as follows:

$$\begin{aligned} A' &= T^{-1}AT \\ B' &= T^{-1}B \\ C' &= CT \\ D' &= D. \end{aligned}$$

This transformation also yields transformed Gramians:

$$\begin{aligned} P' &= T^{-1}PT^{-*} \\ Q' &= T^*QT. \end{aligned}$$

Because P and Q are positive definite, a Cholesky factorization of P can be calculated, $P = R^{\mathrm{T}}R$, with $R \in \mathbb{R}^{n\times n}$. Then the Hankel singular values are derived as the singular values of the matrix RQR^{T}, which are equal to the square root of the eigenvalues of RQR^{T} and QP. So:

$$RQR^{\mathrm{T}} = U^{\mathrm{T}}\Sigma^2 U. \tag{5}$$

Then the transformation $T \in R^{n\times n}$ is defined as:

$$T = R^{\mathrm{T}}U^{\mathrm{T}}\Sigma^{-1/2}. \tag{6}$$

The inverse of this matrix is:

$$T^{-1} = \Sigma^{1/2}UR^{-1}. \tag{7}$$

This procedure is called balancing. It can be shown that T indeed balances the system:

$$\begin{aligned} Q' &= T^{\mathrm{T}}QT = \Sigma^{-1/2}URQR^{\mathrm{T}}U^{\mathrm{T}}\Sigma^{-1/2} = \Sigma^{-1/2}\Sigma^2\Sigma^{-1/2} = \Sigma \\ P' &= T^{-1}PT^{-\mathrm{T}} = \Sigma^{1/2}UR^{-\mathrm{T}}PR^{-1}U^{\mathrm{T}}\Sigma^{1/2} = \Sigma^{1/2}\Sigma^{1/2} = \Sigma. \end{aligned}$$

Since a transformation was defined which transforms the system according to the Hankel singular values [2], now very easily a truncation can be defined.

Σ can be partitioned:

$$\Sigma = \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix}, \tag{8}$$

where Σ_1 contains the largest Hankel singular values. This is the main advantage of this method, since now we can manually choose an appropriate value of the size of the reduction, instead of guessing one.

A', B' and C' can be partitioned in conformance with Σ:

$$A' = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \tag{9}$$

$$B' = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix} \tag{10}$$

$$C' = \begin{pmatrix} C_1 & C_2 \end{pmatrix}. \tag{11}$$

The reduced model is then based on A_{11}, B_1 and C_1:

$$\dot{\tilde{\mathbf{x}}} = A_{11}\tilde{\mathbf{x}} + B_1\mathbf{u}$$
$$\mathbf{y} = C_1\tilde{\mathbf{x}}.$$

It is sometimes proposed to apply Balanced Truncation-like methods as a second reduction step, after having applied a Krylov method. This can be advantageous in some cases, and has also been done by several authors.

A remark should be made on solving the Lyapunov equation. These equations are normally solved by first calculating a Schur decomposition for the matrix A. Therefore, finding the solution of the Lyapunov is quite expensive, the number of operations is at least $O(n^3)$, where n is the size of the original model. Hence, it is only feasible for small systems. Furthermore, because we arrived at this point using the inverse of an ill-conditioned matrix we have to be careful. B can have very large entries, which will introduce tremendous errors in solving the Lyapunov equation. Dividing both equations by the square of the norm of B spreads the malice a bit, which makes finding a solution worthwhile. In recent years, however, quite a lot of progress has been made on solving larger Lyapunov equations. We refer to the work of Benner [3].

Another remark: since the matrices are projected by a similarity transform, preservation of passivity is not guaranteed in this method. In [27] a Balanced Truncation method is presented which is provably passive. Here also Poor Man's TBR [28] should be mentioned as a fruitful approach to implement TBR is a more efficient way. We refer to a later chapter in this book for more information on this topic.

3.6 Optimal Hankel Norm Reduction

Closely related to Balanced Truncation is Optimal Hankel Norm reduction [18]. In the Balanced Truncation norm it was not clear whether the truncated system of size say k was an optimal approximation of this size. It is seen that this optimality can be calculated and reached given a specific norm, the Hankel norm.

To define the Hankel norm we first have to define the Hankel operator $\mathcal{H}$:

$$\mathcal{H} : u \to y = \int_{-\infty}^{0} h(t-\tau)u(\tau), \tag{12}$$

where $h(t)$ is the impulse response in time domain: $h(t) = C\exp(At)B$ for $t > 0$. This operator considers the past input, the energy that was put into the system before $t = 0$, in order to reach this state. The amount of energy to reach a state tells something about the controllability of that state. If, after $t = 0$, no energy is put into the system and the system is stable, then the corresponding output will be bounded as well. The energy that comes out of a state, gives information about the observability of a state. The observability and controllability Gramians were defined in (3).

Therefore, the maximal gain of this Hankel operator can be calculated:

$$\|\Sigma\|_H = \sup_{u\in\mathcal{L}_2(-\infty,0]} \frac{\|y\|_2}{\|u\|_2}. \tag{13}$$

This norm is called the Hankel norm. Since it can be proved that $\|\Sigma\|_H = \lambda_{max}^{1/2}(PQ) = \sigma_1$, the Hankel norm is nothing but the largest Hankel singular value of the system.

There exists a transfer function and a corresponding linear dynamical system which minimizes this norm. In [18] an algorithm is given which explicitly generates this optimal approximation in the Hankel-norm. The algorithm is based on a balanced realization.

3.7 Selective Node Elimination

Krylov based methods build up the reduced order model by iterating, every iteration leading to a larger size of the model. Hence, the first few iterations yield extremely small models that will not be very accurate in general, and only when a sufficient number of iterations has been performed, the approximate model will be sufficiently accurate. Hence, characteristic for such methods is that the space in which approximations are being sought is gradually built up. An alternative would be to start 'at the other end', in other words, start with the original model, and reduce in it every iteration, until we obtain a model that is small and yet has sufficient accuracy. This is the basic idea behind a method termed *selective node elimination*. Although it can, in principle, be applied in many situations, it has been described in literature only for the reduction of electronic circuits. Therefore, we limit the discussion in this section to that application.

Reduction of a circuit can be done by explicitly removing components and nodes from the circuit. If a node in a circuit is removed, the behaviour of the circuit can be preserved by adjusting the components around this node. Recall that the components connected to the node that is removed, are also removed. The value of the remaining components surrounding this node must be changed to preserve the behavior of the circuit. For circuits with only resistors this elimination can be done exactly.

We explain the idea for an RC circuit. For this circuit we have the following circuit equation $Y(s)\mathbf{v} = (G + sC)\mathbf{v} = \mathbf{J}$. The vector $\mathbf{v}$ here consists of the node voltages, $\mathbf{J}$ is some input current term. Suppose the n-th node is eliminated. Then, we partition the matrix such that the (n, n) entry forms one part:

$$\begin{bmatrix} \tilde{Y} & \mathbf{y} \\ \mathbf{y}^{\mathrm{T}} & \gamma_n + s\chi_n \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{v}} \\ v_n \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 \\ j_n \end{bmatrix}.$$

Then the variable v_n is eliminated, which leads to:

$$(\tilde{Y} - E)\hat{\mathbf{v}} = (\mathbf{J}_1 - \mathbf{F}), \tag{14}$$

with

$$E_{ij} = \frac{y_i y_j}{\gamma_n + s\chi_n} = \frac{(g_{in} + sc_{in})(g_{jn} + sc_{jn})}{\gamma_n + s\chi_n} \tag{15a}$$

$$F_i = \frac{y_i}{\gamma_n + s\chi_n} j_n = \frac{g_{in} + sc_{in}}{\gamma_n + s\chi_n} j_n. \tag{15b}$$

If node n is not a terminal node, j_n is equal to 0 and therefore $\mathbf{F}{=}\mathbf{0}$ for all i. We see that the elimination can also be written in matrix notation. Hence, this approach is analogous to solving the system by Gaussian elimination. This approach can be used to solve PDE's in an efficient way.

After the elimination process the matrix is not in the form $\mathbf{G}{+}s\mathbf{C}$ anymore, but is a fraction of polynomials in s. To get an RC-circuit representation an approximation is needed. Given the approximation method that is applied, removing one node leads to a larger error than removing the other.

Many others have investigated methods which are strongly related to the approach described here, for instance a more symbolic approach. The strong attributes of the methods described above is that an RC circuit is the direct result. The error made with the reduction is controllable, but can be rather large. A disadvantage is that reducing an RLC-circuit in this way is more difficult and it is hard to get an RLC-circuit back after reduction.

3.8 Proper Orthogonal Decomposition

Apart from Krylov subspace methods and Truncation methods, there is Proper Orthogonal Decomposition (POD), also known as Karhunen-Loeve decomposition. This method is developed within the area of Computational Fluid Dynamics and nowadays used frequently in many CFD problems. The method is so common there, that it should at least be mentioned here as an option to reduce models derived in an electronic setting. The strong point of POD is that it can be applied to non-linear partial differential equations and is at the moment state-of-the-art for many of such problems.

The idea underlying this method is that the time response of a system given a certain input, contains the essential behavior of the system. The most important

aspects of this output in time are retrieved to describe the system. Therefore, the set of outputs serves as a starting-point for POD. The outputs, which are called 'snapshots', must be given or else be computed first.

A snapshot consists of a column vector describing the state at a certain moment. Let $W \in \mathbb{R}^{N\times K}$ be the matrix consisting of the snapshots. N is the number of snapshots, K is the number of elements in every snapshot, say the number of state variables. Usually we have that $N < K$.

Let $\mathcal{X}$ be a separable Hilbert space with inner product $(.,.)$, and with an orthonormal basis $\{\varphi_i\}_{i\in I}$. Then, any element $T(x,t) \in \mathcal{X}$ can be written as:

$$T(x,t) = \sum_i a_i(t)\varphi_i(x) = \sum_i (T(x,t),\varphi_i(x))\varphi_i(x). \tag{16}$$

The time dependent coefficients a_i are called Fourier coefficients. We are looking for an orthonormal basis $\{\varphi_i\}_{i\in I}$ such that the averages of the Fourier-coefficients are ordered:

$$\langle a_1^2(t)\rangle \geq \langle a_2^2(t)\rangle \geq \ldots, \tag{17}$$

where $\langle.\rangle$ is an averaging operator. In many practical applications the first few elements represent 99% of the content. Incorporating these elements in the approximation gives a good approximation. The *misfit*, the part to which the remaining elements contribute to, is small.

It can be shown that this basis can be found in the first eigenvectors of this operator:

$$C = \langle (T(t),\varphi)T(t)\rangle. \tag{18}$$

In case we consider a finite dimensional problem, in a discrete and finite set of time points, this definition of C comes down to:

$$C = \frac{1}{N}WW^{\mathrm{T}}. \tag{19}$$

Because C is self-adjoint, the eigenvectors are real and can be ordered, such that:

$$\lambda_1 \geq \lambda_2 \geq \ldots \tag{20}$$

A basis consisting of the first, say q eigenvectors of this matrix form the optimal basis for POD of size q.

This leads to the following POD algorithm:

1. Input: the data in the matrix W consisting of the snapshots.
2. Define the correlation matrix C as:

$$C = \frac{1}{N}WW^{\mathrm{T}}.$$

3. Compute the eigenvalue decomposition $C\Phi = \Phi\Lambda$.
4. Output: The basis to project the system on, Φ.

This algorithm contains an eigenvalue problem of size $K \times K$, which can be computationally expensive. Therefore the 'method of snapshots' is designed. The reason underlying this different approach is that the eigenvalues of C are the same as the eigenvalues of $K = \frac{1}{N} W^{\mathrm{T}} W$ but K is smaller, namely $N \times N$.

Let the eigenvalues of K be Ψ_i, then the eigenvectors of C are defined as:

$$\varphi_i = \frac{1}{\|W_{snap}\Psi_i\|} W_{snap}\Psi_i. \tag{21}$$

However, also the singular value decomposition of the snapshot matrix W is a straightforward way to obtain the eigenvectors and eigenvalues of C.

$$W = \Phi\Sigma\Psi^{\mathrm{T}}, \tag{22}$$

$$C = \frac{1}{N} W W^{\mathrm{T}} = \frac{1}{N} \Phi\Sigma\Psi^{\mathrm{T}}\Psi\Sigma\Phi^{\mathrm{T}} = \frac{1}{N} \Phi\Sigma^2\Phi^{\mathrm{T}}. \tag{23}$$

The eigenvectors of C are in Φ:

$$C\Phi = \frac{1}{N} \Phi\Sigma^2\Phi^{\mathrm{T}}\Phi = \Phi\frac{1}{N}\Sigma^2. \tag{24}$$

From which it can be seen that the eigenvalues of C are $\frac{1}{N}\Sigma^2$.

Once the optimal orthonormal basis is found, the system is projected onto it. For this, we will focus on the following formulation of a possibly non-linear model:

$$\begin{aligned} C(\mathbf{x})\frac{d}{dt}\mathbf{x} &= \mathbf{f}(\mathbf{x}, \mathbf{u}) \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}, \mathbf{u}). \end{aligned}$$

$\mathbf{x}$ consists of a part in the space spanned by this basis and a residual:

$$\mathbf{x} = \hat{\mathbf{x}} + \mathbf{r}, \tag{25}$$

where $\hat{\mathbf{x}} = \sum_{k=1}^{Q} a_k(t)\mathbf{w}_k$. When $\hat{\mathbf{x}}$ is taken as state space in (25) an error is made:

$$C(\hat{\mathbf{x}})\frac{d}{dt}\hat{\mathbf{x}} - \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) = \rho \neq 0. \tag{26}$$

This error is forced to be perpendicular to the basis W. Forcing this defines the projection fully. In the following derivation we use that $\frac{d}{dt}\hat{\mathbf{x}} = \sum_{k=1}^{Q} \frac{d}{dt} a_k(t)\mathbf{w}_k$:

$$\begin{aligned} 0 = \left(C(\hat{\mathbf{x}})\frac{d}{dt}\hat{\mathbf{x}} - \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}), \mathbf{w}_k \right) &= \left(C(\hat{\mathbf{x}}) \sum_{k=1}^{Q} \frac{d}{dt} a_k(t)\mathbf{w}_k - \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}), \mathbf{w}_k \right) \\ &= \sum_{k=1}^{Q} \frac{d}{dt} a_k(t) \left(C(\hat{\mathbf{x}})\mathbf{w}_k, \mathbf{w}_k \right) - \left(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}), \mathbf{w}_k \right), \end{aligned} \tag{27}$$

for $j = 1, \ldots, Q$. Therefore the reduced model of order Q can be formulated as:

$$A(\mathbf{a})\frac{d}{dt}\mathbf{a} = \mathbf{g}(\mathbf{a}, \mathbf{u})$$
$$\mathbf{y} = \mathbf{h}\left(\sum_{k=1}^{Q} a_k \mathbf{w}_k, \mathbf{u}\right),$$

where:

$$\begin{aligned} A_{ij} &= \left(C(\textstyle\sum_{k=1}^{Q} a_k(t)\mathbf{w}_k)\mathbf{w}_i, \mathbf{w}_j\right) \\ \mathbf{a}_j &= a_j(t) \\ \mathbf{g}(\mathbf{a}(t), \mathbf{u}(t)) &= \left(\mathbf{f}(\textstyle\sum_{k=1}^{Q} a_k(t)\mathbf{w}_k, \mathbf{u}(t)), \mathbf{w}_j\right) \end{aligned}$$

Obviously, if the time domain output of a system has yet to be calculated, this method is far too expensive. Fortunately, the much cheaper to obtain frequency response can be used. Consider therefore the following linear system:

$$(G + j\omega C)\mathbf{x} = B\mathbf{u}$$
$$\mathbf{y} = L^{\mathrm{T}}\mathbf{x}.$$

Calculate a set of frequency states, for certain choices of ω:

$$\mathbf{x}_{\omega_j} = [j\omega_j C + G]^{-1} B, \tag{29}$$

where $\mathbf{x}_{\omega_j} \in \mathbb{C}^{n\times 1}$. We can take the real and imaginary part, or linear combinations of both, for the POD process. We immediately see that the correlation matrix is an approximation of the controllability Gramian:

$$K = \frac{1}{M}\sum_{j=1}^{M} [j\omega_j C + G]^{-1} B B^* [-j\omega_j C^* + G^*]^{-1}. \tag{30}$$

This approach solves the problem of chosing which time-simulation is the most appropriate.

3.9 Other Methods

In the foregoing sections, we have reviewed a number of the most important methods for model order reduction. The discussion is certainly not exhaustive, alternative methods have been published. For example, we have not mentioned the method of *vector fitting*. This method builds rational approximations of the transfer function in a very clever and efficient way, and can be used to adaptively build reduced order models. The chapter by Deschrijver and Dhaene contains an account of recent developments in this area.

As model order reduction is a very active area of research, progress in this very active area may lead to an entirely new class of methods. The development of such

new methods is often sparked by an industrial need. For example, right now there is a demand for reducing problems in the electronics industry that contain many inputs and outputs. It has already become clear that current methods cannot cope with such problems, as the Krylov spaces very quickly become inhibitively large, even after a few iterations. Hence, new ways of constructing reduced order models must be developed.

References

1. J.I. Aliaga, D.L. Boley, R.W. Freund and V. Hernández. A Lanczos-type method for multiple starting vectors. *Math. Comp.*, 69(232):1577-1601, May 1998.
2. A.C. Antoulas book. *Approximation of Large-Scale Dynamical Systems*. SIAM series on Advances in Design and Control, 2005.
3. P. Benner, V. Mehrmann and D.C. Sorensen. *Dimension Reduction of Large-Scale Systems*. Lecture Notes in Computational Science and Engineering, vol. 45, Springer-Verlag, June 2005.
4. A. Brandt. Multilevel adaptive solutions to boundary value problems. *Math. Comp.*, 31:333-390, 1977.
5. W.L. Briggs, Van Emden Henson and S.F. McCormick. *A multigrid tutorial*. SIAM, 2000.
6. *BSIM3 and BSIM4 Compact MOSFET Model Summary*. Online: http://www-device.eecs.berkeley.edu/ 3/get.html.
7. E. Chiprout and M.S. Nakhla. *Asymptotic Waveform Evaluation and moment matching for interconnect analysis*. Kluwer Academic Publishers, 1994.
8. N.S. Ellner and E.L. Wachspress. Alternating direction implicit iteration for systems with complex spectra. *SIAM J. Numer. Anal.*, 28(3):859.870, June 1991.
9. P. Feldmann and R. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Computer-Aided Design*, 14:137-158, 1993.
10. P. Feldmann and R. Freund. Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm. *Proc. 32nd ACM/IEEE Design Automation Conf.*, June 1995.
11. R.W. Freund and P. Feldmann. *Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm*. Numerical Analysis Manuscript 96-13, Bell Laboratories, Murray Hill, N.J., May 1996.
12. R.W. Freund and P. Feldmann. Interconnect-Delay Computation and Signal-Integrity Verification Using the SyMPVL Algorithm. Proc. 1997 European Conf. Circuit Theory and Design, 408-413, 1997.
13. R.W. Freund and P. Feldmann. The SyMPVL algorithm and its application to interconnect simulation. *Proc. 1997 Int. Conf. Simulation of Semiconductor Processes and Devices*, 113-116, 1997.
14. R.W. Freund and P. Feldmann. Reduced-order modeling of large linear passive multiterminal circuits using Matrix-Padé approximation. *Proc. DATE Conf. 1998*, 530-537, 1998.
15. G. Gildenblat, X. Li, W. Wu, H. Wang, A. Jha, R. van Langevelde, G.D.J. Smit, A.J. Scholten and D.B.M. Klaassen. PSP: An Advanced Surface-Potential-Based MOSFET Model for Circuit Simulation. *IEEE Trans. Electron Dev.*, 53(9):1979-1993, September 2006.
16. E.J. Grimme. *Krylov Projection Methods for model reduction*. PhD thesis, Univ. Illinois, Urbana-Champaign, 1997.

17. G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 3rd edition, 1996.
18. K. Glover. Optimal Hankel-norm approximations of linear multivariable systems and their l_∞-error bounds. *Int. J. Control*, 39(6):115-193, 1984.
19. H.C. de Graaff and F.M. Klaassen. *Compact Transistor Modelling for Circuit Design*. Springer-Verlag, Wien, New York, 1990.
20. Heres2001
21. P.J. Heres. *Robust and efficient Krylov subspace methods for Model Order Reduction*. PhD Thesis, TU Eindhoven, The Netherlands, 2005.
22. M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for the solution of linear systems. *J. Res. Res. Natl. Bur. Stand.*, 49:409-436, 1952.
23. L. Knockaert and D. De Zutter. Passive Reduced Order Multiport Modeling: The Padé-Arnoldi-SVD Connection. *Int. J. Electron. Commun. (AEU)*, 53:254-260, 1999.
24. L. Knockaert and D. De Zutter. Laguerre-SVD Reduced-Order Modelling. *IEEE Trans. Microwave Theory and Techn.*, 48(9):1469-1475, September 2000.
25. J.A. Meijerink and H.A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148-162, 1977.
26. A. Odabasioglu and M. Celik. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design*, 17(8):645-654, August 1998.
27. J.R. Phillips, L. Daniel, and L.M. Silveira. Guaranteed passive balancing transformations for model order reduction. *Proc. 39th cConf. Design Automation*, 52-57, 2002.
28. J.R. Phillips and L.M. Silveira. Poor Man's TBR: A simple model reduction scheme. *IEEE. Trans. Comp. Aided Design ICS*, 24(1):43-55, January 2005.
29. L.T. Pillage and R.A. Rohrer. Asymptotic Waveform Evaluation for Timing Analysis. *IEEE Trans. Computer-Aided Design Int. Circ. and Syst.*, 9(4): 352-366, April 1990.
30. A. Quarteroni and A. Veneziani. Analysis of a geometrical multiscale model based on the coupling of PDE's and ODE's for blood flow simulations. *SIAM J. on MMS*, 1(2): 173-195, 2003.
31. Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 2003.
32. W.H.A. Schilders and E.J.W. ter Maten. *Numerical methods in electromagnetics*. Handbook of Numerical Analysis, volume XIII, Elsevier, 2005.
33. G. Starke. Optimal alternating direction implicit parameters for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 28(5):1431-1445, October 1991.
34. H.A. van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 13(2):631-644, 1992.
35. T. Wittig. *Zur Reduktion der Modellordnung in elektromagnetischen Feldsimulationen*. PhD thesis, TU Darmstadt, 2003.
36. T. Wittig, I. Munteanu, R. Schuhmann, and T. Weiland. Two-step Lanczos algorithm for model order reduction. *IEEE Trans. Magnetics*, 38(2):673-676, March 2001.

http://www.springer.com/978-3-540-78840-9

Model Order Reduction: Theory, Research Aspects and Applications

Schilders, W.H.; van der Vorst, H.A.; Rommes, J. (Eds.)

2008, XI, 471 p., Hardcover
ISBN: 978-3-540-78840-9