CHEMNITZ UNIVERSITY OF TECHNOLOGY

Department of Mathematics

Mathematics in Industry and Technology

# Diploma Thesis

## Matrix decomposition based approaches for model order reduction of linear systems with a large number of terminals

André Schneider

Chemnitz, 6th April 2008

**Supervisor:** Prof. Dr. Peter Benner

**Second referee:** Dr. Georg Denk

**Schneider, André**
Matrix decomposition based approaches for model order reduction of linear systems with a large number of terminals
Diploma Thesis, Department of Mathematics
Chemnitz University of Technology, April 2008

## Acknowledgement

I would like to thank Prof. Dr. Peter Benner, who always believed in me, for his assistance and his pieces of advice.
In addition to this I would like to thank my mother Simone Schneider and my family for everything, Dr. Georg Denk for his advice, Dr. Uwe Feldmann, Corinna Schnerr for all the support over all these wonderful years, Tobias Günnel for his sleepless nights while proofreading this work, Judith Will, Manuel Naumann, Marcus Meyer, Susan Dürigen and Matthias Schieck. You all made it possible for me to finish my studies.
Thanks a lot!

## Danksagung

Ich möchte Prof. Dr. Peter Benner, der immer an mich geglaubt hat, für seine Unterstützung und seine Ratschläge danken.
Zusätzlich möchte ich mich bei meiner Mutter Simone Schneider und meiner ganzen Familie für alles bedanken, bei Dr. Georg Denk für seinen Rat, bei Dr. Uwe Feldmann, bei Corinna Schnerr für all die Unterstützung die ganzen wunderschönen Jahre lang, bei Tobias Günnel für die unzähligen schlaflosen Nächte des Korrekturlesens, bei Judith Will, Manuel Naumann, Marcus Meyer, Susan Dürigen und bei Matthias Schieck. Ihr alle habt es mir ermöglicht, mein Studium zu beenden.
Vielen Dank!

## Task formulation

The task of the diploma thesis is to find out in which way different types of reduced-rank matrix approximations are practical for system and model order reduction methods, specially for linear systems resulting from subcircuits with a large number of terminals. Here several basic approaches will be based on the idea of the singular value decomposition (the so called SVD) and a couple of variations of this idea. One important aspect is that it is possible to estimate the error of the reduction by applying these methods. For the different types of the SVDMOR idea including error estimates source codes should be implemented. Many tests should be realised to find out which approaches and parameters are suitable.

# Contents

# 1 Introduction

Model order reduction (MOR) is a very recent but very important research area. Some topics referring to MOR were introduced about 40 years ago. Today it is a basic method of numerical simulation of mathematical models, especially in electrical engineering and in our case in the area of circuit simulation, see [FNG92, Sil94, Fre03, BMS05]. Industrial circuit simulators like SPICE (Simulation Program with Integrated Circuit Emphasis), and many in-house simulators like TITAN at Qimonda and Infineon, PowerSpice at IBM, Linx at Intel Corporation, and Pstar at NXP Semiconductors are using MOR methods for developing integrated circuits (ICs). In this area it is possible to profit deeply from new insights. Without ICs, nearly everything in daily life using electrical appliances would be unimaginable. Examples are easy to find:

- Personal computers as well as high performance computers need a powerful central processing unit (CPU). This is nothing else than an IC, developed with the help of model order reduction.
- Every car is full of sensors performing various tasks. They are not only used for the passengers' comfort like air conditioning, but more importantly for security systems like braking systems, the electronic stability program (ESP), and airbag systems.
- In cell phones there are very complex ICs with a high integration level. That makes it possible to build them smaller and much more efficient. Also the energy consumption is reduced because small ICs need less power.
- If one wants to store data one needs ICs in most cases. Primary applications therefore are digital cameras, video game consoles, cell phones, and other applications.
- In medical sciences there is enormous progress in targeted developments of efficient medical devices resulting in new methods of treatment which can save human lives.

As one can easily see, the numerous applications of ICs cause a huge interest in model order reduction.

The examples show that the development of new methods for producing smaller, faster, and more powerful ICs is very important for the evolution of our highly developed modern society. Since MOR is essential for developing more efficient ICs, MOR received a lot of attention in recent years.

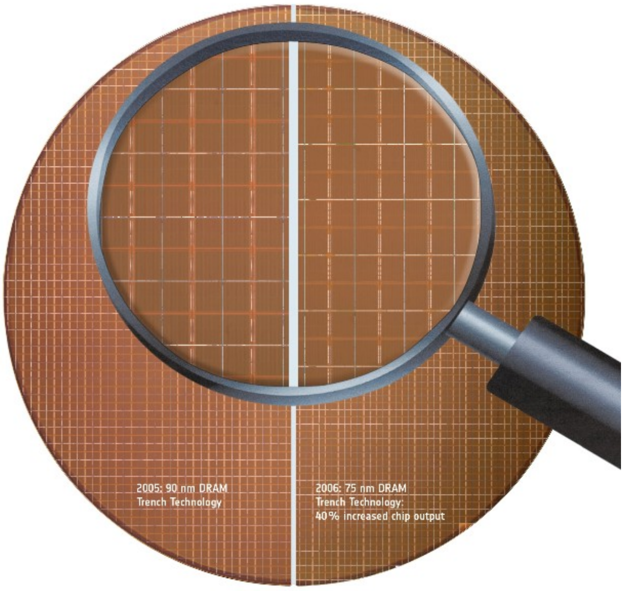

Figure 1.1: Wafer DDR2 90nm vs 75nm, source: Qimonda AG

Today there are new challenges in this area of research. Integrated circuits got significantly more complex over the last few years. Computer chip producers are constantly developing new technical methods to build smaller circuit devices, see Fig. 1.1, to increase the number of elements on these ICs dramatically while keeping the size of the IC constant, see Moore's Law in Fig. 1.2. The earliest patent for an IC, including five transistors, was filed in April 1949 by the German engineer Werner Jacobi. In the 1970s, large scaled integration (LSI) was reached with tens of thousands of transistors per chip. In 2006, the Intel Core 2 Extreme (Conroe XE) central processing unit was published with about 291 million transistors. Intel used a 65 nanometer chip manufacturing process. Today's ICs consist of several layers of circuit elements, which are connected by a complex wiring, see Fig. 1.3.

For data that is interesting for simulations, e. g. computing currents and voltages as a function of currents, voltages, time, frequency, temperature or other parameters, too much computing power would be needed since an equation exists for nearly every element of the circuit. Due to the vast amount of circuit elements, the resulting systems, so called Differential Algebraic Equations (DAEs), are of enormous size. It is possible to get up to $10^5$ equations for the simulation of just one subcircuit.

In addition, in some applications the structure of the parasitic linear subcircuits which

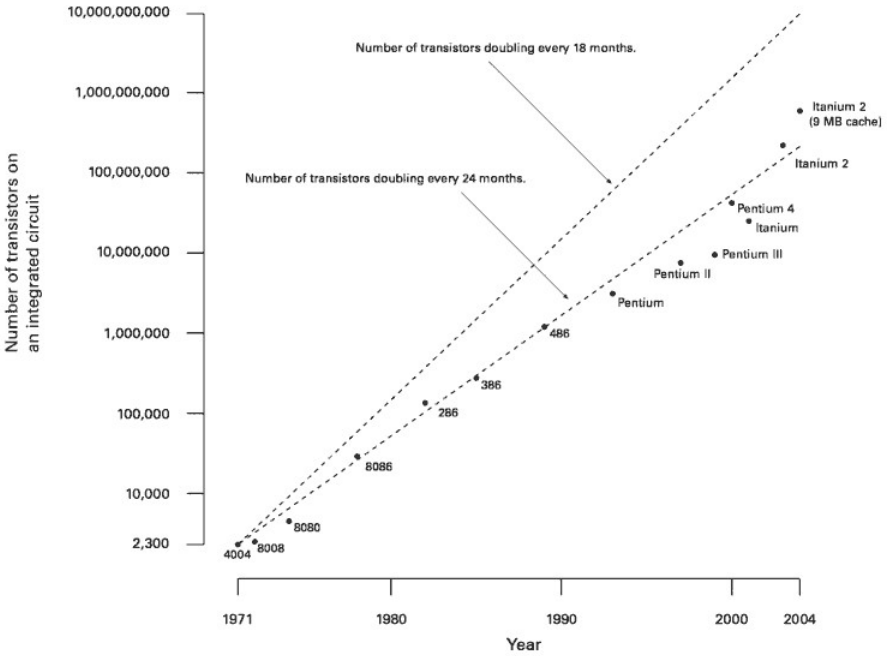

Figure 1.2: Moore's Law (2004), source: http://en.wikipedia.org/wiki/Moore's_law

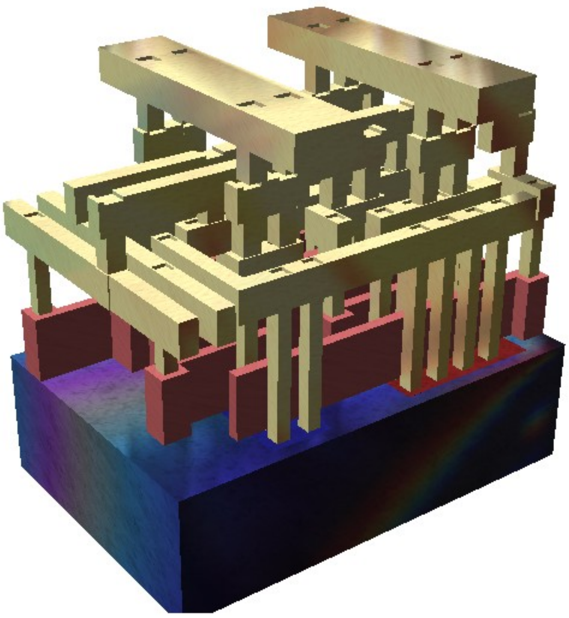

Figure 1.3: Silicon chip 3d, source: http://en.wikipedia.org/wiki/Image:Silicon_chip_3d.png

should be reduced has changed. Parasitic linear subcircuits are parts of the circuit arising in the design and containing linear elements with less or no influence on the result of the simulation. So far, the number of elements in these subcircuits was significantly larger than the number of connections to the whole circuit, the so called pins. This assumption is no longer valid in all cases. Circuits with a lot of elements need an extra power supply network, so called power grids. In clock distribution networks, the clock signal is distributed from a common point to all the elements that need it for synchronization. For simulating these circuits new methods are needed as well.

MOR is one of many numerical methods to facilitate the simulations. The basic idea behind MOR is to disregard the parasitic elements. MOR causes high expectations for future research since the competition for effective methods is rather large.

To provide a short overview, we want to briefly summarize some earlier work on this specific research area. But before that, the basic research in matrix computation should be mentioned. It was done by scientists like Gene Golub, together with William Kahan publisher of an algorithm for computing the singular value decomposition (SVD), and others. Without the results from this work, this thesis would be unimaginable. The book "Milestones in matrix computation. The selected works of Gene H. Golub." [CGO07] is a good reference.

- Motivation for the work about MOR for linear circuits with a large number of terminals was a paper of Peter Feldmann and Frank Liu in 2004 [FL04]. Here a method called SVDMOR is introduced which is explained in Sec. 3.1.1. In this paper SVDMOR is used as a part of a method called RecMOR, which works with the hierarchical decomposition of the matrix transfer function (2.13). Since SVDMOR is a more promising method than RecMOR, we want to follow the work about this approach as an efficient terminal reduction method.

- In 2006, Pu Liu, Sheldon X.-D. Tan, Boyuan Yan and Bruce McGaughy introduced an extended version [LTYM06], the so called ESVDMOR, which is based on SVDMOR. Improvements in accuracy and the mode of action has been achieved. A more detailed analysis was published in 2008 [LTYM08] by the same researchers.

- The papers of Martin Stoll [Sto08], James Baglama and Lothar Reichel [BR05] and Michiel E. Hochstenbach [Hoc01] are all dealing with matrix factorization approaches based on the SVD. We use this to explain the Truncated SVD method in Sec. 3.2.1, which we want to use in form of the MATLAB function `svds`, see Chap. 4.

- Michael W. Berry, Shakhina A. Pulatova and G. W. Stewart introduced in 2004 a paper about "Computing sparse reduced-rank approximations to sparse matrices" [BPS05] which is analyzed in Sec. 3.2.2. Here the sparse column row approximation

(SCRA) is explained and the authors show the differences between the SVD and the SCRA. Also, computational details of the algorithm and a MATLAB implementation are described.

This thesis is structured as follows. Chap. 2 introduces basic knowledge about MOR including the way to model a problem and some existing methods in MOR. In Chap. 3, ideas for reducing the number of input and output terminals based on matrix decomposition are mentioned. One approach, combining an existing method with a truncated singular value decomposition, is then picked up and provides the foundation for the succeeding Chap. 4. There, an algorithm for this method and some results are presented. The last chapter, Chap. 5, serves as a summary and provides an outlook on future work.

# 2 Background in MOR

This chapter provides an overview of the modeling of a MOR problem as well as different approaches and methods to solve it. First, it is explained how to derive the system of DAEs descibing a specific circuit, respectively its design. After that, we will present and explain some MOR methods based on Krylov subspace projection methods and balanced truncation.

## 2.1 The modeling part

Before one can start with any calculations addressing MOR, modeling the problem, that means the creation of problem describing equations, is necessary. The section is mostly based on [GFtM05]. The principle here is to connect compact elements of the circuit design via ideal nodes. Then one has to apply some kind of conservation rules. With the help of these rules it is possible to set up network equations using the information of the network topology and the characteristic equations of the network elements, which describe their physical behavior. As usual, the model is based on simplifying assumptions that have to be met. The result will in general be a differential-algebraic equation of the form

$$f(x,\dot{x},t) = 0 \tag{2.1}$$

with, like mentioned above in general, $\det(\frac{\partial f}{\partial \dot{x}}) \equiv 0$.

### 2.1.1 The Schmitt trigger

Let us look at a very simple but instructive example, the so called Schmitt Trigger, see Fig. 2.1. It is a very simple circuit which is used to transform analog into digital signals.

As one can see in Fig. 2.1, the circuit consists of:

- five linear resistors with conductances $G_1,\ldots,G_5$,
- one power supply voltage source $V_{DD}$,
- one input voltage source $V_{in}$,
- one linear capacitor $C_0$ (which causes the dynamic behavior),
- two bipolar npn-transistors (which cause the nonlinear characteristics).

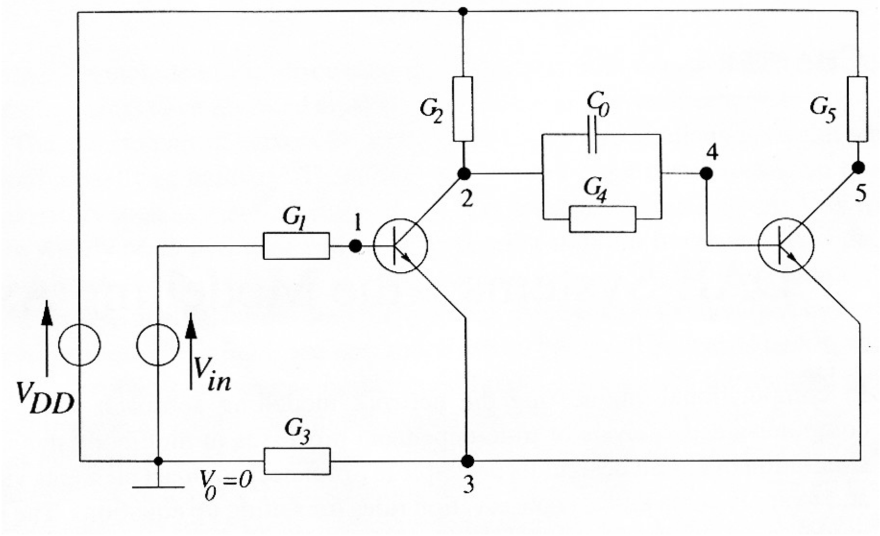

Figure 2.1: The Schmitt trigger circuit, source: [GFtM05].

The aim is now to find a mathematical model which determines the five node potentials $u_1, \dots, u_5$, i.e., current balances for all nodes (except ground). Using the modeling process called the nodal analysis as described in the following sections, we get the DAE

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & C_0 & 0 & -C_0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -C_0 & 0 & C_0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \dot{u}_3 \\ \dot{u}_4 \\ \dot{u}_5 \end{pmatrix} + \tag{2.2}$$

$$\begin{pmatrix} G_1 \cdot (u_1 - V_{in}) + g(u_1 - u_3) \\ G_2 \cdot (u_2 - V_{DD}) + G_4 \cdot (u_2 - u_4) + \alpha \cdot g(u_1 - u_3) \\ G_3 \cdot u_3 - (1+\alpha) \cdot g(u_1 - u_3) - (1+\alpha) \cdot g(u_4 - u_3) \\ G_4 \cdot (u_4 - u_2) + g(u_4 - u_3) \\ G_5 \cdot (u_5 - V_{DD}) + \alpha \cdot g(u_4 - u_3) \end{pmatrix} = 0.$$

The characteristic current $g(U_T)$ and the amplification factor $\alpha$ are described in (2.3). The process of performing a modified nodal analysis is described in Sec. 2.1.3.

### 2.1.2 Kirchhoff's laws

In the following, we want to introduce the basic terminology of electrical engineering that enables the reader to follow the course of this work. Therefore, some definitions are introduced. For more information, see e. g. [Ulb03, SBD$^+$05, Wol88, Wol86, Unb00].

**Definition 2.1 (Electrical network)** *An* electrical network *is any interconnection of electrical elements.*

**Definition 2.2 (Node)** *A* node *of a network is the point where the connection lines between the elements of the network meet. It is pictured as a point.*

**Definition 2.3 (Branch)** *A* branch *is a direct current path between two nodes of a network. It consists of the part from one node to the circuit element and the part from the element to the other node. It is pictured as a line.*

**Definition 2.4 (Loop or mesh)** *A* loop *or* mesh *is a closed connection within the network. It consists of an arbitrary number of connected branches.*

**Definition 2.5 (Pin or terminal)** *A* pin *or* terminal *is another name for a lead. Pins are the connections of an element or the nodes of a subcircuit which connect to the circuit itself. To make a long story short, terminals are the beginning or the end of a branch.*

**Remark 2.1** *The whole network we use is supposed to be electrically ideal, that means for example the connections between the elements as well as the nodes have no resistance and there are no interactions between neighboring branches or elements without connection.*

Furthermore, the following notation is used from here on, in which $n_I$ is the number of branches and $n_U$ is the number of nodes:

- the branch currents $I(t) \in \mathbb{R}^{n_I}$,
- the branch voltages $U(t) \in \mathbb{R}^{n_I}$,
- and the node voltages (voltage drop versus ground node) $u(t) \in \mathbb{R}^{n_U}$.

Due to the laws of conservation of energy, we can formulate the following theorems. Here $A \in \{-1,0,1\}^{n_U \times n_I}$ is an incidence matrix describing the branch-node connections of the network graph.

**Theorem 2.1** ***Kirchhoff´s current law (KCL)*** *The algebraic sum of currents traversing each cutset of the network must be equal to zero at every instant of time. Special case: The sum of currents leaving any circuit node is zero:*

$$A \cdot I(t) = 0.$$

**Theorem 2.2** ***Kirchhoff´s voltage law (KVL)*** *The algebraic sum of voltages along each loop of the network must be equal to zero at every instant of time. This is often used for getting a relation between branch voltage and node voltage in the form:*

$$A^T \cdot u(t) = U(t).$$

**Remark 2.2** *Applying KCL to all terminals of an element leads to the requirement of charge neutrality, i.e. the sum of charges $q_{kl}$ over all terminals $k$ of element $l$ must be constant.*

$$\sum_{k(l)} q_{kl} = const.$$

### 2.1.3 The modified nodal analysis

Back to our example. At first, one has to replace all branch currents by voltage depending functions, the so called characteristic equations of the elements. They are given by the physical behavior of each element. The branch currents $I_T(t)$ and voltages $U_T(t)$ for each element (in case of multiport elements even for each terminal, giving rise to the notation of index $T$) are given in admittance form. That means, $I_T(t)$ is an explicit function of $U_T(t)$. Let us examine some of these characteristic equations.
For linear resistors, Ohm's law

$$I_T(t) = G_i U_T(t)$$

with *conductance* $G_i$ is valid.
For linear capacitors, one has to use Faraday's law

$$I_T(t) = C_j \dot{U}_T(t)$$

with *capacitance* $C_j$. Note that $i$ and $j$ are the numbers of the actual element. These are characteristic equations for one-port elements. One has to distinguish between one-port and multi-port elements. An example for multiport elements are the bipolar transistors in the Schmitt trigger. If $U_{TC}(t) \geq U_{TB}(t) \geq U_{TE}(t)$, three equations are given, one for the base, one for the collector, and one for the emitter terminal,

$$\begin{aligned} I_{TB}(t) &= g(U_{TB}(t) - U_{TE}(t)), \\ I_{TC}(t) &= \alpha \cdot g(U_{TB}(t) - U_{TE}(t)), \\ I_{TE}(t) &= -(1+\alpha) \cdot g(U_{TB}(t) - U_{TE}(t)). \end{aligned}$$

Here, the characteristic current $g(U_T)$ of a pn-junction is

$$g(U_T) := \beta \cdot [\exp(U_T/U_{Temp}) - 1], \tag{2.3}$$

where $\alpha$ denotes the amplification factor, $\beta$ the saturation current and $U_{Temp}$ denotes the thermal voltage at room temperature.

After applying Kirchhoff´s current law we get

$$\begin{aligned} [1] \quad 0 &= G_1 \cdot (u_1 - V_{in}) + g(u_1 - u_3), \\ [2] \quad 0 &= G_2 \cdot (u_2 - V_{DD}) + C_0 \cdot (\dot{u}_2 - \dot{u}_4) + G_4 \cdot (u_2 - u_4) + \alpha \cdot g(u_1 - u_3), \\ [3] \quad 0 &= -(1+\alpha) \cdot g(u_1 - u_3) + G_3 \cdot u_3 - (1+\alpha) \cdot g(u_4 - u_3), \\ [4] \quad 0 &= G_4 \cdot (u_4 - u_2) + C_0 \cdot (\dot{u}_4 - \dot{u}_2) + g(u_4 - u_3), \\ [5] \quad 0 &= G_5 \cdot (u_5 - V_{DD}) + \alpha \cdot g(u_4 - u_3). \end{aligned}$$

After reformulating those equations, one gets the DAE (2.2). If we take a look at modeling electronic circuits in general, more characteristic equations may be required in order to describe different network elements. As mentioned before, there are the one-port-elements

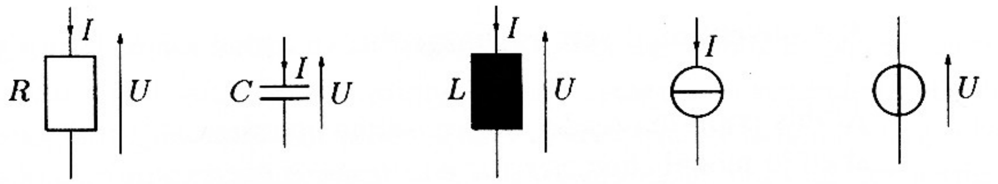

Figure 2.2: Basic network elements: Linear resistor ($I_T = U_T/R = G_i \cdot U_T$), capacitor ($I_T = C_j \cdot \dot{U}_T$), inductor ($U_T = L_k \cdot \dot{I}_T$), independent current source ($I_T = s_1(t)$) and independent voltage source ($U_T = s_2(t)$), source: [GFtM05].

(i.e. elements with 2 terminals), which are explained by equations relating their branch currents and branch voltages. Such basic elements are, for example resistors, inductors, and capacitors. On the other side, there are multiport elements. Here the branch currents entering any terminal and branch voltage across any pair of terminals have to be well-defined. The laws of Kirchhoff are also valid. A few elements are depicted in Fig. 2.2.
In general, there are different types of characteristic equations:

- The admittance form: the terminal currents are explicitly given.
- The impedance form: the branch voltages are explicitly given.
- Other forms including more physical quantities, e. g. electrical charge $q(t) \in \mathbb{R}^{n_q}$, magnetical flux $\phi(t) \in \mathbb{R}^{n_\phi}$ oriented formulations.

The reason for introducing such other formulations is the possibility to model charge conservation as well as the more natural access to this modeling problem. Therefore, we define terminal charges $q_T$, e.g., for a nonlinear capacitor with the charge $q_C(U_T)$, like

$$q_T = q_C(U_T),$$

and get the terminal current $I_T$

$$I_T = \frac{\mathrm{d}q_T}{\mathrm{d}t}.$$

For a direct comparison, please refer to Tab. 2.1. Now we are prepared for the Modified Nodal Analysis (MNA). There are also other methods like the Sparse Tableau Approach and the ordinary Nodal Analysis which are not explained here since in industrial applications (e. g. TITAN), mostly the MNA is used. With the help of Kirchhoff's laws and the characteristic equations for each element of the circuit, its behavior is fully described.

Now, we want to look at the charge/flux oriented formulation of the MNA. In addition to the node voltages, the branch currents $j_V$ and $j_L$ are included in the vector $x$, see (2.1). Note that $j_V$ and $j_L$ represent branch currents of voltage sources and inductors/flux sources,

| Conventional formulation | Charge/flux oriented formulation |
|---|---|
| Linear capacitor<br>$I_T = C_i \cdot \dot{U}_T$ | Charge/current defining element<br>$q_T = q_C(U_T, I_T), \qquad I_T = \mu(U_T) \cdot q_T + \dot{q}_T = I_{T_{dc}} + \dot{q}_T$ |
| Linear inductor<br>$U_T = L_k \cdot \dot{I}_T$ | Flux/voltage defining elements<br>$\phi_T = \phi_L(U_T, I_T), \qquad U_T = \dot{\phi}_T$ |

Table 2.1: Constitutive relations for energy storing elements, source [GFtM05].

i.e., elements without a simple characteristic equation in admittance form. Then, KCL is applied to each node except the ground node. The branch voltages are converted into node voltages with the help of KVL. The incidence matrix $A$ is split into element related incidence matrices $A_C$, $A_L$, $A_R$, $A_V$, and $A_I$ where

- $A_C, A_L$ describe charge/flux storing elements,
- $A_R$ describes resistors,
- $A_V, A_I$ describe voltage/current sources.

The charge/flux oriented formulation of the MNA then becomes:

$$\begin{aligned} A_C\dot{q} + A_R r(A_R^T u, t) + A_L j_L + A_V j_V + A_I i(A^T u, \dot{q}, j_L, j_V, t) &= 0 \\ \dot{\phi} - A_L^T u &= 0 \\ v(A^T u, \dot{q}, j_L, j_V, t) - A_V^T u &= 0 \\ q - q_C(A_C^T u) &= 0 \\ \phi - \phi_L(j_L) &= 0. \end{aligned} \tag{2.4}$$

Note that $u$ are the node voltages, $j_V$ and $j_L$ are the branch currents through voltage and flux controlled elements, $q$ and $\phi$ are charges and fluxes, $r$ are the voltage dependent resistors, $q_C$ and $\phi_L$ are voltage and flux dependent charge and flux sources and $i$ and $v$ the controlled current and voltage sources as in the antecedent example.

After inserting the flux/charge relations into the first three equations of (2.4), we get the conventional formulation of the MNA,

$$\begin{aligned} A_C\hat{C}(A_C^T u)A_C^T\dot{u} + A_R r(A_R^T u, t) + A_L j_L + A_V j_V + A_I i(A^T u, \dot{q}_C(A_C^T u), j_L, j_V, t) &= 0 \\ \hat{L}(j_L, t)\dot{j}_L - A_L^T u &= 0 \\ A_V^T u - v(A^T u, \dot{q}_C(A_C^T u), j_L, j_V, t) &= 0, \end{aligned} \tag{2.5}$$

with $\hat{L}(w) := \frac{\partial \phi_L(w)}{\partial w}$, $\hat{C}(w) := \frac{\partial q_C(w)}{\partial w}$, and $\hat{G}(w) := \frac{\partial r(w,t)}{\partial w}$.

The resulting equations of the MNA are directly related to the structure of the circuit.

### 2.1.4 Some structural aspects

The structure of the resulting DAE is highly dependent on the network elements in use and the topology of the circuit. This can be demonstrated considering the following equations. In general

$$\tilde{C}(A_C^T u) := A_C \hat{C}(A_C^T u) A_C^T$$

and

$$\tilde{G}(A_R^T u,t) := A_R \hat{G}(A_R^T u,t) A_R^T$$

are very sparse and have structural symmetry.
The type of the network elements has a big influence on the structure of the system. If there are no capacitors or inductors, that means no energy storing elements, the system degenerates to pure algebraic equations. Without time dependent elements, e. g., purely time-dependent current and voltage sources, the system is homogeneous. The missing of controlled sources leads to a Jacobian of the left hand side of (2.5) with respect to $u$, $j_L$, and $j_V$ of the form

$$G(A_R^T u,t) := \begin{pmatrix} \tilde{G}(A_R^T u,t) & A_L & A_V \\ -A_L^T & 0 & 0 \\ -A_V^T & 0 & 0 \end{pmatrix},$$

which is also sparse and has structural symmetry.
When using the MNA, cutsets of current sources and loops of voltage sources are strictly prohibited. For that reason, the matrix $(A_C, A_R, A_V, A_L)$ has full row rank and $A_V$ has full column rank. If it is the case that loops of independent voltage sources and/or inductors or cutsets of independent voltage sources and/or capacitors exist, then the Jacobian matrix $G(A_R^T u,t)$ is singular and no steady-state solution can be computed. So almost all circuit simulators are checking these conditions. Finally, if there are some nodes which have no connection to ground via energy storing elements, the matrix $\tilde{C}(A_C^T u)$ is singular. Unfortunately this is the case for most circuits, but violates some requirements for using certain algorithms in MOR. Later we will present a solution for this dilemma. The Jacobian matrix of (2.5) with respect to $\dot{u}$, $\dot{j}_L$, and $\dot{j}_V$ now becomes

$$C(A_C^T u,t) := \begin{pmatrix} \tilde{C}(A_C^T u) & 0 & 0 \\ 0 & \hat{L}(j_L) & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

### 2.1.5 The index concept for linear systems

**Definition 2.6 (Linear systems)** *If the simulated circuit contains only linear elements or the system is linearized at an operating point $(x(t_0), \dot{x}(t_0))$, it is called a* linear system.

If we work with linear systems, the network equations represent linear DAEs of the form:

$$C\dot{x}+Gx=s(t), \quad x(t_0)=x_0. \tag{2.6}$$

In the formulation of the MNA,

$$x=\begin{pmatrix} u \\ j_L \\ j_V \end{pmatrix}, \quad s=\begin{pmatrix} -A_I i(t) \\ 0 \\ -v(t) \end{pmatrix},$$

and in case of a linear system

$$C=\begin{pmatrix} A_C\hat{C}A_C^T & 0 & 0 \\ 0 & \hat{L} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and

$$G=\begin{pmatrix} A_R\hat{G}A_R^T & A_L & A_V \\ -A_L^T & 0 & 0 \\ -A_V^T & 0 & 0 \end{pmatrix}.$$

This is our starting point for an index concept that, in this case, only depends on $C$ and $G$.

The matrix $C$ is regular if and only if the circuit contains no voltage sources and no nodes without path to the ground via capacitors. If $C$ is regular, the system represents a linear implicit system of ODEs, which can be transformed into an explicit ODE system of the form

$$\dot{x}=C^{-1}(-Gx+s(t)).$$

If $C$ is singular, we assume $G$ to be regular and perform the following steps. Multiplying with $G^{-1}$ from the left leads to

$$G^{-1}C\dot{x}+x=G^{-1}s(t). \tag{2.7}$$

After a Jordan decomposition

$$G^{-1}C=T^{-1}\begin{pmatrix} \check{C} & 0 \\ 0 & N \end{pmatrix}T6 \tag{2.8}$$

with a regular matrix $T$, we insert (2.8) in (2.7) and multiply with $T$ from the left. As result we get

$$\begin{pmatrix} \check{C} & 0 \\ 0 & N \end{pmatrix}T\dot{x}+Tx=TG^{-1}s(t)$$

with a regular matrix $\check{C}$ and a nilpotent matrix $N$ with nilpotency index $\nu\in\mathbb{N}$:

$$N^{\nu}=0, \quad N^{\nu-1}\neq 0. \tag{2.9}$$

After a transformation of the form

$$\begin{pmatrix} y \\ z \end{pmatrix} := Tx, \quad \text{and} \quad \begin{pmatrix} \eta(t) \\ \delta(t) \end{pmatrix} := TG^{-1}s(t),$$

the system decouples into an explicit ordinary differential equation and a nilpotent part like

$$\begin{array}{rcl} \dot{y} & = & \check{C}^{-1}(\eta(t) - y), \\ N\dot{z} & = & \delta(t) - z, \end{array} \tag{2.10}$$

referring to the differential variables $y$ and the algebraic variables $z$.

**Definition 2.7 (Algebraic Index)** *The number* $\nu$ *in (2.9) is called the* algebraic index *of the DAE. We distinguish between:*

- *Index-1 case:* $\nu = 1$.
  *Here,* $N = 0$*, and the nilpotent part of (2.10) becomes* $z = \delta(t)$*. The algebraic variables are given by the input signal* $\delta(t)$ *and we need exactly one differentiation to get an explicit ODE for* $z$*.*

- *Higher-index case:* $\nu > 1$.
  *The nilpotent part of (2.10) is more complex. We need a differentiation process. At least, the algebraic variables are given by the input signal* $\delta(t)$ *and its derivatives. After a last differentiation, exactly the* $\nu$*-th one, we get an explicit ODE system. We have to expect numerical problems in higher index cases.*

**Remark 2.3** *The algebraic index is not the only existing definition of the index of a DAE system. Other definitions including the differentiation index, which is for linear systems equal to the algebraic index, the perturbation index, and the tractability index are given in [GFtM05].*

As a summary one can say that the structure (topology, elements, etc.) of a circuit and the structure of the corresponding network equations are significantly related.

## 2.2 The Transfer Function

This section relates to [MS05]. In most industrial applications, the modeling of our subcircuit leads to a linear time-invariant continuous-time system (2.6), which can be represented as

$$\begin{array}{rcl} C\dot{x}(t) & = & -Gx(t) + Bu(t), \quad x(0) = x_0, \\ y(t) & = & Lx(t) + Fu(t), \end{array} \tag{2.11}$$

with

- $C, G \in \mathbb{R}^{n \times n}$,
- $B, L^T \in \mathbb{R}^{n \times m}$,
- $x \in \mathbb{R}^n$ containing internal state variables,
- $u \in \mathbb{R}^m$ the vector of input variables, e. g., the terminal currents,
- $y \in \mathbb{R}^m$ representing the output vector, e. g., the terminal voltages,
- $x_0 \in \mathbb{R}^n$ the initial value and
- $n$ the number of state variables, called the *order* of the system.

Since $F \in \mathbb{R}^{m \times m}$ is either equal to zero or must not be changed during the reduction, we will not consider it in this section. It will be considered later on while discussing balanced truncation methods.

### 2.2.1 Definitions

**Definition 2.8 (State space system, descriptor system)** *If $C = I_n$ (identity matrix), then (2.11) is a* standard state space system. *Otherwise it is a* descriptor system *or* generalized state space system.

We are going to handle descriptor systems. Recall that the *Laplace transformation* of a function $f(t)$, $t \in \mathbb{R}$, is given by

$$\mathbf{f}(s) = \mathcal{L}[f(t)] = \int_0^\infty e^{-st} f(t) \mathrm{d}t,$$

where $s$ is a complex variable. If $s = i\omega$, then $\omega$ is called the*frequency* (see also Def. 2.11). We apply this to the descriptor system (2.11) and utilize that $\mathcal{L}[\dot{x}(t)] = s\mathbf{x}(s) - x(0)$. Thus, we get

$$\mathbf{y}(s) = L(sC+G)^{-1}B\mathbf{u}(s) + L(sC+G)^{-1}Cx(0). \tag{2.12}$$

Because of the term $(sC+G)^{-1}$, (2.12) is only well defined if the matrix pencil $(sC+G)$ is regular. This is one of our assumptions, see Def. 2.13. $\mathbf{u}(s)$, $\mathbf{x}(s)$ and $\mathbf{y}(s)$ are the Laplace transforms of $u(t)$, $x(t)$ and $y(t)$, respectively.

**Definition 2.9 (Transfer function)** *The rational matrix-valued function*

$$H(s) = L(sC+G)^{-1}B \tag{2.13}$$

*is called the* transfer function *of the continuous-time descriptor system (2.11).*

With the help of (2.12), it is obvious that if $Cx(0) = 0$, $H(s)$ gives the relation between Laplace transforms of the input $u(t)$ and the output $y(t)$. Hence the transfer function $H(s)$ directly maps the input vector $u(s)$ onto the output vector $y(s)$ and describes the input/output behavior of (2.11) in the frequency domain. MOR can be seen as a process of approximating the transfer function.

**Definition 2.10 (Proper transfer function)** *The transfer function $H(s)$ is called* proper *if*

$$\lim_{s \longrightarrow \infty} \|H(s)\|_2 < \infty,$$

*where $\| \,.\, \|_2$ is the spectral norm of Rem. 3.4, i. e. a proper transfer function will never grow unbounded as the frequency approaches infinity. Otherwise it is called* improper. *It is called* strictly proper *if*

$$\lim_{s \longrightarrow \infty} \|H(s)\|_2 = 0.$$

**Definition 2.11 (The frequency response)** *A* frequency response *of the descriptor system (2.11) is given by $H(i\omega)$. For an input function $u(t) = e^{i\omega t}u_0$ with $\omega \in \mathbb{R}$ and $u_0 \in \mathbb{R}^m$, we get from (2.11) that*

$$y(t) = H(i\omega)e^{i\omega t}u_0.$$

*Thus, the frequency response $H(i\omega)$ gives a transfer relation from the periodic input $u(t) = e^{i\omega t}u_0$ to the output $y(t)$.*

### 2.2.2 Properties of the transfer function

In order to reach certain reasonable simulation results, the transfer function should satisfy restrictions. In the following we introduce some basic concepts and explain properties of the transfer function like stability and passivity.

**Definition 2.12 (Matrix pencil)** $(\lambda C + G)$ *with the matrices $C, G \in \mathbb{R}^{n \times n}$ is called a* matrix pencil *or just* pencil *where $\lambda \in \mathbb{C}$ is considered undetermined.*

**Definition 2.13 (Regularity)** *Let $C, G \in \mathbb{R}^{n \times n}$. If $C$ and $G$ are square matrices and there exists a $\lambda \in \mathbb{C}$ such that* $\det(\lambda C + G) \neq 0$, *then the pencil $(\lambda C + G)$ is called* regular. *Otherwise, it is called* singular. Eigenvalues of the matrix pencil $(\lambda C + G)$ *are defined to be the roots of the characteristic polynomial, $P(\lambda) = 0$, with*

$$P(\lambda) = \det(\lambda C + G).$$

*$\Lambda(C, G)$ is the set of eigenvalues of the pencil $(\lambda C + G)$.*

**Definition 2.14** *Equivalence Two matrix pencils $(\lambda C_1 + G_1)$ and $(\lambda C_2 + G_2)$ are called* equivalent, *if and only if there exist T and S such that*

$$\lambda C_1 + G_1 = \lambda T C_2 S + T G_2 S.$$

**Definition 2.15 (Weierstrass canonical form)** *Let $(\lambda C + G)$ be a regular matrix pencil and $(\lambda TCS^{-1} + TGS^{-1})$ be a pencil equivalent to $(\lambda C + G)$. Furthermore,*

$$(\lambda C + G) = T \begin{pmatrix} J_{n_1}(\lambda_1) - \lambda I_{n_1} & & & & & \\ & \ddots & & & & \\ & & J_{n_k}(\lambda_k) - \lambda I_{n_k} & & & \\ & & & N_{m_1} & & \\ & & & & \ddots & \\ & & & & & N_{m_r} \end{pmatrix} S^{-1}, \qquad (2.14)$$

*where $J_{n_i}(\lambda_i)$ are the $n_i \times n_i$ Jordan blocks referring to the eigenvalues $\lambda_i$, and*

$$N_{m_i} = I_{m_i} - \lambda J_{m_i}(0),$$

*are $m_i \times m_i$ blocks referring to $\lambda = \infty$ with multiplicity $m_i$. $J_{m_i}(0)$ has a nilpotent structure and (2.14) is called the* Weierstrass canonical form *(WCF) of the matrix pencil $(\lambda C + G)$.*

### Controllability and observability

We need to define controllability and observability because we need these expressions in methods like Balanced Truncation. For descriptor systems, there are a lot of notions of controllability and observability [Cob84, YS81]. Here, we want to mention concepts of complete controllability and observability [MS05].

**Definition 2.16 (Complete controllability (C-controllability))** *The descriptor system (2.11) is called* completely controllable *or* C-controllable *if*

$$\text{rank}[\alpha C + \beta G, B] = n \quad \textit{for all} \quad (\alpha, \beta) \in (\mathbb{C} \times \mathbb{C}) \backslash \{(0,0)\}.$$

In fact, C-controllability implies that for any given initial state $x_0 \in \mathbb{C}^n$ and final state $x_f \in \mathbb{C}^n$ , there exists a control input $u(t)$ that transfers the system from $x_0$ to $x_f$ in finite time.

**Definition 2.17 (Complete observability (C-observability))** *The descriptor system (2.11) is called* completely observable *or* C-observable *if*

$$\text{rank}[\alpha C^T + \beta G^T, L^T] = n \quad \textit{for all} \quad (\alpha, \beta) \in (\mathbb{C} \times \mathbb{C}) \backslash \{(0,0)\}.$$

As one can see, observability is the dual property of controllability. It implies that if the output is zero for all solutions of the system with zero input, then this system has only the trivial solution. The following theorem gives equivalent conditions for system (2.11) to be C-controllable and C-observable.

**Theorem 2.3** [YS81] *Consider a descriptor system (2.11), where* $(\lambda C + G)$ *is regular.*

*i) System (2.11) is C-controllable if and only if* $\operatorname{rank}[\lambda C + G, B] = n$ *for all finite* $\lambda \in \mathbb{C}$ *and* $\operatorname{rank}[C, B] = n$.

*ii) System (2.11) is C-observable if and only if* $\operatorname{rank}[\lambda C^T + G^T, L^T] = n$ *for all finite* $\lambda \in \mathbb{C}$ *and* $\operatorname{rank}[C^T, L^T] = n$.

For more characterizations of controllability and observability see [Cob84, Dai89].

### Stability

A dynamical system is stable if its impulse response is bounded as a function of $t$. For a linear time-invariant system this means, that the impulse response converges to a constant value or a sinusoidal signal, which is generated around a constant value. A continuous time system is stable if and only if all its poles $p_i$ are located in the left-hand side of the complex plane, i. e.,

$$\operatorname{Re}(p_i) \leq 0.$$

If there are poles on the imaginary axis, the multiplicity has to be one. For descriptor systems, stability can be defined more in detail, see [Dai89, Sty02].

**Definition 2.18 (Asymptotic stability)** *The descriptor system (2.11) is called* asymptotically stable *if* $\lim_{t \to \infty} x(t) = 0$ *for all solutions* $x(t)$ *of* $C\dot{x}(t) = -Gx(t)$.

There are more equivalent conditions for the system to be asymptotically stable, which are summarized in the following theorem:

**Theorem 2.4** [Dai89, Sty02] *Consider a descriptor system (2.11) with a regular pencil* $(\lambda C + G)$ *and WCF as in (2.15). The following statements are equivalent.*

*i) System (2.11) is asymptotically stable.*

*ii) All finite eigenvalues of the pencil* $(\lambda C + G)$ *lie in the open left half-plane, i.e.* $\Lambda(C, G) \setminus \{\infty\} \subset \mathbb{C}^- := \{z \in \mathbb{C} | \operatorname{Re}(z) < 0\}$.

*iii) The projected generalized continuous-time Lyapunov equation*

$$C^T X(-G) + (-G)^T XC = -P_r^T Q P_r, \quad X = P_l^T X P_l,$$

*has a unique Hermitian, positive semidefinite solution $X$ for every Hermitian, positive definite matrix $Q$.*
*Here, $k = \sum_i n_i$, $P_l$ and $P_r$ of the form*

$$P_l = T \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix} T^{-1},$$

$$P_r = S \begin{pmatrix} I_k & 0 \\ 0 & 0 \end{pmatrix} S^{-1},$$

*with $T$, $S$, and $n_i$ out of (2.14), are the left and the right spectral projections onto the left and the right deflating subspaces of the matrix pencil $(\lambda C + G)$ corresponding to the finite eigenvalues.*

We call the pencil $(\lambda C + G)$ *c-stable* if it is regular and all the finite eigenvalues of $(\lambda C + G)$ have negative real part.

### Passivity

**Definition 2.19 (Passivity)** *A linear system is* passive *if*

$$\int_{-\infty}^{t} u(\tau)^T y(\tau) d\tau \geq 0, \qquad \forall t \in \mathbb{R}, \forall u \in L_2(\mathbb{R}, \mathbb{R}^m).$$

Passive systems cannot produce energy. A passive system is also stable. Hence our reduced model has to be passive if we want to reduce passive circuit elements. Otherwise it could produce energy from nothing and the values of the function would explode. Each of the following points secures passivity of a system:

- its transfer function $H(s)$ has no poles in $\mathbb{C}^+ := \{s \in \mathbb{C} : \mathrm{Re}(s) > 0\}$ and $\mathrm{Re}(H(i\omega)) \geq 0, \forall \omega \in \mathbb{R}$;

- its transfer function $H(s)$ is *positive real*, i.e.
  1. $\overline{H(\overline{s})} = H(s)$,
  2. $H$ is analytic in $\mathbb{C}^+$,
  3. $H(s) + H^H(s) \geq 0, \forall s \in \mathbb{C}^+$;

- the following three conditions are all true:
  1. $L^T = B$,
  2. $(C + C^T)$ is positive semi-definite, i.e. $x^T(C + C^T)x \geq 0, \forall x \in \mathbb{R}^n$,
  3. $(G + G^T)$ is positive semi-definite.

## 2.3 Overview over different methods

Although this work focuses mainly on subcircuits with a huge number of input and output terminals and the order reduction of the transfer functions of these systems, we want to show some selected ideas of MOR to give an overview. We use these methods later for reducing the order of a terminal reduced transfer function. Besides some special approaches like symbolic MOR, two major concepts are widely used. One is based on frequency domain projection methods and the other one on approximation with the help of balanced truncation. Projection methods are e. g. explicit moment or point matching methods like the asymptotic waveform evaluation (AWE) or implicit ones like the Krylov-subspace methods. The Poor Man's truncated balanced realization (TBR) is motivated as a connection between both major concepts. The idea therefore was firstly introduced as a frequency domain proper orthogonal decomposition (POD). Balanced truncation methods are also well explored. Here we want to explain the square root variant (BTSR) of balanced truncation. There exist a lot of other variants, like e. g. the Generalized Square Root Balancing Free (GSRBF) method, the Generalized Schur-Hammarling Square Root methods or the Generalized LR-ADI method.

Further information about the general MOR concept is available in form of a vast amount of books, papers, articles, etc, e. g. [FF94, WP02, Moo81, OCP98, TH07, SKEW96].

### 2.3.1 Krylov-subspace methods

Throughout this section, we will only consider SISO (Single-Input, Single-Output) systems for the sake of an easier notation. However, all methods covered can be extended to the MIMO (Multiple-Input, Multiple-Output) case. Furthermore, we assume $G$ to be nonsingular so that we can write the system (2.11) as

$$\begin{aligned} E\dot{x}(t) &= x(t) + Du(t) \\ y(t) &= Lx(t) \end{aligned} \tag{2.15}$$

with $E = -G^{-1}C$ and $D = -G^{-1}B$.

As mentioned above, by using a Laplace transformation, (2.15) can be transformed into the frequency domain

$$\begin{aligned} sEx(s) &= x(s) + Du(s), \\ y(s) &= Lx(s). \end{aligned} \tag{2.16}$$

Obviously, the transfer function here becomes

$$H(s) := -L(I_n - sE)^{-1}D.$$

In addition, the concepts of the following section are based on the work of Tobias Rothaug [Rot05] and [SKEW96, FF94, KD99, OCP98, Fro03].

### General Krylov-subspace methods

To understand Krylov-subspace methods, some basic definitions need to be introduced first.

**Definition 2.20 (Order k Krylov-subspace)** *The* order $k$ Krylov-subspace *generated by matrix A and vector x is defined as*

$$K_k(A,x) = \operatorname{span}\left\{x, Ax, A^2x, ..., A^{k-1}x\right\}.$$

Krylov-subspace MOR methods try to reduce the number of internal variables by projecting the vector $x$ onto a subspace $W_q$ of dimension $q \ll n$. Therefore, $x$ can be written as

$$x = W\hat{x} + p,$$

where $\hat{x} \in W_q$ and $p \in W_q^{\perp}$.
$W_q^{\perp}$ is the orthogonal complement of $W_q$, which is defined as follows:

**Definition 2.21 (Orthogonal complement)** *Let V be a vector space and W be a subspace of V. Then the* orthogonal complement *of W in V is the set of vectors u such that u is orthogonal to all vectors in W.*

Inserting this into the system in frequency domain yields

$$\begin{aligned} sE(W\hat{x}(s) + p) &= W\hat{x}(s) + p + Du(s), \\ y(s) &= L(W\hat{x}(s) + p), \end{aligned}$$

with the residuum

$$R = sEW\hat{x}(s) - W\hat{x}(s) - Du(s) = p - sEp.$$

Thus, we have reduced the number of state variables. However, the system still contains $n$ equations. Hence, we multiply by $V^T \in \mathbb{R}^{q\times n}$. If $W$ and $V$ are bi-orthogonal,
i. e., $V^TW = I_q$, and $\operatorname{colspan}(V) \perp R$, we have a reduced order model

$$\begin{aligned} s\hat{E}\hat{x}(s) &= \hat{x}(s) + \hat{D}u(s), \\ \hat{y}(s) &= \hat{L}\hat{x}(s), \end{aligned} \tag{2.17}$$

with $\hat{E} = V^TEW, \hat{D} = V^TD, \hat{L} = LW$.

Next we have to find a proper choice for $W$ and $V$. From (2.16) in frequency domain we get

$$x(s) = -(I_n - sE)^{-1}Du(s),$$

which, if $\|sE\| < 1$, can be transformed by the *Neumann series* to

$$x(s) = -\sum_{k=0}^{\infty} s^k E^k Du.$$

Hence $x \in \text{span}\left\{D, ED, E^2D, ...\right\}$,

$$x \approx \left[D, ED, E^2D, \cdots E^{q-1}D\right]\hat{x} = W\hat{x}.$$

Expanding the original transfer function $H$ into a Taylor series around the expansion point $s = 0$ yields

$$H(s) = H(0) + \left.\frac{dH}{ds}\right|_{s=0} s + \frac{1}{2!}\left.\frac{d^2H}{ds^2}\right|_{s=0} s^2 + \frac{1}{3!}\left.\frac{d^3H}{ds^3}\right|_{s=0} s^3 + \ldots \tag{2.18}$$

$$= m_0 + m_1 s + m_2 s^2 + m_3 s^3 + \ldots \tag{2.19}$$

On the other hand, $H$ can also be presented by means of the Neumann series as

$$H(s) = -L(I_n - sE)^{-1}D \tag{2.20}$$

$$= -\sum_{k=0}^{\infty} (LE^kD)s^k \tag{2.21}$$

$$= -LD - LEDs - LE^2Ds^2 - ... \tag{2.22}$$

$$= \sum_{k=0}^{\infty} m_k s^k. \tag{2.23}$$

Thus, Krylov-subspace methods perform implicit moment matching, i.e., the reduced transfer function is

$$H(s) \approx \hat{H}(s) = \sum_{k=0}^{\infty} \hat{m}_k s^k,$$

with $\hat{m_k} = m_k$ for $k = 0, \cdots, l$.

### Arnoldi method

The original Arnoldi algorithm was introduced in 1951 for the calculation of certain eigenvalues of large, sparse matrices. In 1990 Silveira, Kamon, White, Elfadel and Ling used it first for reducing models of electric circuits [SKEW96]. In this case, we choose $s = 0$ as expansion point of the Taylor series and $W = V$, $W$ is orthogonal, i. e. $W^TW = I_q$, so that

$$K_q(E, D) = \text{span}\left\{D, ED, E^2D, ..., E^{q-1}D\right\},$$

and

$$W = \left[D, ED, E^2D, \cdots, E^{q-1}D\right].$$

Thus we can match the first $q$ moments of the transfer function, but unfortunately the reduced order model may not be passive.

### Padé via Lanczos method (PVL)

This algorithm was designed by Feldmann and Freund in 1994 [FF94]. The expansion point is again $s = 0$ but $W$ and $V$ are bi-orthogonal now, i.e. $V^T W = I_q$. We get

$$K_q(E,D) = \operatorname{span}\left\{D, ED, E^2D, ..., E^{q-1}D\right\}$$

with $W = \left[D, ED, E^2D, \cdots, E^{q-1}D\right]$, and

$$K_q(E^T, L^T) = \operatorname{span}\left\{L^T, E^T L^T, (E^T)^2 L^T, ..., (E^T)^{q-1} L^T\right\}$$

with $V = \left[L^T, E^T L^T, (E^T)^2 L^T, \cdots, (E^T)^{q-1} L^T\right]$.
Thus the first $2q$ moments can be matched. In order to bi-orthogonalize the columns of $W$ and $V$ the Lanczos algorithm is used since it is robust and efficient. Unfortunately, the algorithm does not preserve passivity and stability either.

### SVD-Laguerre method

This method was proposed by Knockaert and De Zutter in 1999 [KD99]. This approach uses the Laguerre functions in the $s$-domain

$$\Phi_k^\alpha(s) = \frac{\sqrt{2\alpha}}{s+\alpha}\left(\frac{s-\alpha}{s+\alpha}\right)^k, k = 0, 1, 2, ...,$$

which form an orthonormal basis in the frequency domain. Hence the transfer function of (2.13) can be expressed in terms of these functions:

$$H(s) = \frac{2\alpha}{s+\alpha} L \sum_{k=0}^{\infty} \left((G+\alpha C)^{-1}(G-\alpha C)\right)^k (G+\alpha C)^{-1} B \left(\frac{s-\alpha}{s+\alpha}\right)^k,$$

so that the following Krylov subspace can be built up as

$$K_q\left((G+\alpha C)^{-1}(G-\alpha C), (G+\alpha C)^{-1}B\right).$$

The orthogonalization of the columns can be done by a SVD after all columns have been computed. That will create progressively linearly dependent columns. A better way is to orthogonalize immediately after a new column has been computed with an Arnoldi-type algorithm. Doing so leads to a stable and passivity preserving algorithm.

### Passive Reduced-order Interconnect Macro-modeling Algorithm (PRIMA)

PRIMA is an improved Arnoldi method. It was presented by Odabasioglu, Celik and Pileggi in 1997 [OCP98]. Again, we choose $V = W$ and $W^T W = I_q$ and use the Krylov-space

$$K_q(-G^{-1}C, G^{-1}B) = \operatorname{span}\left\{G^{-1}B, -G^{-1}CG^{-1}B, ..., (-G^{-1}C)^{q-1}G^{-1}B\right\},$$

so that the first $q$ moments can be matched. The main difference between Arnoldi and PRIMA is that Arnoldi applies the projection to (2.15),

$$\begin{aligned} -G^{-1}C\dot{x}(t) &= x(t) - G^{-1}Bu(t), \\ y(t) &= Lx(t), \end{aligned}$$

in contrast to PRIMA that works on the original system (2.11)

$$\begin{aligned} C\dot{x}(t) &= -Gx(t) + Bu(t), \\ y(t) &= Lx(t). \end{aligned}$$

Matrices $C$ and $G$ are typically positive semi-definite which is important for passivity, see Def. 2.19, but the product $G^{-1}C$ might not be. Hence Arnoldi could lose passivity while PRIMA directly reduces the system's matrices

$$\hat{C} = W^T CW, \hat{G} = W^T GW,$$

thus preserving passivity, see Def. 2.19.

The last two methods, SVDLG and PRIMA, are available in the KROM code package, see the Codestar project [Sch04], which is used in the circuit simulator TITAN, the in-house analog simulator of Qimonda and Infineon.

#### Open Problems with Krylov-subspace methods

However, all Krylov-subspace methods have two major drawbacks in practical applications. First, it is very difficult to control the error. Error estimators do exist for some methods but they require additional computation which can be expensive and awkward to implement and produce error estimates only at single frequency points which leaves open the problem of error estimation over a range of frequencies. Here we try to solve these problems with the help of balanced truncation methods. Second, moment-based methods like PRIMA are known in some cases to produce models that are “too high” in order with the obvious consequences in terms of analysis or simulation cost.

### 2.3.2 Balanced truncation methods

In this subsection we want to give a short overview about MOR with the help of balanced truncation. We want to use this method to realize the reduction of $H(s)$, later also for the reduction of a terminal reduced transfer function $H_r(s)$, see Chap. 3. We follow [Ben03, BQOQO04, MS05]. Please note that the method is essentially mathematically equivalent to methods proposed in [Sty02, Sty04]. We start with some definitions followed by some assumptions.

### Definitions and assumptions

**Definition 2.22 (Realization)** *For any rational matrix-valued function $H(s)$, there exist matrices $B, C, -G$ and $L$ such that $H(s) = L(sC+G)^{-1}B$. A descriptor system (2.11) with these matrices is called a* realization *of H(s). We will denote such a realization by*

$$H = [B, C, -G, L].$$

**Remark 2.4** *A realization is not unique in general.*

**Definition 2.23 (Minimal realization)** *A realization $[B, C, -G, L]$ of the transfer function $H(s)$ is called* minimal *if the dimension of the matrices $C$ and $-G$ is as small as possible, so that still the outputs are the same for all possible inputs.*

For necessary and sufficient conditions for a realization to be minimal please refer to [MS05].

The computation for the reduced order model is based on the following assumptions.

- the matrix pencil $(\lambda C + G)$ is regular,
- the matrix pencil $(\lambda C + G)$ is stable, see Def. 2.18 and Thm. 2.4, i. e.,
$$\Lambda(C, G) \backslash \{\infty\} \subset \mathbb{C}^-,$$
- it should make sense to reduce the system, that means the number of inputs/outputs should be much smaller than the number of generalized states.

In the following, the computational steps for a simple kind of balanced truncation are explained. As a result, we would like to get the reduced matrices $\widetilde{B}, \widetilde{C}, \widetilde{G}, \widetilde{L}$ and $\widetilde{F}$ of the reduced transfer function

$$\widetilde{H}(s) = \widetilde{L}(s\widetilde{C} + \widetilde{G})^{-1}\widetilde{B} + \widetilde{F}.$$

Our first goal is to separate and decouple the transfer function in the following way:

$$H(s) = H_0(s) + H_\infty(s),$$

where $H_0$ is strictly proper (i. e., $H_0(\infty) = 0$, see Def. 2.10) and $H_\infty$ is polynomial. $H_0(s)$ and $H_\infty(s)$ contain the finite and infinite poles of the system, respectively.
To achieve this, we use the disk function [BDG97, BB97]. It is also possible to use other spectral projection methods [Rob80, Lau91] to decouple the transfer function, depending on which method or algorithm is chosen. Furthermore we apply balanced truncation to the strictly proper part $H_0(s)$ and obtain the reduced order model

$$\widetilde{H}(s) = \widetilde{H}_0(s) + H_\infty(s).$$

As mentioned above we are using the matrix disk function, which is explained in the following.

### Decoupling the system

If we are able to find a transformation defined by the nonsingular matrices $U,V \in \mathbb{R}^{n\times n}$ such that

$$-\widehat{G}-\lambda\widehat{C} := U(-G-\lambda C)V = -\begin{bmatrix} G_0 & 0 \\ 0 & G_\infty \end{bmatrix} - \lambda \begin{bmatrix} C_0 & 0 \\ 0 & C_\infty \end{bmatrix}, \tag{2.24}$$

where $-G_0-\lambda C_0$ contains the finite poles and $-G_\infty-\lambda C_\infty$ the infinite poles of the matrix pencil, we can apply this to the rest of the realization and we get

$$\widehat{B} := UB =: \begin{bmatrix} B_0 \\ B_\infty \end{bmatrix}, \quad \widehat{L} := LV^{-1} =: \begin{bmatrix} L_0 & L_\infty \end{bmatrix}.$$

As a consequence, we obtain the decomposition

$$\begin{aligned} H(s) &= \widehat{L}(s\widehat{C}+\widehat{G})^{-1}\widehat{B} \\ &= \begin{bmatrix} L_0 & L_\infty \end{bmatrix} \begin{bmatrix} (sC_0+G_0)^{-1} & 0 \\ 0 & (sC_\infty+G_\infty)^{-1} \end{bmatrix} \begin{bmatrix} B_0 \\ B_\infty \end{bmatrix} \\ &= \left\{L_0(sC_0+G_0)^{-1}B_0\right\} + \left\{L_\infty(sC_\infty+G_\infty)^{-1}B_\infty\right\} \\ &=: H_0(s)+H_\infty(s). \end{aligned}$$

The problem here is to find the appropriate matrices $U$ and $V$ for the block-diagonalization. Since this is fairly difficult, we have to make an intermediate step. We search orthogonal matrices $Q,Z \in \mathbb{R}^{n\times n}$ such that,

$$Q^T(-G-\lambda C)Z = \begin{bmatrix} -G_0 & W_G \\ 0 & -G_\infty \end{bmatrix} - \lambda \begin{bmatrix} C_0 & W_C \\ 0 & C_\infty \end{bmatrix}. \tag{2.25}$$

In order to find such matrices $Q$ and $Z$, we use the Matrix Disk Function in the following way.

Let $Z-\lambda Y$ be regular and in Weierstraß (Kronecker) canonical form, see Def. 2.15,

$$Z-\lambda Y = T\begin{bmatrix} J_0-\lambda I_k & 0 \\ 0 & J_\infty-\lambda N \end{bmatrix} S^{-1},$$

where

- $J_0$ contains Jordan blocks corresponding to $\Lambda(Z,Y)\cap B_1(0)$, $B_1(0) := \{z \in \mathbb{C} \mid |z| < 1\}$,
- $J_\infty - \lambda N$ contains Jordan blocks and nilpotency structure corresponding to $\Lambda(Z,Y)\cap \{|z|>1\}$ including the infinite eigenvalues.

Then

$$\operatorname{disk}(Z,Y) := S\left(\begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & 0 \\ 0 & I_{n-k} \end{bmatrix}\right) S^{-1} =: D_Z - \lambda D_Y,$$

- $D_Z$ is the projector onto the *right* d-stable deflating subspace of $Z - \lambda Y$,
- $D_Y$ is the projector onto the *right* d-antistable deflating subspace of $Z - \lambda Y$.

Note that for index-1 systems, e.g., for systems resulting from the modeling of RLC circuits, we have $N = 0$. In the following, this property will be assumed to be satisfied.

Computing the subspaces works as follows:

- Set the starting values $Z_0 = Z, Y_0 = Y$.
- FOR $j = 0, 1, 2, \ldots$

  Compute QR decomposition $\begin{bmatrix} Y_j \\ -Z_j \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} R_j \\ 0 \end{bmatrix}$,

  Set $Z_{j+1} = U_{12}^T Z_J$ and $Y_{j+1} = U_{22}^T Y_j$.

  END FOR
- Set $Z_\infty := \lim_{j\to\infty} Z_j$ and $Y_\infty := \lim_{j\to\infty} Y_j$.

Then $\operatorname{disk}(Z,Y) = (Z_\infty + Y_\infty)^{-1}(Y_\infty - \lambda Z_\infty) = D_Z - \lambda D_Y$, see [BB97].

In our case, we choose as input to the disk function method $(-\alpha G, C)$. The need for the parameter $\alpha$ in front of $G$ is explained next. The Disk Function computes a spectral projection w.r.t. the unit circle, but the stable poles may be anywhere in $\mathbb{C}^-$. A solution for this is the conformal mapping (Moebius transformation)

$$(-G, C) \to (-\alpha G + \beta C, -\gamma G + \delta C).$$

Now we set $\beta = \gamma = 0, \delta = 1$ so that $B_1(0) \to B_{\frac{1}{\alpha}}(0)$. $\alpha$ is to be chosen such that $B_{\frac{1}{\alpha}}(0)$ encircles all finite poles. Thus, we have to choose $\alpha$ in order to split the finite/infinite poles of the spectrum. Here it is tricky to determine the optimal $\alpha$. This is an open problem, perhaps the master thesis of René Günzel [Gün08] will investigate this particularly.
After computing projectors on the deflating subspaces corresponding to the finite and infinite eigenvalues, a block-triangularization (2.25) of $G - \lambda C$ is achieved. Note that we used an inverse-free subspace extraction algorithm [BDG97] within the disk function to get matrices $Q$ and $Z$ like shown in (2.25).

After this intermediate step, (2.25) is reduced to the required block-diagonal form by solving the generalized Sylvester equation

$$W_G - G_0 Y - X G_\infty = 0, \qquad W_C + C_0 Y + X C_\infty = 0, \tag{2.26}$$

or in case of index-1 systems, like supposed before, (2.26) becomes two linear system equations

$$X G_\infty = W_G - G_0 Y, \qquad W_C = -C_0 Y,$$

since $C_\infty = 0$. Thus, we have achieved our first goal with

$$\begin{aligned} -\widehat{G} - \lambda \widehat{C} &:= \widehat{U}(-G - \lambda C)\widehat{V}^{-1} \\ &:= \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \left( \begin{bmatrix} -G_0 & W_G \\ 0 & -G_\infty \end{bmatrix} - \lambda \begin{bmatrix} C_0 & W_C \\ 0 & C_\infty \end{bmatrix} \right) \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} \\ &= -\begin{bmatrix} G_0 & 0 \\ 0 & G_\infty \end{bmatrix} - \lambda \begin{bmatrix} C_0 & 0 \\ 0 & C_\infty \end{bmatrix}, \end{aligned}$$

where

$$U = \widehat{U} Q^T, \quad V^{-1} = Z \widehat{V}^{-1}$$

are the matrices required in (2.24).

### Balanced truncation

After decoupling the transfer function, we apply balanced truncation [BMS05] to the realization $[B_0, C_0, -G_0, L_0, F_0]$ of the proper part $H_0(s)$ of the transfer function. Furthermore, in the following section some knowledge about the controllability Gramian $P$ and the observability Gramian $Q$ of the descriptor system (2.11) as well as some knowledge about other topics in systems and control theory is required. For more information please refer to literature, e. g. [KFA69, MS05] or one of the numerous other textbooks. From here on we assume $C$ to be nonsingular as it is the case for $C_0$ from above.

**Definition 2.24 (Balanced realization)** *A realization $[B, C, -G, L, F]$ of the transfer function $H(s)$ is called* balanced *if*

$$P = Q =: \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix},$$

*where $P, Q$ are the controllability and the observability Gramian and $\sigma_1 \geq \ldots \geq \sigma_n$ are the Hankel singular values of the realization.*

For balanced truncation we compute the controllability and observability Gramians as solutions to the dual Lyapunov equations

$$(-G_0)^T \widehat{Q} C_0 + C_0^T \widehat{Q}(-G_0) + L_0^T L_0 = 0$$

with $Q := C_0^T \widehat{Q} C_0$ and

$$(-G_0) P C_0^T + C_0 P (-G_0)^T + B_0 B_0^T = 0.$$

Rather than computing $\widehat{Q}$ and $P$, we compute approximate factors $S$ and $R$ of low rank so that

$$P \approx SS^T, \qquad \widehat{Q} \approx RR^T.$$

Next we compute a singular value decomposition of the following form:

$$SR^T = [U_1 U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}. \tag{2.27}$$

Now we set

$$T_l = \Sigma_1^{-1/2} V_1^T R C^{-1}$$

and

$$T_r = S^T U_1 \Sigma_1^{-1/2}$$

with $\Sigma_1 = \mathrm{diag}(\sigma_1, ..., \sigma_k)$ and $\Sigma_1^{-1/2} = \mathrm{diag}(\frac{1}{\sqrt{\sigma_1}}, ..., \frac{1}{\sqrt{\sigma_k}})$ so that we are able to compute the reduced system with

$$(\tilde{C}, -\tilde{G}, \tilde{B}, \tilde{L}, \tilde{F}) = (T_l C_0 T_r, T_l(-G_0)T_r, T_l B_0, L_0 T_r, L_\infty G_\infty^{-1} B_\infty). \tag{2.28}$$

Here, $L_\infty G_\infty^{-1} B_\infty = \tilde{F}$ corresponds to the infinite part of the original system in the reduced system. Remember that $F_0$ is assumed to be a zero matrix. Note, that this is the square root variant of balanced truncation. It is known that the so computed reduced order model has a computable error bound, for details see Chap. 3.

### Remarks

While working with the algorithm of balanced truncation, solutions to the following problems were found, which can be used in the future as well.

One problem was that the system matrices $B, C, G$ and $L$ are overdetermined because of the modeling with the help of the Kirchhoff equations. For that reason, our matrix pencil could be singular. This causes serious problems for all MOR algorithms. A solution is to delete one row and/or column of the system realization (the one corresponding to the so called reference node) before starting MOR. The information of these data is considered later after the MOR process again.

Another problem was that the matrix $G$ could be singular because of cutting out the linear subcircuit. It occurred that some boundary nodes had no direct connection to the rest of the subcircuit anymore, leading to the following matrix structure,

$$G = \begin{pmatrix} G_2 + G_3 & -G_2 & 0 & -G_3 \\ -G_2 & G_1 + G_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -G_3 & 0 & 0 & G_3 \end{pmatrix}.$$

The Disk Function was not working correct and other problems were caused as well. A solution was to add a small value $\varepsilon$ to the entries of the main diagonal for the boundary nodes. Here we choose $\varepsilon = 10^{-9}$, so we get

$$G = \begin{pmatrix} G_2 + G_3 & -G_2 & 0 & -G_3 \\ -G_2 & G_1 + G_2 & 0 & 0 \\ 0 & 0 & \varepsilon & 0 \\ -G_3 & 0 & 0 & G_3 + \varepsilon \end{pmatrix}.$$

Seen as a physical element, one can imagine that this is a huge resistance of $\varepsilon^{-1}$ Ohm. The influence to the solution is nearly zero, but the elements are connected and $G$ is regular.
This technological solution can be replaced by a mathematical one:
We apply an MOR approach to

$$(-G - \mu I, C), \quad \mu \in \mathbb{R} \text{ with } \mu > 0$$

and compute the reduced order model. In case of balanced truncation, we apply the obtained projection matrices $T_r, T_l$ to the original, unshifted version. This guarantees the usual mathematical properties (except in some cases the computable error bound) of this method and preserves the "disconnection" represented by zero poles in the reduced order model.

# 3 Matrix decomposition for MOR via low rank approximation

In this chapter we want to explain how matrix decomposition approaches can be useful for MOR.
The ideas we will present in this chapter are motivated by the problematic handling of linear descriptor systems resulting from the modeling of very large scaled integration (VLSI) ICs. MOR here also operates with parasitic linear subcircuits but here there is something special. Since the number of terminals, i.e., the number of connections of the subcircuit to the circuit itself, is very large, new approaches are required.
After outlining the problem in general and a solution path, several possible approaches based on matrix decomposition will be considered. This results in the introduction of a method combining the general approach, a chosen matrix decomposition approach, and an established MOR method explained in Sec. 2.3. At last, we analyze possible error sources and derive error bounds.

## 3.1 MOR in VLSI design

Suppose the transfer function (2.13)

$$H(s) = L(sC+G)^{-1}B,$$

with $C, G \in \mathbb{R}^{n\times n}$, $L \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times m}$ is given.

Note that we assume the number of inputs $m$ to be equal to the number of outputs. In general, that is not the case but in many VLSI examples it is. By means of (2.11), it is easy to see that the information about the I/O-ports of the subcircuit is included in the matrix $L$ and matrix $B$. The assumption, that $m$ is much smaller than $n$, is not valid anymore. In contrast to standard MOR methods, see Sec. 2.3, the sizes of $L$ and $B$ need to be reduced as well to allow efficient simulation. This leads to more compact models.

The aim is to find a projection of the $(m\times m)$-transfer function $H(s)$ onto a $(p\times p)$-transfer function $H_r(s)$. We achieve this by means of a decomposition of the transfer function such that $H(s)$ is closely approximated by

$$\widehat{H}(s) = V_L \underbrace{\left[W_L^T L(sC+G)^{-1}BV_B\right]}_{:=H_r(s)} W_B^T$$

with $V_L$, $W_L$, $V_B$, $W_B \in \mathbb{R}^{m\times p}$.
In addition to requiring projector properties, i.e.,

$$W_L^T V_L = I_p \qquad \text{and} \qquad W_B^T V_B = I_p,$$

it should hold that $m \gg p$ so that a reasonable reduction of the transfer function is achieved.

The so-obtained internal transfer function

$$H_r(s) = W_L^T L(sC+G)^{-1} B V_B$$

is, as requested, the projection of the $(m\times m)$-transfer function onto a $(p\times p)$-transfer function. This internal transfer function $H_r(s)$ can be further reduced with any method for model reduction of linear descriptor systems.

Recent studies have shown that we can make use of a large degree of correlation between the plurality of input and output terminals. The next subsections introduce the SVD based method SVDMOR [FL04, TH07] as well as an extended version of SVDMOR, the so called ESVDMOR [LTYM06, LTYM08, TH07], which will be the foundation for our work. Note that those methods have been criticized affecting their accuracy [TH07]. Also, the correlation between the port-reduced system and the original system may not be good in terms of timing and delay. For further information about alternatives, e. g. the *TermMerg* algorithm, please refer to [LTL$^+$05, TH07]. This could play a role in future research as well, see Chap. 5.

Nonetheless, the ESVDMOR method is well suited for our investigations, because the used moment matrix gives good terminal correlations in our industrial interconnect benchmark circuits.

### 3.1.1 SVDMOR

In this subsection, we discuss a method which was proposed recently and which gives us a basis for reducing the terminals of an IC, the SVDMOR [FL04]. As suggested by its name, the singular value decomposition (SVD) is a main part of this method. SVD will be explained in more detail in Def. 3.2. In Sec. 3.2, we will show some alternative ideas to SVD in conjunction with MOR.

Starting point is the given transfer function

$$H(s) = L(sC+G)^{-1}B.$$

The matrix $H(s)$ is supposed to have highly dependent entries, which means it should be possible to approximate $H(s)$ with a matrix of lower rank. We want to know some information about circuit response correlations. Many terminals are not independent in terms of their timing information, which can be reflected in their frequency domain moments.

**Definition 3.1 (The i-th block moment)** *Consider a linear descriptor system (2.11) with its transfer function (2.13). The $i$-th* block moment *of the system is defined as*

$$\mathbf{m_i} = L(-G^{-1}C)^i G^{-1}B,$$

*which is an $m \times m$ matrix function.*

**Remark 3.1** *The block moment* $\mathbf{m_i}$ *can be computed recursively. For $i > 1$,*

$$\begin{array}{rclrcl} \mathbf{a_0} & = & G^{-1}B; & \mathbf{m_0} & = & L\mathbf{a_0} \\ \mathbf{a_1} & = & -G^{-1}C\mathbf{a_0}; & \mathbf{m_1} & = & L\mathbf{a_1} \\ \vdots & & & \vdots & & \\ \mathbf{a_i} & = & -G^{-1}C\mathbf{a_{i-1}}; & \mathbf{m_i} & = & L\mathbf{a_i}. \end{array}$$

In general, there are a lot of choices to acquire the needed information. The DC response matrix $H_{DC} = \mathbf{m_0}$, the first moment of the response

$$\mathbf{m_1} = LG^{-1}CG^{-1}B,$$

frequency shifted moments

$$\mathbf{m_{s_0}} = L(G + s_0C)^{-1}B, \tag{3.1}$$

or even combinations of these are practical approaches.
Although different matrix factorization approaches are explained in more detail in Sec. 3.2, we want to introduce certain aspects of the SVD [GK65, GR70, CGO07] here in order to enable the reader to follow the SVDMOR approach.

**Definition 3.2 (The singular value decomposition (SVD))** *Consider $A \in \mathbb{R}^{m \times n}$ with $m \geq n$. The triple factorization*

$$A =: U\Sigma V^T$$

*is called* singular value decomposition*, if*

$$U^TU = V^TV = VV^T = I_n \quad \textit{and} \quad \Sigma = \operatorname{diag}(\sigma_1, ..., \sigma_n).$$

*The matrix $U \in \mathbb{R}^{m \times n}$ consists of $n$ orthonormalized eigenvectors associated with the $n$ largest eigenvalues of $AA^T$, and the matrix $V \in \mathbb{R}^{n \times n}$ consists of the orthonormalized eigenvectors of $A^TA$. The columns of $U$ are called the* left singular vectors*, the columns of*

*$V$ are called the* right singular vectors. *The diagonal elements of $\Sigma \in \mathbb{R}^{n \times n}$ are the non-negative square roots of the eigenvalues of $A^T A$. They are called* singular values. *We shall assume that*

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0.$$

*Thus if* $\operatorname{rank}(A) = r$, $\sigma_{r+1} = \sigma_{r+2} = \ldots = \sigma_n = 0$.

Without loss of generality, we choose the DC response matrix $H_{DC}$ for explaining SVD-MOR. Let us assume that we have a regular circuit and we expect the response to be highly correlated. We perform the approximation with the help of an SVD. Later, we consider other approaches as well. We get

$$H_{DC} = LG^{-1}B = U\Sigma V^T,$$

where $\Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_m)$.
To obtain an efficient reduction, we only keep the important singular values. Thus, if the number $r$ of dominant singular values is small, i.e. $\sigma_1, \ldots, \sigma_r$ with $r \ll m$, then the error of setting the rest of the singular values equal to zero will also be small. In (3.2) one can see that $U_r$ and $V_r^T$ consist of the singular vectors referring to the dominant singular values $\sigma_i$, $i = 1, \ldots, r$. For details in error estimation, see Sec. 3.3. Hence, we get

$$U\Sigma V^T \approx U_r \Sigma_r V_r^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T. \tag{3.2}$$

**Definition 3.3 (The Moore-Penrose pseudoinverse [GV96, Dem97])** *Suppose that matrix $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and* $\operatorname{rank}(A) = n$. *Furthermore,*

$$A = QR = U\Sigma V^T$$

*are the QR decomposition and SVD of A, respectively. Then*

$$A^+ := (A^T A)^{-1} A^T = R^{-1} Q^T = V \Sigma^{-1} U^T$$

*is called the* (Moore-Penrose) pseudoinverse *of A.*
*If $m < n$,* $\operatorname{rank}(A) = m$, *then $A^+ := A^T (AA^T)^{-1}$. If $m = n =$* $\operatorname{rank}(A)$, *then $A^+ := A^{-1}$.*

**Remark 3.2** *Typically, $A^+$ is defined to be the unique matrix $X \in \mathbb{R}^{n \times m}$ that satisfies the four* Moore-Penrose conditions*:*

$$\begin{array}{llllllll} (i) & AXA & = & A & (iii) & (AX)^T & = & AX \\ (ii) & XAX & = & X & (iv) & (XA)^T & = & XA. \end{array}$$

Using this definition, we approximate

$$L \approx U_r L_r, \qquad B \approx B_r V_r^T, \tag{3.3}$$

where $B_r$ and $L_r$ result from applying the Moore-Penrose pseudoinverse of $U_r$ and $V_r^T$, respectively, to $L$ and $B$. Moreover, we bear in mind that $U_r$ and $V_r$ are orthogonal matrices, i.e. $U_r^T U_r = I_r$ and $V_r^T V_r = I_r$. Thus, we get

$$L_r = U_r^+ L = (U_r^T U_r)^{-1} U_r^T L = U_r^T L \tag{3.4}$$

and

$$B_r = B V_r^{T+} = B V_r (V_r^T V_r)^{-1} = B V_r. \tag{3.5}$$

Now, the transfer function becomes

$$H(s) \approx \widehat{H}(s) = U_r L_r (G + sC)^{-1} B_r V_r^T.$$

This leads to a new transfer function

$$H_r(s) = L_r (G + sC)^{-1} B_r, \tag{3.6}$$

which can be reduced with the help of the established standard MOR methods. If $\tilde{H}_r(s)$ is the result of any MOR method, then the approximation of the whole transfer function is

$$H(s) \approx U_r \tilde{H}_r(s) V_r^T.$$

### 3.1.2 Extended-SVDMOR

Motivated by the fact that the SVDMOR algorithm has some drawbacks in terms of accuracy in cases where the difference between the number of inputs and the number of outputs is large as well as the presence of some passivity preserving problems, an extended version of SVDMOR, the so called ESVDMOR, was developed [LTYM06, LTYM08]. The main idea here is to perform the low-rank approximation separately for input and output terminals. The use of information of higher order moments ensures the accuracy of the reduced model.

To show all the advantages of this method, we assume that there is a difference between the number of inputs $m_{in}$ and the number of outputs, here $m_{out}$. To show how to combine the block moment information, consider the $i$-th block moment of a general linear system

$$\mathbf{m_i} = \begin{bmatrix} m^i_{1,1} & m^i_{1,2} & \cdots & m^i_{1,m_{in}} \\ m^i_{2,1} & m^i_{2,2} & \cdots & m^i_{2,m_{in}} \\ \vdots & \vdots & \ddots & \vdots \\ m^i_{m_{out},1} & m^i_{m_{out},2} & \cdots & m^i_{m_{out},m_{in}} \end{bmatrix},$$

where $\mathbf{m_i}$ is a $m_{out} \times m_{in}$ matrix. Here, column $j$ of $\mathbf{m_i}$ represents the moment vector of all output terminals due to the input terminal $j$ and row $k$ of $\mathbf{m_i}$ represents the moment vector of output terminal $k$ due to all input terminals. Since we planned on using the information of higher order moments, we need to define the following new matrices.

**Definition 3.4 (The input response matrix $M_I$)** *Consider the i-th block moment* $\mathbf{m_i}$ *as defined in Def. 3.1. The $rm_{out} \times m_{in}$ matrix $M_I$ of the form*

$$M_I = \begin{bmatrix} \mathbf{m_0} \\ \mathbf{m_1} \\ \vdots \\ \mathbf{m_{r-1}} \end{bmatrix}$$

*is called the* input response matrix.

**Definition 3.5 (The output response matrix $M_O$)** *Consider the i-th block moment* $\mathbf{m_i}$ *as defined in Def. 3.1. The $rm_{in} \times m_{out}$ matrix $M_O$ of the form*

$$M_O = \begin{bmatrix} \mathbf{m_0}^T \\ \mathbf{m_1}^T \\ \vdots \\ \mathbf{m_{r-1}}^T \end{bmatrix}$$

*is called the* output response matrix.

In both matrices, the column vectors represent the moment series for the outputs referring to the inputs, respectively. The rows in both matrices lose the terminal-related physical information but they represent the different orders of moments. Now, SVD is performed as a low rank approximation to both,

$$M_I = U_I \Sigma_I V_I^T \approx U_{I_{r_i}} \Sigma_{r_i} V_{I_{r_i}}^T, \tag{3.7}$$

$$M_O = U_O \Sigma_O V_O^T \approx U_{O_{r_o}} \Sigma_{r_o} V_{O_{r_o}}^T, \tag{3.8}$$

where

- $\Sigma_{r_i}$ is an $r_i \times r_i$ diagonal matrix
- $\Sigma_{r_o}$ is an $r_o \times r_o$ diagonal matrix
- $V_{I_{r_i}}^T$ is an $r_i \times p$ matrix
- $V_{O_{r_o}}^T$ is an $r_o \times q$ matrix
- $r_i$ and $r_o$ are the numbers of significant singular values.

Consequently, the analogous steps as in the SVDMOR approach are performed. We also approximate the matrices $B$ and $L$ resulting in

$$B \approx B_r V_{I_{r_i}}^T$$

and

$$L \approx V_{O_{r_o}} L_r,$$

where $B_r$ and $L_r$ are again the results of applying the Moore-Penrose pseudoinverse of $V_{I_{r_i}}^T$ and $V_{O_{r_o}}$ (which are orthogonal again), respectively, to $B$ and $L$. In detail, we have

$$B_r = BV_{I_{r_i}}^{T+} = BV_{I_{r_i}}(V_{I_{r_i}}^T V_{I_{r_i}})^{-1} = BV_{I_{r_i}}$$

and

$$L_r = V_{O_{r_o}}^+ L = (V_{O_{r_o}}^T V_{O_{r_o}})^{-1} V_{O_{r_o}}^T L = V_{O_{r_o}}^T L,$$

where $B_r \in \mathbb{R}^{n \times r_i}$ and $L_r \in \mathbb{R}^{r_o \times n}$.

This yields the approximation of the transfer function

$$H(s) \approx \widehat{H}(s) = V_{O_{r_o}} L_r (G + sC)^{-1} B_r V_{I_{r_i}}^T$$

A new “internal” transfer function is

$$H_r(s) = L_r (G + sC)^{-1} B_r, \tag{3.9}$$

which is reduced with a standard MOR algorithm to $\tilde{H}_r(s)$. Again, the final result is a very compact terminal and order reduced model

$$H(s) \approx V_{O_{r_o}} \tilde{H}_r(s) V_{I_{r_i}}^T$$

However, several drawbacks exist. One of them is the computational cost needed by the SVD. In the following Sec. 3.2, alternative ideas are proposed. Another problem is preserving passivity. If the inputs to the original model consist of current and voltage sources, problems can arise. In this case neither the SVDMOR nor the ESVDMOR approach ensures passivity. That is a big problem and we mention this also in Chap. 5. For more information see [LTYM08].

## 3.2 Matrix decomposition approaches

In the previous section we presented a general way to handle problems of VLSI circuits with a large number of terminals. Unfortunately, the use of an SVD has a huge drawback which prevents the application of the idea of ESVDMOR for really large-scale applications like those arising in power grid models. We want to explain approaches to replace the SVD as a method for decomposing the output and the input response matrices $M_O$ (3.8) and $M_I$ (3.7) because it is too expensive to use the SVD there. As an example, suppose we have a matrix with dimension $n = 10^6$ and a modern CPU with 3 GHz. An SVD needs about $22n^3$ flops. In our example that would be $22 \cdot 10^{18}$ flops. That means we would need a total CPU time of approximately 230 years. Obviously, this is way too long.

In general, a matrix factorization (or decomposition) is a right hand side product of the form

$$A = F_1 F_2 \cdots F_k.$$

In most cases $k = 2$ or $k = 3$.

Recall that the reason for the decomposition is the approximation of a matrix by a matrix of lower rank. So our problem requires the knowledge of just the largest singular values which leads to the idea to use a Truncated SVD. Later in Sec. 3.2.2, another, non SVD based alternative is introduced, which could be a possibility for the future.

### 3.2.1 Truncated SVD

In this section, we want to explain one of many alternative variations of the SVD, the so called truncated singular value decomposition (TSVD). Our goal is a low rank approximation of a matrix. Without loss of generality, we choose

$$H_{DC} = LG^{-1}B.$$

That means we want to compute a few dominant singular values, so the motivation to use the TSVD is the saving of computing power and storage space as well as time efficiency. The difference to the SVD is that we do not compute the hole decomposition

$$H_{DC} = U\Sigma V^T,$$

and then set the most of the singular values equal to zero. Here we just compute the singular values and the singular vectors we need, so that

$$H_{DC} \approx U_r \Sigma_r V_r^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T. \tag{3.10}$$

Recall that $r$ denotes the number of significant singular values and vectors. We do not know that number so we specify it depending on the error tolerance of the approximation. We show in Sec. 3.3 that we need a useful, that means a rapidly decreasing, spectrum formed by the singular values $\sigma_i$. Then it is possible to approximate $H_{DC}$ by a matrix of much lower rank making just a small and acceptable error.

There are several algorithms computing a TSVD like [CH90, Sto08]. We do not know which method is the most suitable for our problem. At the moment, the MATLAB function `svds` is used. It changes the problem of computing some singular values of matrix $H_{DC}$ into the problem of computing several eigenvalues of a matrix

$$A = \begin{pmatrix} 0 & H_{DC} \\ H_{DC}^T & 0 \end{pmatrix}. \tag{3.11}$$

It can be shown that the positive eigenvalues of $A$ are equal to the non-negative square roots of the eigenvalues of $H_{DC}^T H_{DC}$, and those square roots are equal to the singular values of $H_{DC}$.

For computing the eigenvalues of (3.11), another MATLAB function called `eigs` is used. It computes the largest magnitude eigenvalues of $A$ ordered by size and it works with the help of implicitly restarted Arnoldi methods. For detailed information please see [Sor92, LS96, LSY98].

### 3.2.2 Non SVD based approaches

In this section we explain what hides behind the pivoted QR factorization (PQR), the semi pivoted QR factorization (SPQR) with the help of the quasi-Gram-Schmidt method and finally the sparse column-row approximation (SCRA). All these methods do not alter the matrix we want to decompose, which we want for the sake of convenience denote as $A$. These methods only require the formation of matrix-vector products. The idea to use the SCRA in a MOR application is a new one. Before we get to the SCRA, we provide some details of the underlying theory [BPS05, Ste99].

#### PQR factorization

One alternative to the SVD is the pivoted QR factorization (PQR). It gives comparable results to the SVD and it has the form

$$AP = QR,$$

where without loss of generality $m \geq n$, and

- $A \in \mathbb{R}^{m \times n}$ is the matrix to decompose,
- $P \in \mathbb{R}^{n \times n}$ is a permutation matrix,
- $Q \in \mathbb{R}^{m \times n}$ is orthonormal,

- and $R \in \mathbb{R}^{n\times n}$ is upper triangular.

In the following, let $B = AP$.
A rank $k$ approximation to $A$ can be obtained by partitioning the PQR factorization

$$\left( \begin{array}{cc} B_1^{(k)} & B_2^{(k)} \end{array} \right) = \left( \begin{array}{cc} Q_1^{(k)} & Q_2^{(k)} \end{array} \right) \left( \begin{array}{cc} R_{11}^{(k)} & R_{12}^{(k)} \\ 0 & R_{22}^{(k)} \end{array} \right), \tag{3.12}$$

where $k$ is the number of columns of $B_1^{(k)}$. We use $k$ instead of $r$ to avoid any possibility of confusion with $r$ as a column of matrix $R$.
The approximation is

$$\widetilde{B}^{(k)} = Q_1^{(k)} \left( \begin{array}{cc} R_{11}^{(k)} & R_{12}^{(k)} \end{array} \right),$$

that means we delete the columns $Q_2^{(k)}$ of $Q$ and the rows of $R$ that include $R_{22}^{(k)}$. The approximation does not change the dimension,

$$\dim(B) = \dim(\widetilde{B}^{(k)}).$$

Describing this in terms of $A$ leads to

$$\widetilde{A}^{(k)} = Q_1^{(k)} \left( \begin{array}{cc} R_{11}^{(k)} & R_{12}^{(k)} \end{array} \right) P^T. \tag{3.13}$$

An efficient algorithm does not compute the whole decomposition (3.12). It selects columns of $A$ and iteratively computes additional columns of $Q$ and rows of $R$. The $k$-th step of this iteration leads to (3.13). The selection of the columns of $A$ is called pivoting. The permutation matrix $P$ determines the order of selecting the columns such that the error of the approximation is small. Actually, the error analysis of this approach is an interesting point, see Sec. 3.3.

### SPQR factorization

An extension of the PQR factorization is the semi-PQR (SPQR) factorization. We explain the quasi-Gram-Schmidt method because it is the base for extending PQR.

Let

$$B = QR \tag{3.14}$$

be a normal QR factorization. We want to compute

$$\left( \begin{array}{cc} B & a \end{array} \right) = \left( \begin{array}{cc} Q & q \end{array} \right) \left( \begin{array}{cc} R & r \\ 0 & \rho \end{array} \right), \tag{3.15}$$

in which $a$ is an additional column to $B$, $q$ is an additional column to $Q$ and $(r, \rho)^T$ is an additional column to $R$. The scalar $\rho$ is the last entry in this column of $R$ and at the same time the last and only entry of an additional row $(0,\rho)$ of $R$. The last column of (3.15) obviously gives

$$a = Qr + \rho q.$$

Since $Q$ is orthonormal and $\begin{pmatrix} Q & q \end{pmatrix}$ should be, i. e. $Q^T Q = I_n$ and $Q^T q = 0$, we have

$$Q^T a = Q^T Qr + Q^T \rho q = r. \tag{3.16}$$

Finally,

$$q = \rho^{-1}(a - Qr) \tag{3.17}$$

and if $||q|| = 1$ then

$$\rho = \|a - Qr\|. \tag{3.18}$$

**Remark 3.3** *Note that here* $\|\,.\,\|_2$ *is the vector 2-norm or Euclidean norm, defined by*

$$\|x\|_2 = \sqrt{|x_1|^2 + \ldots + |x_n|^2}.$$

We are able to compute a PQR factorization of $A$ by selecting columns of $A$ and updating the QR factorization of $B$ with the help of the quasi-Gram-Schmidt method, see (3.16), (3.17), and (3.18). Consider $a$ as the first column selected of the matrix $A$.
We set

$$R_{11}^{(1)} = \rho = \|a\|,$$

and

$$Q_1^1 = \rho^{-1} a.$$

Hence, we can compute

$$B_1^{(k)} = Q_1^{(k)} R_{11}^{(k)}$$

step by step.

But there are also drawbacks. The Gram Schmidt method is known for numerical problems while computing $q$. Cancellations in the formation of $a - Qr$ lead to a $q$ which is not orthogonal to the columns of $Q$. An intermediate step for reorthogonalization has to be done.
We also just compute $R_{11}^{(k)}$. Calculating $R_{12}^{(k)}$ is possible but it could be expensive. The $k$-th column of $R_{12}^{(k)}$ is computable from the $k$-th column of $Q_1^{(k)}$ and the $(n-k)$ columns of $A$ that are not in $B_1^{(k)}$.
For large dimensions, it is possible that we are not able to store the matrix $Q$. In that case, recall the following formulation of (3.14),

$$Q = BR^{-1},$$

and consequently

$$\begin{pmatrix} Q & q \end{pmatrix} = \begin{pmatrix} B & a \end{pmatrix} \begin{pmatrix} R & r \\ 0 & \rho \end{pmatrix}^{-1}.$$

We use (3.17) and the quasi-Gram-Schmidt algorithm, where $q$ is computed in the following way,

$$q = \rho^{-1}(a - BR^{-1}r).$$

Now we can apply the quasi-Gram-Schmidt step to the columns of $A$. We get a pivoted, Q-less PQR factorization of $A$, which we call a semi-PQR (SPQR) factorization. The corresponding approximation

$$\widetilde{B}^{(k)} = \begin{pmatrix} B_1^{(k)} & R_{11}^{(k)-1} \end{pmatrix} \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \end{pmatrix}$$

is called the SPQR approximation.
This factorization could be a good alternative to the TSVD as a low rank approximation. We will discuss that later in Chap. 4 and Chap. 5.

### SCR approximation

If the matrix $A$ is nearly square, and this could easily be the case if we think of the (E)SVDMOR approach, the storage of $R$ in SPQR becomes a problem. The solution to this problem lies in using another factorization, the so called sparse column-row approximation (SCRA).

At first we apply the quasi-Gram-Schmidt algorithm to the columns of $A$. We will get a set of orthonormalized columns $X$ of $A$ and an upper triangular matrix $R$ corresponding to $R_{11}$, like

$$AP = XR. \tag{3.19}$$

Now we do the same with $A^T$ and similarly we get a set of orthonormalized rows $Y^T$ and another upper triangular matrix $S$,

$$A^T\hat{P} = Y^T S. \tag{3.20}$$

Now one can find a matrix $T$, such that

$$A \approx XTY^T, \tag{3.21}$$

and

$$\|A - XTY^T\|_2 = min. \tag{3.22}$$

**Remark 3.4** $||\,.\,||_2$ *denotes the* matrix 2-norm *or* spectral norm*, defined by*

$$||A||_2 = \sup_{x\neq 0} \left|\left| A\left(\frac{x}{||x||_2}\right)\right|\right|_2 = \max_{||x||_2=1} ||Ax||_2.$$

*Remember the vector 2-norm or Euclidean norm of Rem. 3.3.*

The approximation (3.21) is called the sparse column-row approximation (SCRA). The right choice of $T$ for the minimization of (3.22) is

$$T = R^{-1}R^{-T}(X^TAY)S^{-1}S^{-T}, \tag{3.23}$$

with matrix $R$ of (3.19) and $S$ of (3.20), see [Ste99]. An error analysis is shown in Sec. 3.3. To use the SCRA as an alternative method for TSVD in MOR approaches, one can rewrite $XTY^T$ in the form

$$XTY^T = (XR^{-1})(R^{-T}(X^TAY)S^{-1})(S^{-T}Y^T) \equiv JWK^T.$$

If we now compute the SVD of

$$W = M\Sigma N^T \tag{3.24}$$

and set

$$U = JM \quad \text{and} \quad V = KN,$$

we get the relation

$$XTY^T = U\Sigma V^T.$$

Although we do not compute $U$ and $V$ explicitly, that means we keep the factored form $U = XR^{-1}M$ and $V = YS^{-1}N$, it is not clear if this is useful. On the one side, we do not exactly know if the approximation is good enough, on the other side we also have to perform an SVD while decomposing $W$.

## 3.3 The approximation error

In MOR, error estimation is very important but also very difficult. Using the ESVDMOR approach, the error committed while reducing the number of terminals as well as the error which accrues while performing the established MOR method has to be analyzed.

At first we want to analyze the terminal reduction errors. This of course depends on the chosen factorization.

### 3.3.1 The error of SVD

In (3.2), (3.7), and (3.8) we use the SVD. The error of this approximation is given by the following theorem:

**Theorem 3.1** [GV96] *Let $A \in \mathbb{R}^{m\times n}$ be a matrix with $m \geq n$. If* $\operatorname{rank}(A) \geq k$ *and*

$$A = U\Sigma V^T \approx U_k\Sigma_k V_k^T = A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

*are the decomposition of Def. 3.2 and the approximation performed in (3.2) and (3.10), then*

$$\min_{\operatorname{rank}(B)=k} \|A-B\|_2 = \|A-A_k\|_2 = \sigma_{k+1}.$$

Theorem 3.1 states that we can compute the error for the approximation and that the approximation is optimal with respect to the spectral norm.

### 3.3.2 The error of (S)PQR

If we choose the (S)PQR method, we have to determine $R_{22}^{(k)}$. Recall that the factorization (3.12) leads to the approximation

$$\widetilde{B}^{(k)} = Q_1^{(k)} \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \end{pmatrix}.$$

We recognize that

$$B - \widetilde{B}^{(k)} = Q_2^{(k)} ( \begin{array}{cc} 0 & R_{22}^{(k)} \end{array} ),$$

and we know that $Q_2^{(k)}$ is orthonormal. So the error is

$$\|B-\widetilde{B}^{(k)}\| = \|R_{22}^{(k)}\|,$$

and reformulating this in terms of $A$ we get

$$\|A-\widetilde{A}^{(k)}\| = \|R_{22}^{(k)}\|.$$

This norm is returned, for instance, by algorithms for the rank-revealing QR factorization [BQO98]. In the algorithm described in [BPS05], the norm $\|R_{22}^{(k)}\|$ is a by-product and we can get it almost for free in every step. Thus it can be used to compute error bounds or estimates.

### 3.3.3 The error of SCRA

If we choose the SCRA, we get a combination of the errors above. As shown in Sec. 3.2.2, we perform the SPQR twice. The error we made in the first factorization (3.19) is denoted by $\varepsilon_{col}$ and the one of (3.20) is denoted by $\varepsilon_{row}$. [Ste99] shows that for T of (3.23)

$$||A - XTY^T||^2 \leq \varepsilon_{col}^2 + \varepsilon_{row}^2.$$

Of course there is also an error if we decompose matrix $W$ in (3.24) with the help of the SVD.

### 3.3.4 The error of Balanced Truncation

In (3.6) and (3.9) a transfer function $H_r(s)$ is described, which is reduced with the help of the methods introduced in Sec. 2.3. One can say that there is no useable error bound for Krylov-subspace approaches. This makes balanced truncation methods, see Sec. 2.3.2, very interesting.

It is known that the computed reduced order model of (2.28) has a computable error bound

$$||H(i\omega) - \tilde{H}(i\omega)||_2 \leq \delta = 2 \sum_{j=r+1}^{n} \sigma_j,$$

almost everywhere in $\omega = [-\infty, \infty]$. The $\sigma_j$ in the sum above represent the neglected singular values contained in $\Sigma_2$ of (2.27).

# 4 Numerical examples

Based on the fact that this thesis is of more theoretical nature, the chapter about numerical examples is not as extensive. We want to introduce an algorithm using the methods of Chap. 3, which is not as efficient and fast as desired yet. The main motivation is to show that results for computing some academic as well as industrial benchmarks lead to very useful conclusions. Note that we are still learning in this area of research to reduce terminals while simulating electrical large scale ICs. Every new piece of knowledge is important and is integrated into future research, see Chap. 5.

## 4.1 A MATLAB implementation

Inputs are the system matrices $C$, $G$ ,$B$ and $L$ of (2.11). We work in the frequency domain, i. e. we want to know the frequency response which gives the relation of an input $u(t)$ to the output $y(t)$. We are using a computer with an Intel Pentium(R) 4 CPU with 2.81 GHz, 1 GB DDR-RAM, and Windows XP Professional Version 2002 SP2. Our MATLAB version is 7.6.0.324 (R2008a).

### 4.1.1 Computation of the moments of the transfer function

In fact, we use the (E)SVDMOR method, which requires at first the computation of the moments of the transfer function. Like mentioned in Rem. 3.1, we do this explicitly in a numerical unstable way with the help of standard MATLAB functions.

```
a0 = G\B;
m0 = L*a0;

GinvC = G\C;

a1 = GinvC*a0;
m1 = L*a1;

a2 = GinvC*a1;
m2 = La2;

a3 = ...
```

Significant improvements with the implementation of efficient algorithms from the research area numerical linear algebra could be achieved. Recall that we deal with system matrices of enormous size. We will mention this later in Chap. 5.

Also frequency shifted moments $\mathbf{m_{s0}}$, see (3.1), for testing the SVDMOR approach are implemented. The example shows a frequency $s_0 = 10^5$.

```
s0=1e5;

ms0= L/(G+(s0*C))*B;
```

After computing the particular moments, the input and the output response matrices $M_I$ and $M_O$ of the ESVDMOR approach, see Def. 3.4 and Def. 3.5, are built. This is quite simple to realize in MATLAB.

```
Mi = [m1;m2;m3];

Mo = [m1';m2';m3'];
```

### 4.1.2 Matrix decomposition and approximation

The next step is the key step in this algorithm. We decompose the chosen moment matrix, i. e. $\mathbf{m_0}$, $\mathbf{m_1}$, $\mathbf{m_{s_0}}$, etc. for SVDMOR or $M_I$ and $M_O$ for the ESVDMOR approach, with one of the decomposition methods in Sec. 3.2. Therefore, we use the MATLAB function `svds` which performs the truncated SVD, see Sec. 3.2.1.

A very important but also very difficult question is, how much singular values are significant and need to be computed to get a useful approximation. The example shows the computing of the largest 30 singular values of the DC response moment $H_{DC} = \mathbf{m_0}$. It is mentioned above, that it is not necessary to compute the moment explicitly. This will be part of future research. We have

```
[Ur,Sigmar,VrT] = svds(m0,30).
```

The next section will show that the decision about the number of needed singular values and their corresponding vectors depends heavily on the simulated IC and the resulting range of its singular values.

Computing the Moore-Penrose pseudoinverse, see Def. 3.3, of $U_r$ (3.4) and $V_r^T$ (3.5) leads to the approximation of $L$ and $B$ (3.3), so that we are able to compute the new system matrices $C$, $G$, $B_r$ and $L_r$ of $H_r(s)$, see (3.6). Thus,

| Name of the circuit | Number of nodes | Number of terminals | Percentage of terminals in % | Source |
|---|---|---|---|---|
| RC549 | 141 | 70 | 49.65 | Qimonda AG |
| RC590 | 220 | 65 | 29.55 | Qimonda AG |
| circuit3 | 3916 | 1905 | 48.65 | NEC Laboratories Europe,<br>IT Research Division |

Table 4.1: Overview of the chosen circuit examples.

```
Lr = Ur'*L;

Br = B*VrT.
```

Now we can clear the used memory except the memory for matrices $U_r$ and $V_r^T$, which is needed again later on. For ESVDMOR the algorithm works equivalently using the matrices $V_{I_{r_i}}^T$ and $V_{O_{r_o}}^T$ from the decomposition (3.7) and (3.8). Our algorithm ends by computing the terminal reduced transfer function again and taking a look at the results. In future work, a combination of this algorithm with a standard MOR method will be implemented, which is described in Chap. 5.

## 4.2 Examples and results

This section shows some results of the algorithm described in Sec. 4.1. We want to show three kinds of circuits, which are representative and lead to useful conclusions. Two of them are provided by the research group of TITAN, the analog in-house circuit simulator of the Qimonda AG. The third one is made available by the NEC Laboratories Europe, IT Research Division.

In the following, we want to give some details about the chosen circuit examples. Table 4.1 gives an overview. Please note that the information given here is limited to a few facts due of corporate secrecy.

- The first circuit is called *RC549*. It is a parasitic RC subcircuit of a much larger IC of a memory chip. The RC subcircuit is useful because it is very small but has

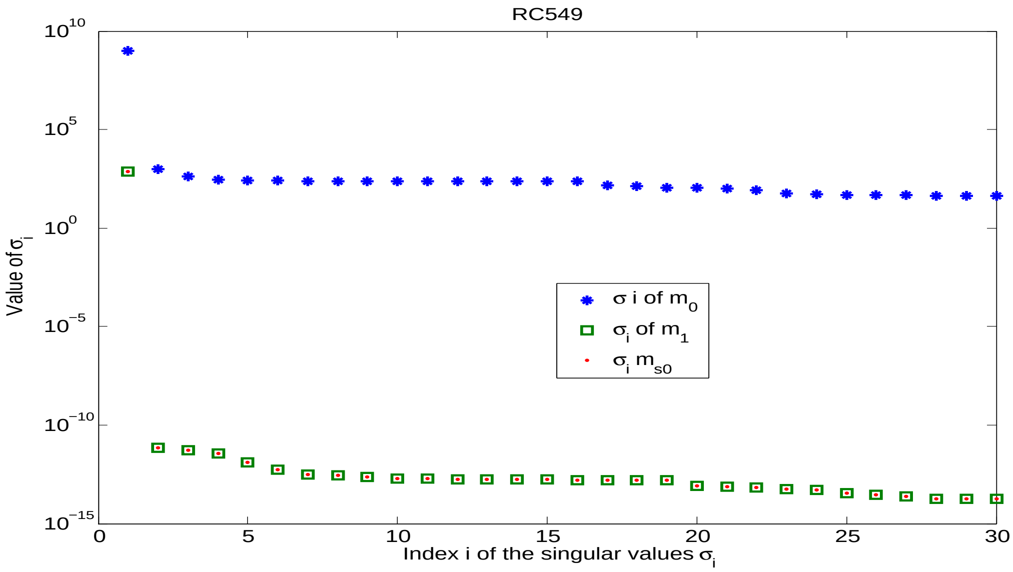

Figure 4.1: Range of the largest 30 singular values of *RC549*.

good attributes. The small number of nodes, i.e. $n = 141$, does not need much time for computation such that lot of tests can be performed in a short amount of time. However, the number of inputs and outputs with $m = 70$ is still nearly half of the nodes of the circuit. Now, tests can show how many pins could be reduced to reach the goal of an efficient reduction with an acceptable error.

- The second circuit is called *RC590*. It is like *RC549* one of the more than thousand parasitic subcircuits of the large memory chip IC. Both nearly have the same properties, but *RC549* has $m = 65$ terminals and $n = 220$ nodes altogether, i. e. the part of the pins is less than one third of all nodes.

- We are not allowed to give details about the third circuit (yet). It is larger than the others, rather $n = 3916$ and $m = 1905$. It is called *circuit3*.

Like mentioned in Sec. 4.1, it is important to know the range of the singular values in order to know how good the approximation will be. This can be achieved by using the MATLAB function `svds`, which supports the specification of an error tolerance.

Figure 4.1 shows the range of the singular values of *RC549*. To get an overview, we plotted the first 30 singular values. The range of $\mathbf{m_0}$ shows that the DC response matrix does not reveal the circuit response correlation in a useful way. The range of the singular values (SVs) of $m_1$ and $m_{s0}$ is much better, because it shows a big gap between the first and the second SV. Circuit *RC590* shows the same behavior, see Fig. 4.2. This encourages the

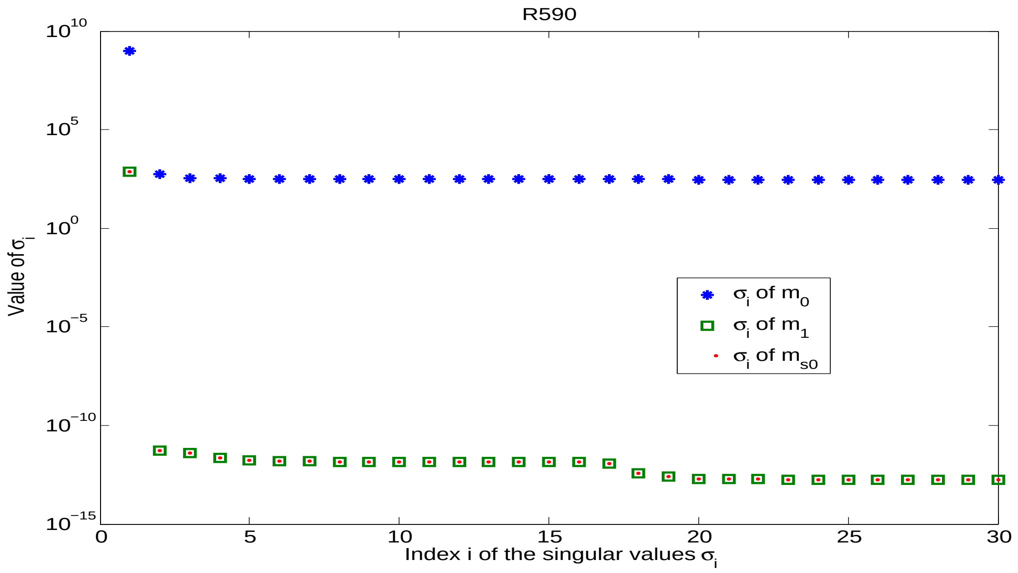

Figure 4.2: Range of the largest 30 singular values of *RC590*.

conclusion that the computing of $H_{DC}$ is not sufficient to determine the number $r$ of dominant singular values. Figure 4.3 shows the range of the first 500 SVs of $H_{DC}$ of *circuit3*. Unfortunately, there is no advanced analysis of *circuit3* available yet. Because of Fig. 4.1 and Fig. 4.2 we suspect that the range of higher order moments is useful for this example as well.

Recall that the result of the algorithm we implemented is a terminal reduced transfer function

$$H(s) \approx \hat{H}(s) = U_r H_r(s) V_r^T,$$

or

$$H(s) \approx \hat{H}(s) = V_{O_{r_o}} H_r(s) V_{I_{r_i}}^T.$$

Note that we have not reduced $H_r(s)$ with the help of further MOR methods to get $\tilde{H}_r(s)$. The results for the circuits *RC549* and *RC590* are shown in the following. Figure 4.4 and Fig. 4.5 show the analysis of the relative error

$$\varepsilon_{rel} = \frac{\|H(j\omega) - \hat{H}(j\omega)\|_2}{\|H(j\omega)\|_2}$$

depending of the frequency in the spectral norm. We recognize the similarity of the relative errors for both computed results. We expected this because of the similar structure of the circuits. ESVDMOR approach shows no advantages. The equality of the number of input

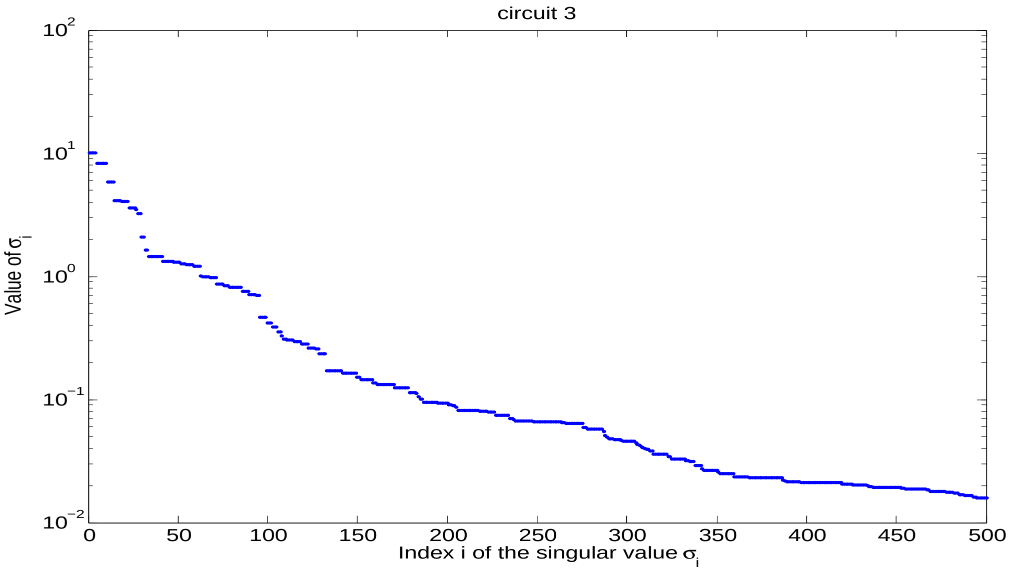

Figure 4.3: Range of the largest 500 singular values of *circuit3*.

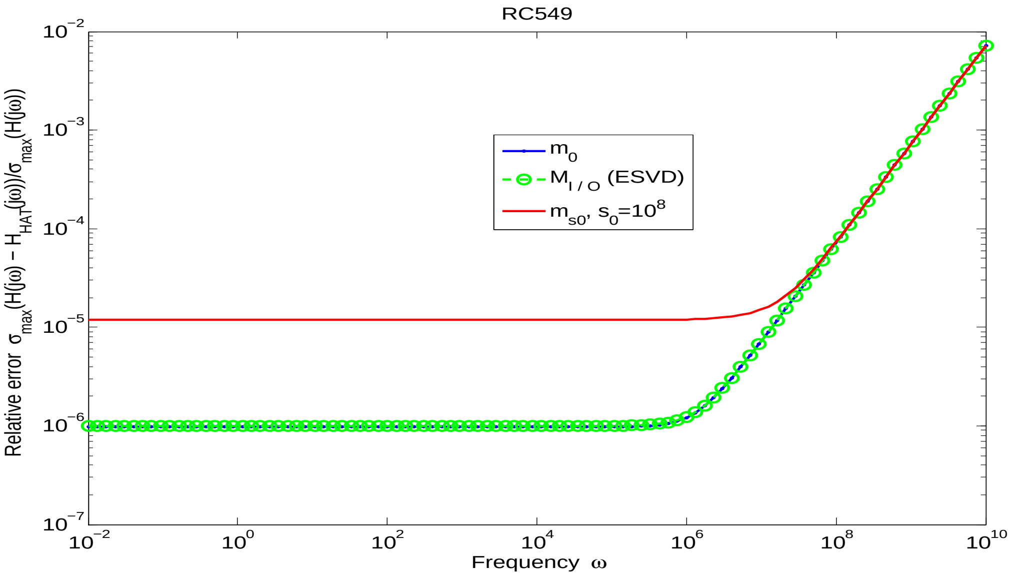

Figure 4.4: Relative error $\varepsilon_{rel}$ of *RC549* with the help of SVDMOR ($\mathbf{m_0}$, $\mathbf{m_{s_0}}$) and ESVDMOR ($M_{I/O}$).

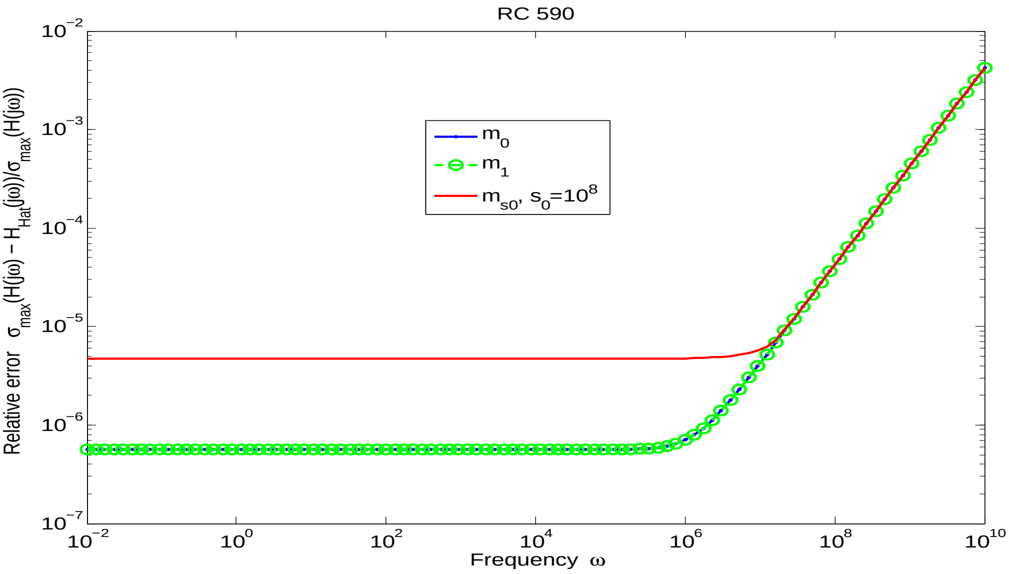

Figure 4.5: Relative error $\varepsilon_{rel}$ of *RC590* with the help of SVDMOR ($\mathbf{m_0}$, $\mathbf{m_1}$, $\mathbf{m_{s_0}}$).

and output pins is responsible for that. We reduced in both cases to *one* pin for input and output. The results are not as good as expected, but the degree of reduction in connection with the results satisfies. The error is increasing for higher frequency levels. However, this is not of importance and was expected since both subcircuits are part of a memory chip, which is working in low frequency ranges. Fig. 4.6 shows the absolute error

$$\varepsilon_{abs} = \|H(j\omega) - \hat{H}(j\omega)\|_2$$

for *RC590* in the the spectral norm. The particular feature here is the behavior of the absolute error of $\mathbf{m_{s0}}$. It drops around the frequency $s_0$, here $s_0 = 10^8$. This is consistent with the behavior of the relative error. The functional characteristics in Fig. 4.7, which shows the values of the transfer functions $H(s)$ and $\hat{H}(s)$ and their difference depending on the frequency in the spectral norm, show very good results.

In tests with *circuit3*, we reduced to $r = 30$ input and output pins. The results are very bad with an unacceptable error in the whole frequency range. However, it is possible that the approximation is not useful due to the bad range of the singular values in Fig. 4.3 and therefore the error we caused is the reason for the bad result. One can see a gap in the range of singular values around $i = 135$. Perhaps we get better results if we reduce to that number of pins. But this is also more expensive. In Tab. 4.2 the time needed for computing the

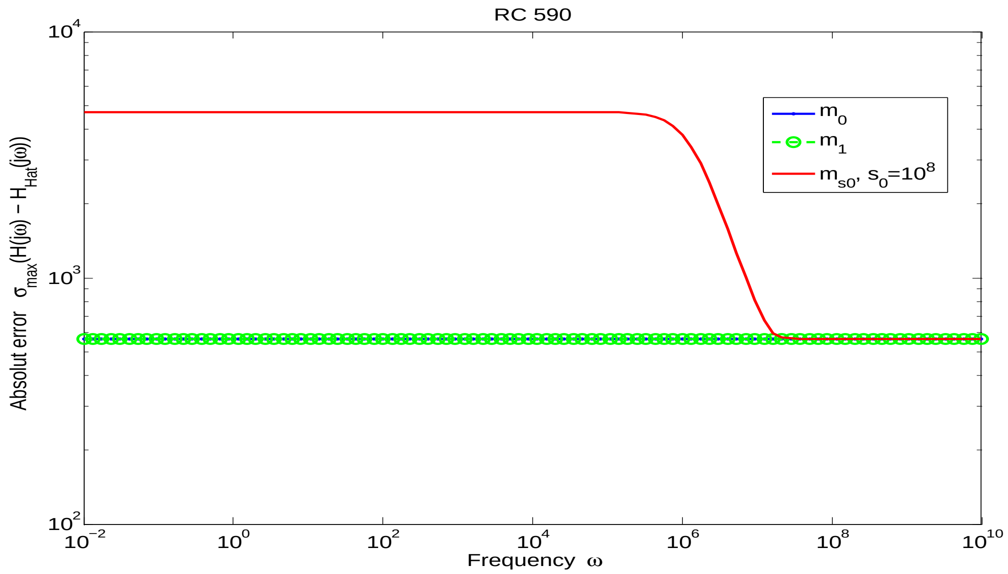

Figure 4.6: Absolute error $\varepsilon_{abs}$ of *RC590* with the help of SVDMOR ($\mathbf{m_0}$, ($\mathbf{m_1}$, and $\mathbf{m_{s_0}}$).

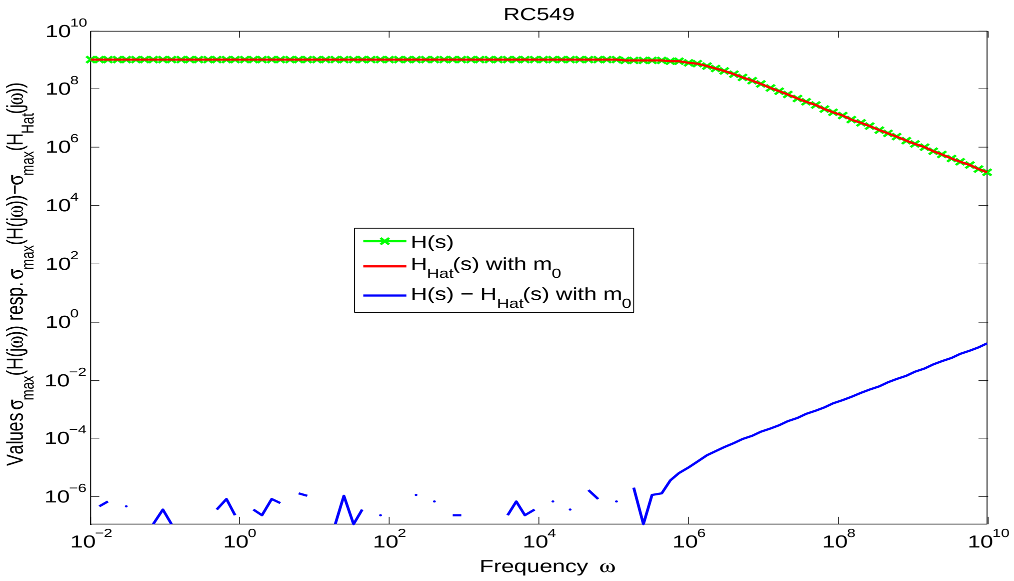

Figure 4.7: Values of $H(s)$ and $\hat{H}(s)$ in spectral norm and their difference with the help of SVDMOR ($\mathbf{m_0}$).

| Number $r$ of computed singular values | Time in seconds |
|---|---|
| 5 | 0.1502 |
| 10 | 0.3404 |
| 30 | 1.1316 |
| 50 | 1.7925 |
| 100 | 4.1860 |
| 150 | 10.4850 |
| 500 | 716.2098 |

Table 4.2: Time for computing the $r$ largest singular values $\sigma_i$ of $\mathbf{m_0}$ of *circuit3* with the MATLAB function `svds`.

singular values of $m_0$ of *circuit3* is shown. One can see the necessity of sparse algorithms. The example *circuit3* will be investigated in future research, see Chap. 5.

# 5 Summary and Outlook

## 5.1 Summary

In this work we have investigated approaches of terminal and model order reduction by the means of low rank matrix factorization.

After a short introduction, basic knowledge of modeling a problem in circuit simulation was introduced in Chap. 2. We started with explaining required definitions, Kirchhoff's laws and certain circuit element depending characteristic equations. We showed how to obtain a differential algebraic equation that describes the circuit with the help of a common modeling method, the modified nodal analysis. Furthermore, we defined the terms descriptor system and transfer function. Properties of the transfer function were followed by the explanation of several mainstream MOR methods. The first part dealt with Krylov subspace methods, the second one with balanced truncation.

Chapter 3 introduced recent MOR methods for reducing the number of terminals of a circuit during the simulation. We showed two basic methods, SVDMOR and ESVDMOR. Both are based on the singular value decomposition, which is the biggest drawback of these methods. We showed, that the ESVDMOR approach needs more computational steps and causes higher costs but it leads to more exact results in case of a large difference between the number of input and output terminals. If this difference is small, the SVDMOR approach is the cheaper and consequently the better choice. The SVD was explained in Sec. 3.1.1. Due of the enormous numerical costs of this method, we showed alternative matrix factorization methods. The first method was the truncated SVD, which was also used later in terms of a MATLAB function. Furthermore, other approaches were shown, for example a sparse pivoted QR decomposition (SPQR), based on the quasi-Gram-Schmidt algorithm, and the sparse column row approximation (SCRA). We showed the possibility to use the SPQR or the SCRA instead of the SVD in the approaches of Sec. 3.1. At the end of this Chapter 3, we described the errors and some error bounds for most of the methods above.

In Chap. 4, we explained how the implemented algorithm works. We also showed some parts of the source code in Sec. 4.1 and mentioned assets and drawbacks of the algorithm.

We recognized that the DC response moment is not useful to determine the level of the approximation. In Sec. 4.2 we showed that the selected approaches lead to results which were very useful for conclusions. Furthermore, reasons for using the selected approaches as well as the algorithm as a base for future research were given. The results showed, that it is necessary to implement sparse algorithms but that we can get a very useful terminal and order reduced model with the help of the ideas shown in this work.

## 5.2 Outlook

There are many things left that need to be investigated. This will be done within the research network *System Reduction for Nanoscale IC Design* (SyreNe) within the program *Mathematics for Innovations in Industry and Services* funded by the German Federal Ministry of Education and Science (BMBF).

In the near future, the development of an efficient algorithm, which also uses balanced truncation or Krylov subspace methods for the second part of the reduction, will be emphasized. However, before this last step can be attempted, a large number of problems need to be solved. First, an efficient way of computing higher order moments needs to be developed. Anyway, the goal is to get a sparse approach for the whole computation. Regarding efficiency, one point is to deal with as less memory as possible, which is the main problem for examples of higher dimension $n$ right now.

Another key problem is to determine the number of terminals we want to reduce to. That means we have to improve the analysis of the range of the singular values without using too many computational resources. In many cases, the relationship $B = L^T$ holds, which should be taken advantage of in our algorithm. Furthermore, other matrix decomposition approaches like SCRA or SPQR need to be implemented. The final step is to link an efficient source code into the industrial circuit simulator TITAN of the Qimonda AG.

# Bibliography

[BB97] P. Benner and R. Byers. Disk functions and their relationship to the matrix sign function. In *Proc. European Control Conf. ECC 97,* Paper 936. BELWARE Information Technology, Waterloo, Belgium, 1997, 1997. CD-ROM.

[BDG97] Z. Bai, J. Demmel, and M. Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numerische Mathematik*, 76(3):279–308, 1997.

[Ben03] P. Benner. Spectral projection methods for balanced truncation of descriptor systems. 8th SIAM Conference on Applied Linear Algebra, Williamsburg, July 15-19, 2003.

[BMS05] P. Benner, V. Mehrmann, and D.C. Sorensen. *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

[BPS05] M. W. Berry, S. A. Pulatova, and G. W. Stewart. Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices. *ACM Trans. Math. Softw.*, 31(2):252–269, 2005.

[BQO98] C. H. Bischof and G. Quintana-Orti. Computing rank-revealing QR factorizations of dense matrices. *ACM Trans. Math. Softw.*, 24(2):226–253, 1998.

[BQOQO04] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Parallel model reduction of large-scale linear descriptor systems via Balanced Truncation. In *High Performance Computing for Computational Science. Proc. 6th Intl. Meeting VECPAR'04, June 28–30, 2004, Valencia, Spain*, pages 65–78, 2004.

[BR05] J. Baglama and L. Reichel. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM J. Sci. Comput.*, 27(1):19–42, 2005.

[CGO07] R.H. Chan, C. Greif, and D.P. O'Leary, editors. *Milestones in matrix computation. The selected works of Gene H. Golub. With commentaries by Anne Greenbaum, Åke Björk, Nicholas Higham, Walter Gautschi and G. W. Stewart.* Oxford: Oxford University Press, 2007.

[CH90] T. F. Chan and P. C. Hansen. Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations. *SIAM J. Sci. Stat. Comput.*, 11(3):519–530, 1990.

[Cob84] D.J. Cobb. Controllability, observability and duality in singular systems. *IEEE Transactions on Automatic Control*, 29:1076–1082, 1984.

[Dai89] L. Dai. *Singular Control Systems*, volume 118 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1989.

[Dem97] J. W. Demmel. *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[FF94] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *Proceedings of EURO-DAC '94 with EURO-VHDL '94, Grenoble, France*, pages 170–175. IEEE Computer Society Press, 1994.

[FL04] P. Feldmann and F. Liu. Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 88–92, Washington, DC, USA, 2004. IEEE Computer Society.

[FNG92] L. Fortuna, G. Nummari, and A. Gallo. *Model Order Reduction Techniques with Applications in Electrical Engineering*. Springer-Verlag, 1992.

[Fre03] R. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.

[Fro03] A. Frohberger. Model reduction techniques for linear and nonlinear dynamical systems. Internship report, Philips Electronics, June 2003.

[GFtM05] M. Günther, U. Feldmann, and E.J.W. ter Maten. Modelling and discretization of circuit problems. In W.H.A. Schilders and E.J.W. ter Maten, editors, *Numerical Analysis in Electromagnetics, Special Volume of Handbook of Numerical Analysis*, volume XIII, pages 523–659, 2005. Elsevier Science BV, Amsterdam,.

[GK65] G.H. Golub and W. Kahan. Calculating the singular values and pseudo inverse of a matrix. *SIAM J. Numer. Anal.*, pages 205–224, 1965.

[GR70] G.H. Golub and C. Reinsch. Singular value decompositions and least squares solutions. *Numer. Math.*, 14:403–420, 1970.

[GV96] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[Gün08] R. Günzel. Balanced Truncation for Model Reduction of Descriptor Systems Arising in Circuit Simulation. Master's thesis, Chemnitz University of Technology, Department of Mathematics, Germany, to be published, 2008.

[Hoc01] M. E. Hochstenbach. A Jacobi–Davidson type SVD method. *SIAM J. Sci. Comput.*, 23(2):606–628, 2001.

[KD99] L. Knockaert and D. De Zutter. Passive reduced order multiport modeling: the Padé-Laguerre, Krylov-Arnoldi-SVD connection. *Int. J. Electronics and Communications*, 53(5):254–260, 1999.

[KFA69] R.E. Kalman, P.L. Falb, and M.A. Arbib. *Topics in Mathematical System Theory*. McGraw-Hill, New York, 1969.

[Lau91] A.J. Laub. Invariant subspace methods for the numerical solution of Riccati equations. In S. Bittanti, A.J. Laub, and J.C. Willems, editors, *The Riccati Equation*, pages 163–196. Springer-Verlag, Berlin, 1991.

[LS96] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

[LSY98] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK user's guide. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.* Software - Environments - Tools, 6. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. 142 p., 1998.

[LTL$^+$05] P. Liu, S. X. D. Tan, H. Li, Z. Qi, J. Kong, B. McGaughy, and L. He. An efficient method for terminal reduction of interconnect circuits considering delay variations. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM International Conference on Computer-Aided Design*, pages 821–826, Washington, DC, USA, 2005. IEEE Computer Society.

[LTYM06] P. Liu, S.X.-D. Tan, B. Yan, and B. Mcgaughy. An extended SVD-based terminal and model order reduction algorithm. In *Proceedings of the 2006 IEEE International, Behavioral Modeling and Simulation Workshop*, pages 44–49, Sept. 2006.

[LTYM08] P. Liu, S. X. D. Tan, B. Yan, and B. McGaughy. An efficient terminal and model order reduction algorithm. *Integr. VLSI J.*, 41(2):210–218, 2008.

[Moo81] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, 26:17–32, 1981.

[MS05] V. Mehrmann and T. Stykel. Balanced truncation model reduction for large-scale systems in descripter form. In P. Benner, V. Mehrmann, and D.C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 83–115. Springer-Verlag, Berlin/Heidelberg, Germany, 2005.

[OCP98] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm. *IEEE Transaction on Computer Aided-Design of Integrated Circuits*, 17(8):645–654, 1998.

[Rob80] J.D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980. (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department, 1971).

[Rot05] T. Rothaug. Linking KROM to TITAN. Internship report, Infineon Technologies AG, 2005.

[SBD$^+$05] R. Süße, P. Burger, U. Diemar, B. Marx, and T. Ströhla. *Theoretische Grundlagen der Elektrotechnik 1.* B.G. Teubner Verlag, Wiesbaden, Germany, 1st edition, 2005.

[Sch04] Wim Schoenmaker. Compact modelling of on-chip passive structures at high frequencies (CODESTAR), final report with contributions from partners. Technical report, Information Society Technologies (IST) Programme, 2004.

[Sil94] L. Silveira. *Model order reduction techniques for circuit simulation.* PhD thesis, MIT, Cambridge, 1994.

[SKEW96] L. M. Silveira, M. Kamon, I. Elfadel, and J. White. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In *ICCAD '96: Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*, pages 288–294, Washington, DC, USA, 1996. IEEE Computer Society.

[Sor92] D. C. Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13(1):357–385, 1992.

[Ste99] G. W. Stewart. Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numerische Mathematik*, 83(2):313–323, 1999.

[Sto08] M. Stoll. A Krylov-Schur approach to the truncated SVD. *Submitted to SIAM J. Sci. Comput.*, 2008.

[Sty02] T. Stykel. *Analysis and Numerical Solution of Generalized Lyapunov Equations*. Dissertation, TU Berlin, 2002.

[Sty04] T. Stykel. Gramian-based model reduction for descriptor systems. *Math. Control Signals Systems*, 16:297–319, 2004.

[TH07] S. Tan and L. He. *Advanced Model Order Reduction Techniques in VLSI Design*. Cambridge University Press, New York, NY, USA, 2007.

[Ulb03] G. Ulbricht. *Grundgesetze der Elektrotechnik*. J. Schlembach Fachverlag, Wilburgstetten, Germany, 2003.

[Unb00] R. Unbehauen. *Grundlagen der Elektrotechnik 2*. Springer-Verlag, Berlin/Heidelberg, Germany, 5th edition, 2000.

[Wol86] I. Wolff. *Grundlagen der Elektrotechnik 2*. Verlagsbuchhandlung Nellissen - Wolff, Aachen, Germany, 2nd edition, 1986.

[Wol88] I. Wolff. *Grundlagen der Elektrotechnik 1*. Verlagsbuchhandlung Nellissen - Wolff, Aachen, Germany, 4th edition, 1988.

[WP02] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA*, 40(11):2323–2330, 2002.

[YS81] E.L. Yip and R.F. Sincovec. Solvability, controllability and observability of continous descriptor systems. *IEEE Trans. Automat. Control*, AC-26:702–707, 1981.

# Theses

### Thesis 1

The modified nodal analysis is very suited to the modeling of electrical circuits as a mathematical model. Basis for this are Kirchhoff's laws and element describing characteristic equations. The result in general is a differential algebraic equation which reflects the structure of the modeled circuit.

### Thesis 2

Modeling a linear circuit leads to a descriptor system and to its transfer function with a regular matrix pencil. The transfer function allows statements about passivity and stability of the system.

### Thesis 3

Krylov subspace model order reduction methods are based on the projection of the system onto a Krylov subspace of lower dimension. These methods are robust and well known but there are no known error bounds and they can not handle more than a few numbers of terminals.

### Thesis 4

Balanced truncation methods are also well known for linear problems. They have the big advantage of an error bound. That makes an error estimation of the MOR process possible. A drawback is that these methods also cannot handle systems with many terminals.

### Thesis 5

The decomposition of the input and output defining matrices of the transfer function is a suitable approach to handle systems with a large number of terminals. This enables the usage of the methods mentioned above for this kind of problems. Useful basic ideas are the (E)SVDMOR approaches.

### Thesis 6

Weakness of the (E)SVDMOR methods is the very expensive full singular values decomposition. The usage of other matrix decomposition approaches like a truncated SVD, SPQR, SCRA or JDSVD improve this and lead to an algorithm which can handle large problems with acceptable results and errors and gives a very compact terminal and order reduced system.

### **Thesis** 7

Error bounds and estimations exist for a lot of single steps in the whole algorithm. The development of a global error bound is within the realms of possibility.

# Selbstständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Chemnitz, 6th April 2008

André Schneider