CHAPTER 1

Introduction

This chapter provides an introduction to the topics treated in this thesis. We first motivate the general need for system analysis and later on focus on electrical networks. Different methods for system analysis are surveyed and some historical background information is given especially for symbolic analysis. Using complexity considerations of problems from electrical engineering, the need for model order reduction techniques is motivated, for both the symbolic and the numerical case. The research area of reduction methods is then briefly traced in the fourth section and general ideas for symbolic and the most popular numerical techniques are outlined. Two application examples from weather prediction and electrical circuit design are given to further motivate the need for both types of system analysis in combination with reduction techniques. The chapter finally concludes with a description of the aims of this work.

1.1. System Analysis

In our today's technological world, physical processes are mainly described by mathematical models, i.e. dynamical systems whose future behavior depends on their past evolution. The resulting systems of equations in general consist of *differential-algebraic equations* (DAE), since the differential equations coming from the dynamical part of the *real* system mostly are restricted by certain physical laws contributing additional algebraic constraints. Equations containing PDE parts usually are semidiscretized w.r.t. space. System analysis investigates the behavior of a *real* system by making use of such mathematical models. The input-output behavior of the *real* system then corresponds to solving the mathematical system of equations.

Such mathematical models are employed in versatile areas. Their main application, however, is for simulation and control in order to predict or modify the behavior of a *real* system. Especially for electrical networks, simulator tools play a key role in the design process already for decades. In the 50's and 60's of the last century, electrical circuits were designed by using discrete components for a prototype realization and a subsequent adjustment (*redesign*) by taking measurements. A little later, when integrated circuits (ICs) with increased integration density were introduced, these techniques had reached their limits.

On the other hand, using a simulator permits an easy verification of the circuit performance without having it realized as hardware. Advantages such as the possibility for fast modifications and an easy testing of ideas for design are obvious. Particularly redesigns of an accidentally deficiently designed electrical circuit, accompanied with immense time and financial costs in the industrial production of modern ICs, can be made on the computer screen and be verified by a subsequent simulation run. Parasitic effects caused by measurement setups such as inductances due to cables are avoided and analysis under very different operating conditions is possible. Therefore, simulator tools are indispensable in modern industrial circuit design.

1.2. Analysis Methods

For the analysis of a system, there are different methods which will be described and compared to each other in this section. While the most widely used method, *numerical analysis*, can be applied to systems of large size, *symbolic analysis* allows deeper insights into functional dependences among system parameters.

Numerical Analysis. The most commonly used method for system analysis is numerical analysis. Here, all parameters in the describing system of equations have to be given by their numerical values. Then the system is solved and the obtained solution data can be analyzed and displayed graphically.

In the context of analog circuits, starting from a netlist description of the circuit topology, numerical simulators are able to automatically generate and subsequently solve a system of describing equations for the circuit's behavior. For the numerical calculation, all parameters of the system have to be given by suitable numerical values, i.e. a complete dimensioning of the circuit's components has to be provided.

Using numerical methods for circuit analysis, lots of important characteristics such as amplification factors of amplifiers are computable. Moreover, numerical analysis is applicable to systems of large size. On the other hand, particularly for its *analog* parts which are usually much smaller than the entire circuit, an accurate prediction of the fully sized system's behavior is opposed by no qualitative insights into functional dependences among system parameters and their effects on the system's behavior. This is due to the output solely consisting of "tables of numbers" which is a severe drawback especially for early stages in the design process. For investigations aiming at a deeper understanding of the circuit functionality, one has to carry out further simulations with different numerical parameter values, but this still does not guarantee any success.

Hand Analysis. If one is interested in an analytical description, e.g. in order to gain insights into the system's behavior, a direct analysis of the system equations is necessary.

Usually the access to "internal" equations of a simulator is not given, so up to around 1970, analytic computations had to be carried out by hand. In order to keep the level of complexity low, simplified models for the system components were used which omit all but the most relevant effects, whereas numerical analysis usually uses very precise models taking almost all possible physical effects into account. The validity of the use of simplified models was mostly proven afterwards, if ever.

By intuition, it is clear that *hand computations* are very tedious and error-prone and, particularly in the context of electrical circuits, need a lot of experience and knowledge in circuit design. Furthermore, there is no possibility for an error check during the computations.

Symbolic Analysis. With the development of computer algebra systems, the approaches of the symbolic *hand analysis* could be automated. Around 1970, the first computer programs for *symbolic circuit analysis* were introduced in order to capture the circuit's behavior analytically and in dependence on its parameters. In this context, *symbolic* means that not only the variables, but also the *parameters* are given as symbols in the corresponding system of equations. Solving the *symbolic* equations then yields visible dependences of the system's behavior on its parameters.

Nevertheless, the limits in efficiency and disc space were reached soon, thus preventing symbolic methods for the use with problems of practical size. Therefore, interest was lost for several years. Just at the end of the 1980's, this area of research became revitalized by the enormous increase of computer performance and the development of *symbolic reduction methods*[1] that allowed an application to circuit problems of larger size. Since then it was possible to use symbolic analysis for *linear* circuits even of industrial size. Due to this, in the 1990's one tried to transfer these approaches to nonlinear circuits [**Bor97, Hen**].

Particularly for early stages in the design process of analog circuits, symbolic analysis tools proved to be useful. Especially when the symbolic equations can be solved explicitly for its output variables, they provide an automatic generation of mathematical formulas that express performance characteristics in terms of circuit parameters. Unlike waveforms produced by numerical simulators, symbolic expressions allow to read off influences of components on circuit characteristics and, hence, to identify those parameters that have to be altered in order to meet certain design specifications.

In combination with *symbolic reduction methods*, symbolic analysis is further used to *automatically* derive *behavioral models*, i.e. parameterized systems of equations describing the approximated circuit behavior, which can be employed for accelerated simulation and optimization of analog circuits. Therefore, symbolic analysis is an indispensable tool that simplifies design, dimensioning, and optimization of analog circuits or, more generally, nonlinear systems.

1.3. Complexity Considerations

In order to get an impression of the high computational complexity, consider the development of ICs or, more precisely, *VLSI circuits* [**Bec, FelParFar, Reis06, Tis**] up to today; while the Intel 4004 released in 1971 incorporated 2300 components with feature sizes of $\sim 10\,\mu$m and an operating frequency of 64 kHz, the Intel Pentium 4 released in 2001 already had about 42 million components with dimensions around 180 nm arranged in seven layers, an interconnecting structure of 2 km length, and an operating frequency around 2 GHz (see [**wikipedia**], Figure 3.1, and Table 3.1 in Chapter 3). The *passive* parts modelling the interconnecting structure finally yield systems with $n \approx 10^5$ to 10^6 equations. Consequently, simulations of the full systems cannot be handled anymore.

[1] Sometimes one can find the term *symbolic approximation methods* in literature. It has to be mentioned that these approximations are not meant in the sense of *Taylor approximations* or *interpolation polynomials* etc.

The situation is worse yet in the symbolic case. Even for small circuits, the limits of symbolic circuit analysis, beyond which mathematically exact computations are feasible, are reached quickly. Considering the number of terms in a linear system's *transfer function* as a measure for its computational complexity, an estimate for this number shows that symbolic circuit analysis even for *linear* circuits has an asymptotical order of complexity in between $\mathcal{O}(a^n)$ (exponential) and $\mathcal{O}(n^n)$ (superexponential), where n denotes the number of nodes in the circuit [**Hen, Moo**]. This shows that *exact* symbolic circuit analysis becomes infeasible very quickly. Furthermore, lengthy expressions of more than one line do not allow qualitative insights and exact results turn out to be useless.

The above considerations motivate the need for complexity reduction techniques. They are inevitable not only to avoid the enormous costs of very large systems. Following the maxim that high precision is less important than physical interpretability, particularly *symbolic* reduction methods are designed to neglect all insignificant information from the describing equations. In this context, from now on we will consider *system analysis with complexity reduction* as follows:

> ***System analysis with complexity reduction*** *is the description of a real physical process by using a suitable system of equations together with its* ***complexity reduction*** *with the aim of* ***analyzing the system's behavior*** *and the* ***generation of behavioral models****.*

The following section introduces model order reduction, reviews some of its application areas, and provides some background of its history.

1.4. Model Order Reduction

The general task of model order reduction techniques is the derivation of approximate models from given large-scale systems which have a significantly lower level of complexity, but still capture the dominant input-output behavior of the original system. The approximate model should satisfy certain user-specified accuracy requirements and preserve important system properties such as stability and passivity.

Lots of reduction approaches have been developed in various areas of research such as electrical and mechanical engineering, control design, computational fluid dynamics, or biological and chemical engineering, see, e.g., [**Ant, BenMehSor, ObiAnd, Rew, SchVorRom**] and references therein. The most popular *numerical* techniques are taylored for *linear* systems and rely on projections onto lower dimensional subspaces of the original system's state space. The corresponding projection bases can be constructed by methods based on Krylov subspaces or singular value decompositions (SVD) of appropriate system matrices.

Although Krylov methods have certain drawbacks, they are well-suited for applications with large-scale systems and, therefore, are very popular particularly in electrical engineering. In contrast to this, SVD methods provide error bounds and preserve certain system properties. But since they cause high numerical costs, their applicability is limited to systems of medium size.

Symbolic model order reduction techniques aim at approximating symbolic systems by models with reduced complexity and increased interpretability. They are indeed costly to compute, but particularly for nonlinear systems of DAEs they additionally allow deeper insights into functional relations among the system parameters given symbolically [**Hen, Wic04**]. Symbolic methods further allow the generation of parameterized behavioral models for various uses.

The origin of symbolic analysis of nonlinear systems using symbolic reduction methods is found in applications of analog circuit design. There, it is mainly used in addition to numerical simulations as a tool for design, analysis, dimensioning, and optimization of nonlinear systems. Symbolic reduction methods are hybrid numerical and symbolic algorithms which are able to *automatically* reduce the complexity of a given symbolic system of equations according to a user-specified accuracy. Starting at a netlist-description level, the analog circuit is mapped on a symbolic system of DAEs by means of graph-theoretical methods such as the *modified nodal analysis* (cf. Section 2.1.5). Subsequently, comparisons to numerical reference simulations are used to detect the dominant terms of the system. Neglecting the insignificant ones guarantees the preservation of the system's dominant behavior.

At the Fraunhofer ITWM[2], both numerical and particularly symbolic model reduction techniques for complex systems are being developed and applied. Symbolic reduction techniques are the core of the software package *Analog Insydes* [**AI**] developed by the ITWM which is an add-on for the computer algebra system *Mathematica* [**MMA**]. *Analog Insydes* is the implementation platform for the algorithms developed in this thesis.

1.5. Motivations for Symbolic and Numerical Methods

This section motivates both symbolic and numerical analysis for dynamical systems by describing two applications in which such systems of high complexity arise. While the first example concerns *weather prediction* and *data assimilation*, the second is taken from *electrical engineering*.

North Sea Wave Surge Forecast. For this paragraph, we follow the notes of [**Ant**, Section 2.2.1]. The problem has originally been studied in [**HeeVerSeg, Ver**].

Since parts of The Netherlands are below sea level, the monitoring of wave surges at river mouths is important. In case of need, water barriers can be closed in order to prevent a flooding of the landscape. However, for certain reasons a respective warning has to be given six hours in advance.

To be able to forecast such wave surges, one uses the *shallow water equations*[3] as a PDE model predicting their evolution. In addition to this, the water level and the movement of sea currents are measured at various locations. The resulting data assimilation problem affords a prediction of wave surges based on the model and the measurements.

[2] Fraunhofer Institute for Industrial Mathematics (Fraunhofer Institut für Techno- und Wirtschaftsmathematik), Kaiserslautern, Germany.

[3] The average depth of the North Sea is not more than 100 m.