Applications of Model Order Reduction for IC Modeling

The work described here is financially supported by NXP Semiconductors

A catalogue record is available from the Eindhoven University of Technology Library
ISBN: 978-90-386-2470-9

Applications of Model Order Reduction for IC Modeling

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College
voor Promoties in het openbaar te verdedigen
op woensdag 27 april 2011 om 16.00 uur

door

Maria Vladimirovna Ugryumova

geboren te Novosibirsk, Rusland

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. W.H.A. Schilders

Copromotor:
dr. M.E. Hochstenbach

Contents

Chapter 1

Introduction

The semiconductor industry is concerned with many different areas. One of those areas is the modeling of integrated circuits (ICs), which are used in all kinds of electronic devices such as computers, mobile phones, navigation equipment, etc. Integrated circuit modeling helps to understand the way in which integrated circuits work. Besides it is also useful for predicting errors and for optimizing the design before fabricating ICs. Modeling of ICs uses mathematical models describing the behaviour of both the individual components and the interactions between them. Predicting the behaviour of an IC before building it improves the efficiency and provides useful information to the circuit designers. This is the case, because building ICs just to perform some tests is usually very expensive and time consuming.

We can summarize this by saying that, modeling of ICs is a key tool in the electronics industry. One of the characteristics in this industry is the always increasing complexity. The number of components in a single chip keeps increasing day after day. For example, in Figure 1.1, we can observe a graph showing the growth of the number of transistors in different commercial integrated circuits. This is known as Moore's Law.

In addition to the former, nowadays digital technologies operate with frequencies of the order of GHz. This results into non-negligible layout effects, so-called parasitic effects, which may be critical for ICs performance. As a result, an IC may not meet the necessary requirements or may not function at all. Therefore, models that are aimed at capturing parasitic effects should be developed closely to the requirements prescribed by the physics of an IC.

All together, this implies that in order to describe an IC properly, it is necessary to construct mathematical models which contain many variables. To be able to perform

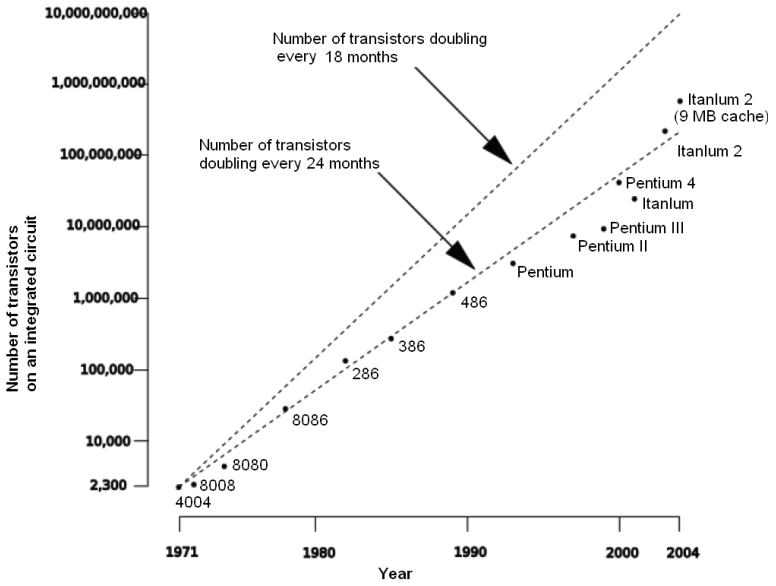

Figure 1.1: *Moore's Law, source Wikipedia*

simulations, it becomes necessary to approximate such models, by models of lower complexity, while approximately keeping the same behaviour. This is known as model order reduction (MOR) problem.

Roughly speaking, the problem of model order reduction is to replace a given mathematical model by a much "smaller" model, which describes accurately enough certain aspects of interest of the original model. MOR involves a number of interesting issues. The first issue is selecting an appropriate approximation scheme that allows to define a suitable reduced-order model. In addition, it is often important that the reduced-order model preserves certain properties of the original model, such as stability or passivity. Other issues include the characterization of the quality of the reduced-order models, and efficient, numerically stable computational procedure.

In more recent years much research has been done in the area of model order reduction. Consequently, a large variety of methods is already available [77] [9] [85]. Some are tailored to specific applications, others are more general. One of the most promising directions of research in the area, is the desire to have physically motivated reduced models. In this thesis we consider both kinds of approaches. For instance, we consider the physically motivated approach, used in the layout simulation tool Fasterix [20] [93], which was developed by Philips for the electromagnetic analysis of complex interconnect structures. In Fasterix the interconnect structure is discretized and the boundary

element method is used to model the structure as an RLC circuit[1]. In the current implementation, this is done by the use of the so-called "super node algorithm", which works based on physical principles. Besides physically motivated approaches for reduction, in the thesis we also consider more general ones based on approximating large resistor networks by optimizing the conductance matrices. Thus, the practical necessity of model order reduction for ICs modeling inspired us to study the topic of this thesis. This research was carried out in a project from NXP Semiconductors which provided realistic industrial problems considered in this thesis.

1.1 Outline of this thesis

As for an outline of this thesis, in Chapter 2 we start with a brief introduction to existing methods used for modeling of interconnect and substrate. In particular, we review the approach used in the electromagnetic tool Fasterix [20] [93] which is currently used at NXP Semiconductors. We also showed that modeling homogeneous substrate requires us to solve the Laplace equation. Since modeling of interconnect and substrate is closely related to ideal electrical circuits, we also discuss the main properties of circuit equations and the problem of circuit synthesis.

The next chapter reviews methods for model order reduction. Nowadays, this is a very popular field of research, both in the scientific computing community and in the area of dynamical systems and control. In the former field, projection methods are the basis of most techniques, while in the field of control one is usually concerned with the solution of large systems of Lyapunov equations and calculation of Hankel singular values. Since modern IC design results in many-terminal networks, we review state of the art methods for reduction of many-terminal networks and discuss the current challenges. In particular, we concentrate on the reduction technique used in the tool Fasterix, and methods for reduction of resistor networks, as this constitutes the basis of the methods developed in this thesis.

Chapter 4 then discusses in detail the reduction technique, the super node algorithm (SNA), from the tool Fasterix. We note that SNA delivers stable reduced models. Nevertheless, we prove that the passivity is not always guaranteed. To overcome this problem we consider two new approaches and discuss their applicability for realistic interconnect models.

In Chapter 5, we discuss the challenges in the reduction of large resistor networks. We suggest a novel method for reduction of large resistor networks. Our approach is based on replacing the original network by an approximate one with much less resistors, while

[1] circuit consisting of resistors, inductors and capacitors

keeping the error within some given margin. We suggest a few errors for controlling the quality of approximation and derive correspondent estimations. A key component in these methods, is that we derive analytical estimations, which keep under control the quality of approximation.

The problem of homogeneous substrate modeling is considered in Chapter 6. In this case, we concentrate mostly on two discretization techniques, the finite element method and the boundary element method. We compare both techniques on qualitative level and show the advantages of using the boundary element method to model homogeneous substrates.

An interesting part of the research concentrated on modeling a power MOS transistor. What is given here is a large resistor network containing the top and the bottom ports (transistor fingers), and a nonlinear relation between the voltage difference at the bottom ports and the current flowing through the bottom transistors. In Chapter 7, we suggest an algorithm for computing output current at the top ports when voltage excitations are given. The suggested algorithm exploits the connection between the resistor and the transistor networks and benefits from the sparsity of the conductance matrix. We also discuss the performance of some methods developed in Chapter 5 to reduce the resistor network of the power MOS transistor.

The thesis is concluded with Chapter 8, which summarizes the obtained results and presents recommendations for future research.

Chapter 2

Modeling and simulation of PCBs, ICs and electrical circuits

Due to the rapid developments in very large-scale integration (VLSI) technology, parasitic effects arising in chip's performance are becoming non-negligible. Therefore, in the design of chips at high frequencies, it is important to be able to simulate electrical behaviour of the chip before it is fabricated to avoid many expensive design iterations. In this chapter we briefly review existing methods for interconnect and substrate modeling, while some of these methods will be used later on in the thesis. Since these methods in general lead to linear circuits with elements such as resistors, inductors and capacitors, we also review a generic formulation and main properties of electrical circuits.

2.1 Parasitics in integrated circuits

Figure 2.1 schematically shows a vertical cross section of an integrated circuit (IC). It consists of three layers: interconnect, which includes conductors and dielectric layers, layout with devices, and substrate. Ideally interconnect is an ideal conductor, devices (transistors, diodes) are ideal switches, and substrate is an ideal insulator. However, in practice this is not the case. Interconnect is not an ideal conductor, which causes delays along the interconnect; transistors are not ideal switches; and the substrate is mainly a resistive domain which may cause crosstalk between different parts of the IC. These phenomena are called *parasitics*. At low frequencies these parasitic effects are usually negligible because switching delay in the transistors and delays along the

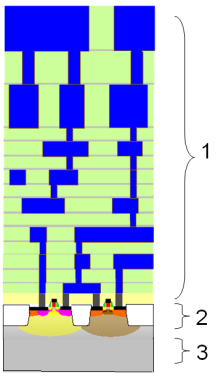

Figure 2.1: *Layers of IC: 1. interconnect, 2. devices, 3. substrate*

interconnect are much smaller in comparison to the speed of the signal [78]. Since field couplings through the dielectric and the resistive domain are weak at low frequencies, the crosstalk between signals is negligible.

Nowadays, digital technologies operate with frequencies of the order of gigahertz (GHz). This results into non-negligible delays and field couplings which may be critical for IC performance. Delays may cause synchronization problems, while parasitic field coupling through the dielectric and substrate may cause signal-integrity problems. As a result, the circuit may not meet the necessary requirements or may not function at all. Therefore, models that are aimed at capturing parasitic effects should be developed closely to the requirements prescribed by the physics of the integrated circuit.

To exploit the underlying physics of IC it is reasonable to subdivide the whole modeling problem of IC into three subproblems: modeling of interconnect, modeling of devices, and modeling of substrate. Further we will be concentrated on interconnect and substrate modeling. Modeling of devices [33] is a field of research in itself and, therefore, is outside the scope of this thesis.

2.2 Modeling of interconnect

Nowadays, interconnects arise at various levels of design hierarchy such as ICs, packaging structures, printed circuit boards (PCB), etc. The need of high clock frequencies has led to the previous negligible effects arising in interconnect such as signal delay, crosstalk, attenuation. As a result, interconnect becomes responsible for majority of signal degradation in high speed systems. If not considered during the design stage, these interconnect effects can distort analog signal that fails an IC to meet specifications. Since extra iterations in the design are costly, accurate prediction of these effects is a necessity

for high-speed designs. Therefore, it is very important for designers to simulate the entire design along with interconnect subcircuits as efficiently as possible while retaining the accuracy of simulation.

2.2.1 Interconnect models

Depending on the operating frequency, signal rise times and interconnect structure, the interconnects can be modeled as lumped, distributed, or full-wave models [3].

At low frequencies the interconnect can be modeled using circuit models consisting of resistors and inductors (RC circuit) or circuit consisting of resistors, inductors, capacitors, and conductances (RLCG circuit). The conventional approach for modeling of distributed interconnects located above a ground plane at low frequencies is to divide the line (interconnect) into segments of length Δz [69]. If the segment is smaller than the wavelength, i.e., $\Delta z \ll \lambda$, then the segment can be replaced by RLCG circuit presented in Figure 2.2. In case of a perfect conductor in a homogeneous surrounding medium (i.e., the medium has the same properties at all points), the circuit elements $R\Delta z$, $L\Delta z$, $C\Delta z$, and $G\Delta z$ are proportional to segment's length, Δz, where the parameters R, L, C, G are resistance, inductance, capacitance, and conductance of the line, respectively.

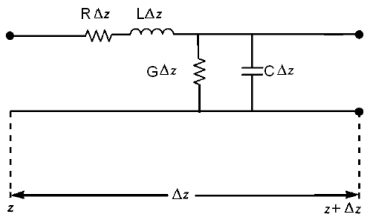

Figure 2.2: *Equivalent circuit of the transmission line segment. Image taken from [69].*

In case of a perfect conductor, i.e. $R = 0$, the current in a conductor at low frequencies is distributed uniformly throughout its cross sections. In case of an imperfect conductor, i.e. $R = R(\omega)$, the currents will be uniformly distributed over the conductor cross section at low frequencies but at high frequencies will migrate toward the surfaces of the conductors. This phenomenon can be categorized as skin, edge, and proximity effects [3] [69]. Due to the skin effect the current is concentrated in a thin layer near the conductor surface. As a result, this reduces the cross section available for signal propagation and, therefore, increases the resistance to signal propagation. Due to the edge effect, the current is concentrated near the sharp edges of the conductor, while the proximity effect causes the current to concentrate in the sections of ground plane which are close to the signal conductor. To account for this effect, modeling based on frequency-dependent parameters $R = R(\omega)$, $L = L(\omega)$, $C = C(\omega)$, and $G = G(\omega)$ may

be necessary.

When operating frequencies are of the order of GHz, the model which is suitable for low frequencies becomes inefficient due to electromagnetic (EM) spatial effects. As a result, full-wave EM simulations are needed. The modeling of the interconnect can be performed, for instance, by constructing partial equivalent circuit (PEEC) model. PEEC models are RLC circuits which are extracted from the geometry of interconnect using solution of Maxwell's equations which we will consider later on in this thesis. An advantage of PEEC model is that result of such EM simulation can be integrated with the simulation of the chip's circuit elements [90].

In this thesis we will concentrate on the EM modeling approach, which is very similar to the PEEC method and currently used in the EM tool Fasterix [20] [93]. The tool Fasterix analyzes EM behaviour of interconnect and constructs a circuit, which describes behaviour of interconnect up to a given maximum frequency. An important feature is that the circuit is generated only once, i.e., does not have to be repeated for each time and frequency. In the next section we briefly present the main steps in Fasterix to derive the RLC circuit. Further it will be assumed that interconnect consists of planar and thin conductors.

2.2.2 Interconnect extraction in Fasterix

The goal of parasitic interconnect extraction in Fasterix [20] [93] is to determine a relation between the currents and the voltages at the terminals (ports) of the conductors. For an n-port model at the frequency ω, this relation is described by the admittance matrix $Y_p \in \mathbb{C}^{p\times p}$ [53]:

$$\mathbf{J}_p = Y_p \mathbf{V}_p, \tag{2.1}$$

where $\mathbf{J}_p$ and $\mathbf{V}_p$ are the vectors of the port currents and voltages, respectively. Below we show the main steps to obtain relation (2.1) from Maxwell's equations. More details about each step can be found in [20] [93] and Chapter 4 of this thesis.

Electric field integral equations

The Maxwell's equations for harmonic fields are given by [20] [97]

$$\nabla \times \mathbf{E} = i\omega\mathbf{B} \quad \text{(Faraday's law)}, \tag{2.2}$$
$$\nabla \times \mathbf{H} = \mathbf{J} - i\omega\mathbf{D} \quad \text{(Ampere's law)}, \tag{2.3}$$
$$\nabla \cdot \mathbf{B} = 0 \quad \text{(Gauss's law)}, \tag{2.4}$$
$$\nabla \cdot \mathbf{D} = \rho, \tag{2.5}$$

where **E** is the electric field, **H** is the magnetic field, **B** is the magnetic induction, **D** is the electric displacement, **J** is the current density, ρ is the charge density.

In view of the later applications, it is convenient to introduce the potential ϕ to obtain a smaller number of equations, which are equivalent to Maxwell's equations:

$$\frac{\mathbf{J}}{\sigma} + \nabla\phi - i\omega \int_\Omega \mathbf{G_A}\ \mu\ \mathbf{J}\ d\mathbf{x}' = 0, \tag{2.6}$$

$$\nabla \cdot \mathbf{J} - i\omega\rho = 0, \tag{2.7}$$

$$\phi - \int_\Omega G_\phi \frac{\rho}{\epsilon}\ d\mathbf{x}' = 0, \tag{2.8}$$

where Ω is the conductor area, $\mathbf{G}_A$ and $\mathbf{G}_\phi$ denote solutions of Helmholtz equations [93], ϵ and μ denote permittivity and permeability of medium. Equations (2.6)–(2.8) are collectively referred to as the mixed potential electric field integral equations (EFIE). Additionally, it is required that no current flows through the boundary Γ of the conductor, i.e.,

$$\mathbf{J} \cdot \mathbf{n} = 0, \quad \mathbf{x} \in \Gamma, \tag{2.9}$$

and the potential at the ports of the conductor are known, i.e.,

$$\phi(x) = \mathbf{V}_{\text{fixed}}, \quad x \in \Gamma_V,$$

where Γ_V denotes the boundary of the conductor which plays the role of ports.

Weak formulation and discretization

Constructing a weak formulation of (2.6)–(2.9) leads to

$$\int_\Omega \left(\frac{\mathbf{J}}{\sigma} \cdot \widetilde{\mathbf{J}} + \phi\nabla \cdot \widetilde{\mathbf{J}} - i\omega \int_\Omega \mathbf{G_A}\mu\ \mathbf{J}\ d\mathbf{x}'\ \widetilde{\mathbf{J}} \right) d\mathbf{x} = 0, \tag{2.10}$$

$$\int_\Omega \left(\nabla \cdot \mathbf{J} - i\omega\rho\right) \widetilde{\phi}\ d\mathbf{x} = 0, \tag{2.11}$$

$$\int_\Omega \left(\phi - \int_\Omega G_\phi \frac{\rho}{\epsilon}\ d\mathbf{x}' \right) \widetilde{\rho}\ d\mathbf{x} = 0, \tag{2.12}$$

where $\widetilde{\rho}$, $\widetilde{\phi}$, $\widetilde{\mathbf{J}}$ are test functions which will be defined further.

Now we will present an overview of the discretization method presented [93] for the above equations. For this goal, the surfaces of the thin planar conductors are divided into a number of quadrilateral elements (quadrilaterals). Let $\mathcal{N}$ denote the total number of quadrilaterals, and let $\mathcal{F}$ denote the total number of edges of quadrilaterals, which are not at the boundary. For example, Figure 2.3 demonstrates a thin conductor discretized into nine quadrilaterals, i.e., $\mathcal{N} = 9$, and $\mathcal{F} = 12$. Let each quadrilateral have constant

potential and charge density. Thus, the scalar potential is defined as

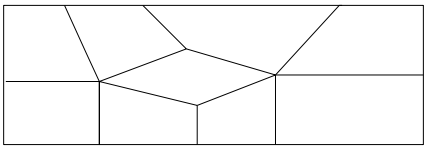

Figure 2.3: *Discretization of a thin conductor*

$$\phi(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}} V_j b_j(\mathbf{x}), \tag{2.13}$$

where V_j is the potential at the j-th quadrilateral and $b_j(\mathbf{x}) = 1$, if $\mathbf{x}$ is on the j-th quadrilateral, zero otherwise. The surface charge density is defined as

$$\rho(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}} Q_j c_j(\mathbf{x}),$$

where Q_j denotes the charge of j-th quadrilateral and $c_j(\mathbf{x})$ is a basis function that adapted to include the singularity of the charge density near the conductor boundary [93]. The current density is defined as a sum over the currents, which flow through the edges defined between two neighboring quadrilaterals:

$$\mathbf{J}(\mathbf{x}) = \sum_{k=1}^{\mathcal{F}} I_k \widetilde{\mathbf{w}}_k(\mathbf{x}),$$

where I_k is the current through the edge k and $\widetilde{\mathbf{w}}_k(\mathbf{x})$ is a basis function, chosen such that the current through each quadrilateral is conserved, i.e., the sum of currents flowing inside of Ω equals current flowing outside of Ω.

Substituting the scalar potential, surface charge and current density into the weak formulation, and choosing the test functions as basis functions, i.e., $\widetilde{\phi} = b_j(\mathbf{x})$, $\widetilde{\rho} = c_j(\mathbf{x})$, and $\widetilde{\mathbf{J}} = \widetilde{\mathbf{w}}_k$, one obtains the following linear system

$$(R - i\omega L)\mathbf{I} - P\mathbf{V} = 0, \tag{2.14}$$

$$-P^T\mathbf{I} + i\omega M\mathbf{Q} = 0, \tag{2.15}$$

$$M^T\mathbf{V} - D\mathbf{Q} = 0, \tag{2.16}$$

where $\mathbf{I} \in \mathbb{R}^{\mathcal{F}}$ is the vector of currents through the edges of quadrilateral elements, $\mathbf{V} \in \mathbb{R}^{\mathcal{N}}$ is the vector of potentials at each quadrilateral, and $\mathbf{Q} \in \mathbb{R}^{\mathcal{N}}$ is the vector of charges at each quadrilateral, R is a $\mathcal{F} \times \mathcal{F}$ sparse symmetric positive definite matrix of

resistances with elements

$$R_{kl} = \int_{\Omega_h} \frac{1}{\sigma} \widetilde{\mathbf{w}}_l(\mathbf{x}) \cdot \widetilde{\mathbf{w}}_k(\mathbf{x}) \; d\mathbf{x},$$

L is a $\mathcal{F} \times \mathcal{F}$ dense symmetric positive definite matrix of partial inductances with elements

$$L_{kl} = \int_{\Omega_h} \widetilde{\mathbf{w}}_l(\mathbf{x}) \cdot \left\{ \int_{\Omega_h} G_A(\mathbf{x}, \mathbf{x}') \mu . \widetilde{\mathbf{w}}_k(\mathbf{x}') \; d\mathbf{x}' \right\} d\mathbf{x}.$$

Elements of P, M, and D are defined as

$$P_{kj} = \int_{\Omega_j} b_j(\mathbf{x}) \nabla \cdot \widetilde{\mathbf{w}}_k(\mathbf{x}) \; d\mathbf{x}, \quad M_{ij} = \int_{\Omega_j} c_j(\mathbf{x}) b_i(\mathbf{x}) \; d\mathbf{x},$$

$$D_{ij} = \int_{\Omega_j} c_j(\mathbf{x}) \left\{ \int_{\Omega_i} C_\phi(\mathbf{x}, \mathbf{x}') \frac{c_i(\mathbf{x}')}{\mathcal{F}} \; d\mathbf{x}' \right\} d\mathbf{x},$$

where $i, j = 1, \ldots, \mathcal{N}$ and $k = 1, \ldots, \mathcal{F}$. Eliminating $\mathbf{Q}$ from (2.14)–(2.16), and subdividing all nodes into internal (index i) and ports (index p), the linear system can be rewritten as

$$\begin{pmatrix} R + sL & -P_i & -P_p \\ P_i^T & sC_{ii} & sC_{ip} \\ P_p^T & sC_{pi} & sC_{pp} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \mathbf{V}_i \\ \mathbf{V}_p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{J}_p \end{pmatrix}, \tag{2.17}$$

where $\mathbf{J}_p \in \mathbb{R}^p$ denotes the vector of currents injected into the ports, and value s is a complex number with negative imaginary part, i.e., $s = -i\omega$. Since $\mathbf{V}_p \in \mathbb{R}^p$ is supposed to be given ($\mathbf{V}_p = \mathbf{V}_{\text{fixed}}$), the vector of currents at the ports, $\mathbf{J}_p$, can be found from the linear system (2.17). This exactly defines the relation in (2.1).

Construction of the RLC circuit

As we have seen, Y_p in (2.1) is frequency dependent. Fasterix approximates Y_p by an RLC circuit with frequency independent elements such as resistances, inductances, and capacitances. This circuit, when observed from its ports, will be equivalent to the interconnect structure up to a maximum predefined frequency. As soon as the circuit is defined, it is submitted to the circuit simulator PSTAR [2], where frequency and time domain analysis can be performed. In case of frequency analysis, the output is a list of nodal voltages for a number of frequencies. From these data Fasterix can compute the current density and the effects of radiation on the interconnect. Details of constructing the RLC circuit in Fasterix can be found in [20], [93] and are discussed also in Chapter 4.

2.2.3 Test structures of interconnect

In this subsection we introduce two examples of interconnect structures, which will be used as test structures in Chapter 4, where we discuss the performance of a model order reduction technique used in Fasterix. The reduced circuits constructed by Fasterix for these structures lead to unstable simulations in the time domain. Therefore, we will also use these examples to demonstrate our developed approaches, which overcome the problem of unstable simulations.

Two parallel striplines

The model consists of two printed striplines, which are parallel to each other. The striplines are 1 mm wide, and the length is 15 mm. The model has five ports: IN_1, OUT_1, IN_2, OUT_2 and OUT_3. For the maximum frequency 1 MHz, Fasterix generates a mesh with 10 elements, see Figure 2.4. Some elements have shape of quadrilaterals, and some elements are boundaries. To build an RLC circuit with frequency independent elements,

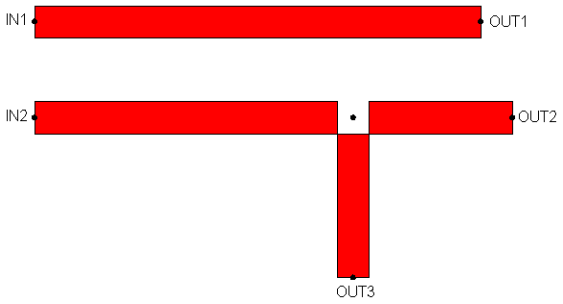

Figure 2.4: *Mesh for the two parallel striplines model generated for the maximum frequency 1MHz*

Fasterix chooses 6 particular elements which are marked by the black dots. These black dots are called *super nodes*. Roughly speaking, these super nodes will play a role of ports in the circuit. The way how the super nodes are chosen, and how an RLC circuit with frequency independent elements is constructed can be found in [20] and will be discussed in Chapter 4. Resulting RLC circuit will contain 60 resistors, 60 inductors, and 60 capacitors and it will be equivalent to the interconnect model up to 1 MHz.

Lowpass filter model

A piece of metal of approximately 10 mm length is shown in Figure 2.5. The structure has two ports, one is complete left (IN), and the other is on the right side (OUT). For

the maximum frequency 10 GHz, Fasterix generates a mesh with 257 elements and 452 common edges between each two neighboring elements. Resulting RLC circuit with frequency independent elements will be build on 98 quadrilateral elements marked by black dots and will contain 19012 resistors, 19012 inductors, and 4851 capacitors.

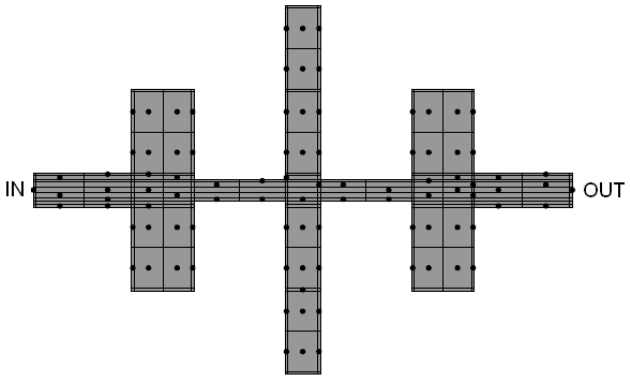

Figure 2.5: *Mesh of lowpass structure generated for the maximum frequency 10GHz*

2.3 Modeling of substrate

Semiconductor behaviour can be described by the semiconductor equations which are mostly relevant for the modeling of devices. Nevertheless, the global characteristics of the substrate can be captured with sufficient accuracy by taking into account dominant factors of the semiconductor behaviour. In this thesis the modeling approach is aimed at modeling the global behaviour of a homogeneous substrate and, therefore, does not take into account the full semiconductor equations.

Since describing the substrate via Maxwell's equations is unnecessary complicated, we consider the following assumptions [78]: 1) the fields are quasi-static, 2) the domain does not contain fixed charges or current sources, and 3) the domain is assumed to be purely resistive.

The quasi-static assumption means that the electric and magnetic fields are highly localized within the circuit elements [97]. Although the electric displacement $\mathbf{D}$ is dominant within a capacitor, it is negligible outside, so that Ampere's law (2.3) can neglect variations of $\mathbf{D}$ making the current divergence free, i.e.,

$$\nabla \cdot \mathbf{J} = 0. \tag{2.18}$$

This means that the algebraic sum of all currents flowing into or out of a node is zero,

which is Kirchhoff's current law. Similarly, variations of magnetic induction **B** in (2.2) is assumed negligible outside of inductors, so the electric field is curl free, i.e., $\nabla \times \mathbf{E} = 0$. In fact this is Kirchhoff's voltage law, i.e., the algebraic sum of voltage drops is zero, i.e.,

$$\mathbf{E} = -\nabla\phi, \tag{2.19}$$

where ϕ denotes the potential.

2.3.1 Mathematical formulation of the problem

An example of a substrate is shown in Figure 2.6. It consists of a domain Ω with a conductivity σ, and ideally conducting terminals (black quadrilateral elements) on top of the substrate. The boundary of the substrate includes terminal areas (Γ_1) and non-terminal area (Γ_2).

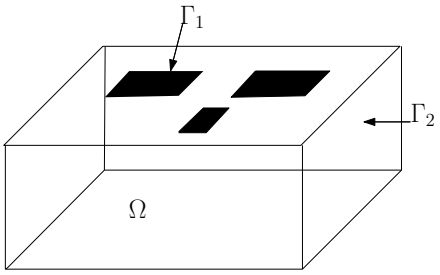

Figure 2.6: *A substrate with three contacts on the top.*

Under the above assumptions, the Maxwell's equations can be simplified to the following formulation. Let ϕ be the potential, σ be the conductivity, **J** be the current density vector, and **E** be the electric field vector. Then, a distributed formulation of Ohm's law can be written as

$$\mathbf{J} = \sigma\mathbf{E}. \tag{2.20}$$

Substituting (2.19) into (2.20) and substituting the result into (2.18), we obtain the following differential equation

$$\nabla \cdot (\sigma\nabla\phi) = 0. \tag{2.21}$$

If the conductivity in the domain is homogeneous, i.e., σ is constant, then (2.21) becomes the Laplace equation

$$\sigma\nabla^2\phi = 0. \tag{2.22}$$

Since we have to find a resistive network, which describes a homogeneous substrate (i.e. σ is constant), one has to solve the Laplace equation with appropriate boundary conditions. The boundary conditions are chosen such that the current can enter or leave the

domain through the contact areas, while remaining boundary has insulating properties. This requires to define Dirichlet boundary conditions on the contact areas, i.e.,

$$\phi = \bar{\phi} \quad \text{on} \quad \Gamma_1, \tag{2.23}$$

and homogeneous Neumann boundary conditions on the remaining boundary, i.e.,

$$\frac{\partial \phi}{\partial \mathbf{n}} = \bar{q} = 0 \quad \text{on} \quad \Gamma_2, \tag{2.24}$$

where $\mathbf{n}$ is the normal to the boundary $\Gamma = \Gamma_1 \bigcup \Gamma_2$ (note that $\Gamma_1 \bigcap \Gamma_2 = 0$). There is a connection between $\frac{\partial \phi}{\partial \mathbf{n}}$ and the normal component of the current density, J_n, through the contact:

$$J_n = \sigma \frac{\partial \phi}{\partial \mathbf{n}} \quad \text{on} \quad \Gamma_2. \tag{2.25}$$

Homogenous Neumann boundary condition implies that $J_n = 0$, i.e., no current flowing through the boundary Γ_2. Extraction of the network from the substrate requires to solve the Laplace equation with the above boundary conditions. In Chapter 6 we will consider two methods to solve the Laplace equation: boundary element method and finite element method. We will compare both methods on a qualitative level.

2.4 Electrical circuit modeling

2.4.1 Circuit Equations

A general electrical circuit can be modeled as a directed graph whose nodes correspond to the nodes of the circuit, and whose branches represent either simple wires or components like resistors, inductors, capacitors, diodes, and transistors. Additional to these components we consider the circuit sources. The topology of the circuit is described by means of the so-called incidence matrix A. The rows of the matrix A correspond to the branches of the circuit, and the columns correspond to the circuit nodes. By convention, a row has +1 in the corresponding source node, -1 in the destination node, and 0 everywhere else. A column which corresponds to a ground node has to be removed to avoid redundancy. A simple circuit with 3 nodes is shown in the Figure 2.7. The incidence matrix for this circuit is

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix}.$$

The columns of A are ordered according to the labeling of the nodes in Figure 2.7.

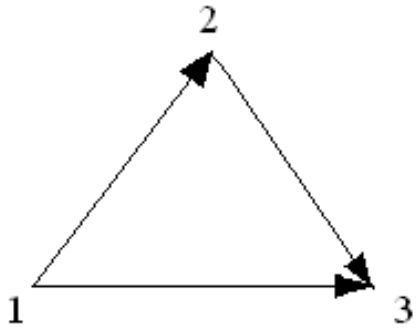

Figure 2.7: *Circuit with three nodes*

Kirchhoff's current law states that the sum of all branch currents entering and leaving a node is zero. Kirchhoff's voltage law states that the sum of all branch voltages along any closed loop in a circuit must be zero. To formulate them mathematically, let $\mathbf{i}_b$, $\mathbf{v}_b$, and $\mathbf{v}_n$ be the vectors of branch currents, branch voltages, and node voltages respectively. Then Kirchhoff's current and voltage laws read

$$A^T \mathbf{i}_b = 0, \tag{2.26}$$

$$A\mathbf{v}_n = \mathbf{v}_b. \tag{2.27}$$

To describe the system, we will formulate equations that describe the behaviour of the branch elements. We will consider RLC circuits consisting only five kinds of elements: resistors (conductances), inductors, capacitors, current, and voltage sources. We decompose the matrix A and the corresponding vectors as follows:

$$A = \begin{pmatrix} A_G \\ A_C \\ A_L \\ A_I \\ A_E \end{pmatrix}, \quad \mathbf{v}_b = \begin{pmatrix} \mathbf{v}_G \\ \mathbf{v}_C \\ \mathbf{v}_L \\ \mathbf{v}_I \\ \mathbf{v}_E \end{pmatrix}, \quad \mathbf{i}_b = \begin{pmatrix} \mathbf{i}_G \\ \mathbf{i}_C \\ \mathbf{i}_L \\ \mathbf{i}_I \\ \mathbf{i}_E \end{pmatrix}, \tag{2.28}$$

where the subscripts G, C, L, I and E denote conductance, capacitor, inductor, current source, and voltage source respectively. Then Kirchhoff's current law appears in the following way:

$$A_G^T \mathbf{i}_G + A_C^T \mathbf{i}_C + A_L^T \mathbf{i}_L + A_I^T \mathbf{i}_I + A_E^T \mathbf{i}_E = 0. \tag{2.29}$$

Kirchhoff's voltage is

$$A_G \mathbf{v}_n = \mathbf{v}_G, \quad A_C \mathbf{v}_n = \mathbf{v}_C, \quad A_L \mathbf{v}_n = \mathbf{v}_L, \quad A_I \mathbf{v}_n = \mathbf{v}_I, \quad A_E \mathbf{v}_n = \mathbf{v}_E. \tag{2.30}$$

Then we take into account the following branch constitutive relations, which hold for

all the components:

$$\mathbf{i}_G = \mathcal{G}\mathbf{v}_G, \quad \mathbf{i}_C = \mathcal{C}\frac{d}{dt}\mathbf{v}_C, \quad \mathbf{v}_L = \mathcal{L}\frac{d}{dt}\mathbf{i}_L, \quad \mathbf{i}_I = \mathbf{I}_t(t), \quad \mathbf{v}_E = \mathbf{V}_t(t), \tag{2.31}$$

where $\mathcal{G}$ is a diagonal matrix, which includes conductances, and $\mathcal{C}$ is a diagonal matrix, which includes capacitances. $\mathbf{I}_t(t)$ and $\mathbf{V}_t(t)$ denote the vectors of current-source and voltage-source values. $\mathcal{L}$ denotes a matrix of inductances. $\mathcal{L}$ is diagonal, if there is no inductive coupling. Inductive coupling adds off-diagonal terms, but the matrix remains symmetric and positive definite. We will describe the system by the use of $\mathbf{i}_L$, $\mathbf{i}_E$ and $\mathbf{v}_n$. Combining the Kirchhoff equations (2.29)–(2.30) with (2.31) and eliminating current unknowns (except $\mathbf{i}_L$ and $\mathbf{i}_E$) we get

$$A_G^T\mathcal{G}A_G\mathbf{v}_n + A_C^T\mathcal{C}A_C\frac{d}{dt}\mathbf{v}_n + A_L^T\mathbf{i}_L + A_I^T\mathbf{I}_t(t) + A_E^T\mathbf{i}_E = 0, \tag{2.32}$$

$$A_L\mathbf{v}_n - \mathcal{L}\frac{d}{dt}\mathbf{i}_L = 0, \tag{2.33}$$

$$A_E \cdot \mathbf{v}_n = \mathbf{V}_t(t). \tag{2.34}$$

This set of equations is called the Modified Nodal Analysis formulation (MNA) [43]. Multiplying the last two equations by -1, the MNA equations can be written in a compact matrix form:

$$G\mathbf{x} + C\frac{dt}{dt}\mathbf{x} = B\mathbf{u}(t), \tag{2.35}$$

where

$$G = \begin{pmatrix} A_G^T\mathcal{G}A_G & \mathcal{A}_L^T & \mathcal{A}_E^T \\ -\mathcal{A}_L & 0 & 0 \\ -\mathcal{A}_E & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} A_C^T\mathcal{C}A_C & 0 & 0 \\ 0 & \mathcal{L} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{2.36}$$

$$B = \begin{pmatrix} -A_I^T & 0 \\ 0 & 0 \\ 0 & I \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \mathbf{v}_n \\ \mathbf{i}_L \\ \mathbf{i}_E \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} \mathbf{I}(t) \\ \mathbf{V}(t) \end{pmatrix}.$$

Before investigating the properties of the above matrices, we introduce some definitions. Notation $\mathbf{x}^*$ denotes the conjugate transpose of vector $\mathbf{x}$.

Definition Matrix $A \in \mathbb{R}^{n\times n}$ is called *positive definite (positive semidefinite)*, if $\mathbf{x}^*A\mathbf{x} > 0$ ($\mathbf{x}^*A\mathbf{x} \geq 0$) for each $\mathbf{x} \in \mathbb{C}^n$ and $\mathbf{x} \neq 0$.

Definition Matrix $A \in \mathbb{R}^{n\times n}$ is called *positive real* (*nonnegative real*), if $Re(\mathbf{x}^*A\mathbf{x}) > 0$ ($Re(\mathbf{x}^*A\mathbf{x}) \geq 0$) for all $\mathbf{x} \in \mathbb{C}^n$ and $\mathbf{x} \neq 0$, where $Re(\mathbf{z})$ stands for the real part of $\mathbf{z}$.

We will assume that the values of resistors, inductors, and capacitors in the electrical circuit are nonnegative. Then we notice that C in (2.36) is positive semidefinite. We will

show that G in (2.36) is nonnegative real:

$$\mathbf{x}^* G\mathbf{x} = \begin{pmatrix} \mathbf{x}_1^* & \mathbf{x}_2^* \end{pmatrix} \begin{pmatrix} A_G^T \mathcal{G} A_G & P^T \\ -P & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} =$$

$$\mathbf{x}_1^*(A_G^T \mathcal{G} A_G)\mathbf{x}_1 - \mathbf{x}_2^* P^T \mathbf{x}_1 + \mathbf{x}_1^* P \mathbf{x}_2 = \mathbf{x}_1^*(A_G^T \mathcal{G} A_G)\mathbf{x}_1 + 2iIm(\mathbf{x}_1^* P^T \mathbf{x}_2),$$

where $P = \begin{pmatrix} \mathcal{A}_L^T & \mathcal{A}_E^T \end{pmatrix}$, and $Im(\mathbf{z})$ stands for the imaginary part of $\mathbf{z}$. Thus, $Re(\mathbf{x}^* G\mathbf{x}) = Re(\mathbf{x}_1^*(A_G^T \mathcal{G} A_G)\mathbf{x}_1) = Re(\mathbf{y}^* \mathcal{G}\mathbf{y}) \geq 0$.

The above properties of C and G are important to establish the properties of the circuit, such as stability and passivity, which will be discussed in Section 2.4.4.

There are other methods to formulate circuit equations. For instance the Sparse Tableau formulation [39] can also be used. This formulation is less compact than MNA, but the matrices are sparser. In this case, the vector of unknowns includes the node voltages, branch currents, and branch voltages.

2.4.2 Properties of the circuit equations

We will consider a linear time invariant differential algebraic equation (2.35) which describes the circuit:

$$C\frac{d}{dt}\mathbf{x}(t) + G\mathbf{x}(t) = B\mathbf{u}(t), \tag{2.37}$$

with initial conditions:

$$\mathbf{x}(t_0) = \mathbf{x}_0. \tag{2.38}$$

In general, C may be singular. Thus (2.37) becomes a differential algebraic equation (DAE). In this case, the linear time invariant system is called a descriptor system.

If $C \in \mathbb{R}^{n\times n}$ is non-singular, (2.37) is an ordinary differential equation (ODE). The general solution of (2.37) has the form [10]:

$$\mathbf{x}(t) = e^{-C^{-1}G(t-t_0)}\mathbf{x}_0 + \int_{t0}^{t} e^{-C^{-1}G(t-\tau)}C^{-1}B\mathbf{u}(\tau)d\tau. \tag{2.39}$$

A possible option to analyze the circuit behaviour is to solve the initial value problem (2.37)-(2.38) by using classical numerical methods for ODEs, for instance, the Euler method or multistep methods [15]. However, there is another method that is particularly well suited for these problems. This method is based on the Laplace transform. Moreover, this method provides more information on properties of stability and passivity via

the analytic transfer function of the system. The Laplace transform is defined as:

$$L[f(t)](s) = \int_0^\infty f(t)e^{-st}dt.$$

If we apply the Laplace transform to (2.37) and assume the initial condition $\mathbf{x}_0 = 0$, the system is then transformed to a purely algebraic system of equations:

$$(G + sC)X(s) = BU(s), \tag{2.40}$$

or

$$X(s) = (G + sC)^{-1}BU(s),$$

where $X(s) = L[\mathbf{x}(t)](s)$ and $U(s) = L[\mathbf{u}(t)](s)$ denote the Laplace transformed variables. The differential equation in time becomes an algebraic equation in the complex frequency variable. The inverse Laplace transform of $X(s)$ represents the time-domain solution of the circuit. Let a linear network be described by (2.37) and let output of the network be of the form

$$\mathbf{y}(t) = L^T\mathbf{x}(t). \tag{2.41}$$

Applying the Laplace transform to (2.41), one obtains

$$Y(s) = L^T X(s). \tag{2.42}$$

Combining (2.42) with (2.40) and eliminating the space vector $X(s)$, one obtains a relation between the input $U(s)$ and the output $Y(s)$:

$$Y(s) = L^T(G + sC)^{-1}BU(s).$$

Now we set

$$H(s) = L^T(G + sC)^{-1}B. \tag{2.43}$$

This is the so-called *transfer function*. If L and B are vectors (single-input-single-output system), then $H(s)$ is a scalar function of s. If the system has more than one input and more than one output (multi-input-multi-output system), then B and L have more than one column, respectively. This makes $H(s)$ a matrix function. The (i, j)-th entry in $H(s)$ denotes the transfer from input i to output j. Transfer function, $H(s)$, allows to predict the zero-state response to any excitation by multiplying this function with the Laplace transform of excitation.

2.4.3 Poles and residues

We will show that (2.43) can be presented in the pole-residue form.

The pair of matrices (C, G) is called a *pencil*. A pencil is called *regular*, if there is at least

one $s \in \mathbb{C}$ for which $det(G + sC) \neq 0$. Suppose the pencil (C, G) is regular. If G is nonsingular, then

$$(G + sC) = G(I + sG^{-1}C).$$

We assume that an eigenvalue decomposition $-G^{-1}C = S\Lambda S^{-1}$ exists, where Λ is diagonal matrix with eigenvalues λ_i. Thus

$$(G + sC) = G(I - sS\Lambda S^{-1}) = GS(I - s\Lambda)S^{-1}. \tag{2.44}$$

If G is singular, then there exists a scalar s_0 such that $G + s_0C$ is nonsingular. Substituting (2.44) into (2.43), one obtains

$$H(s) = L^T S(I - s\Lambda)^{-1}S^{-1}G^{-1}B.$$

Note that $I - s\Lambda$ is the diagonal matrix with elements $1 - s\lambda_i$, $i = 1, \ldots, n$ on the diagonal; therefore, the inverse of it can be easily computed. Let L and B have only one column, i.e. correspond to a single-input single output system. Thus,

$$H(s) = \sum_{k=1}^{n} \frac{a_k b_k}{1 - s\lambda_i} = \sum_{k=1}^{n} \frac{r_k}{s - p_k}, \tag{2.45}$$

where $a_k = (L^T S)_k$ and $b_k = (S^{-1}G^{-1}B)_k$. The expression in (2.45) is called the *pole-residue* representation of the transfer function. The poles p_k and the residues r_k are

$$p_k = \frac{1}{\lambda_k}, \quad r_k = -\frac{a_k b_k}{\lambda_k}.$$

In case of a multi-input multi-output system with m_1 inputs and m_2 outputs, the residues r_k are $m_1 \times m_2$ matrices. Moreover, each entry in the transfer function will have the same poles.

Note that poles p_k of the transfer function are a subset of the generalized eigenvalues of the matrix pencil (C, G). For example, if G is symmetric positive definite and C is symmetric positive semidefinite, then $p_k \in R$. Poles are a general property of the network, while residuals are related to the input and output matrices, i.e., B and L. Residues can be rewritten by defining the right and left eigenvectors $\mathbf{v}_i$, $\mathbf{w}_i \in \mathbb{C}^n$ of the matrix pencil (C, G):

$$-G\mathbf{v}_i = p_i C\mathbf{v}_i, \quad \mathbf{v}_i \neq 0,$$
$$-\mathbf{w}_i^* G = p_i \mathbf{w}_i^* C, \quad \mathbf{w}_i \neq 0.$$

Thus, residues can be defined as

$$r_i = (L^T \mathbf{v}_i)(\mathbf{w}_i^* B).$$

If both G and C are singular, then the transfer function has the form [52]:

$$H(s) = r_\infty + d + \sum_{i=1}^{\widetilde{n}} \frac{r_i}{s - p_i}, \tag{2.46}$$

where r_∞ is the constant contribution of the poles at infinity (often zero), and the d term comes from the descriptor system. $\widetilde{n} \leq n$ is the number of finite first order poles.

2.4.4 Stability and passivity

Stability is an important property of linear dynamical systems. In a stable system the output signal remains limited. We consider a linear dynamical system

$$C\frac{d}{dt}\mathbf{x} = -G\mathbf{x} + B\mathbf{u}, \tag{2.47}$$

$$\mathbf{y} = L^T\mathbf{x}. \tag{2.48}$$

The system (2.47) and (2.48) is called *stable* if the solution $\mathbf{x}(t)$, $t > 0$ of $C\frac{d\mathbf{x}}{dt} = -G\mathbf{x}$ with an initial condition $\mathbf{x}(0) = \mathbf{x}_0$ remains bounded as $t \to \infty$ for any initial vector $\mathbf{x}$.

There is another treatment of stability in terms of matrix pencil (C, G). The system is stable if, and only if, the real part of all finite generalized eigenvalues p of the matrix pencil (C, G) is less or equal than zero, i.e., $Re(p) \leq 0$, and all finite eigenvalues p of the matrix pencil (C, G) with $Re(p) = 0$ are simple.
As we discussed before, matrix C from MNA formulation (2.35) is positive semidefinite, and G is non-negative real. The following theorem, which can be also found in [42], is helpful to prove that the corresponding system is stable.

Theorem 1. *Let G be a nonnegative real matrix, (i.e., $Re(\boldsymbol{x}^* G\boldsymbol{x}) \geq 0$ for each $\boldsymbol{x} \in \boldsymbol{C}^n$, $\boldsymbol{x} \neq 0$) with at most one zero eigenvalue, and let C be positive semidefinite, i.e. $C \geq 0$. Assume that the pencil (C, G) is regular. Then the system (2.47)–(2.48) is stable.*

Proof. We consider the generalized eigenvalue problem:

$$G\mathbf{x} = pC\mathbf{x}. \tag{2.49}$$

We may assume that the eigenvalue decomposition of G exists, i.e. $G = EDE^{-1}$. Substituting it into (2.49) and setting up $\mathbf{y} = E^{-1}\mathbf{x}$, one obtains

$$D\mathbf{y} = pE^{-1}CE\mathbf{y}.$$

Multiplying from the left by $\mathbf{y}^*$, one obtains

$$\mathbf{y}^* D\mathbf{y} = p\mathbf{y}^* E^{-1} CE\mathbf{y}.$$

Matrix D is diagonal with positive real eigenvalues, and $E^{-1}CE$ contains nonnegative real eigenvalues. Since the matrix pencil (C, G) is regular, $\mathbf{y}^* D\mathbf{y}$ and $\mathbf{y}^* E^{-1} CE\mathbf{y}$ cannot both be zero. If $\mathbf{y}^* D\mathbf{y}$ and $\mathbf{y}^* E^{-1} CE\mathbf{y}$ are nonzero, then the generalized eigenvalue p has positive real part. If it happens that $\mathbf{y}^* D\mathbf{y} = 0$, then $p = 0$. If it happens that $p\mathbf{y}^* E^{-1} CE\mathbf{y} = 0$, then $p = \infty$. Thus the system is stable. □.

Often reduced-order modeling is applied to large linear subcircuits which contain large amount of resistors, inductors and capacitors. After reduction, such subcircuits are usually used for simulation of the whole system. To guarantee stability of the whole system, each subcircuit has to be *passive*. Passivity is stronger than stability and means that a circuit does not generate energy. Passivity is closely related to positive realness of the transfer function. A system (2.47) and (2.48) is passive if and only if its transfer function is *positive real*:

1. $H(s)$ is analytic for s with $Re(s) > 0$,
2. $H(s)^* = H(\bar{s})$, for s with $Re(s) > 0$,
3. $H(s) + H^*(s) \geq 0$ for s with $Re(s) > 0$, i.e., $H(s) + H^*(s)$ is nonnegative definite for $Re(s) > 0$.

Note that $\bar{s}$ denotes the complex conjugate of s, and $*$ denotes the complex conjugate transpose. The first condition means that $H(s)$ has no right-half plane poles (i.e., there are no poles p with $Re(p) > 0$). The second condition is always satisfied for the transfer function of the form (2.43) since it involves only real scalars (except for s). The third condition is usually difficult to check since it requires special matrix manipulations.

In the previous theorem we have shown how the properties of the matrices G and C can be exploited to prove stability of the system. The following theorem demonstrates how the properties of these matrices can be useful for positive realness of the transfer function.

Theorem 2. *Let $G, C \in \mathbb{R}^{N\times N}$, and $B \in \mathbb{R}^{N\times m}$. Assume that G is nonnegative real (i.e., $Re(\boldsymbol{x}^* G\boldsymbol{x}) \geq 0$ for each $\boldsymbol{x} \in \boldsymbol{C}^n$, $\boldsymbol{x} \neq 0$), and C is positive semidefinite, i.e., $C = C^T \geq 0$, and that (C, G) is a regular matrix pencil. Then, the transfer function $H(s) = L^T(G + sC)^{-1}B$, $s \in C$, is positive real.*

A proof of the theorem can be found in [27]. This theorem is a sufficient condition of positive realness and it implies that the circuit defined by (2.35) is passive.

2.4.5 DC, AC, and transient analysis

An important type of circuit analysis is the direct current (DC) analysis, i.e., solution of the circuit equation (2.37) that does not vary with time (the input $\mathbf{u}(t)$ is assumed to be a constant value). The DC equations are formed from (2.37) by assuming that $\frac{d}{dt}\mathbf{x}(t) = 0$ for all t. Thus the circuit simulator solves $G\mathbf{x} = B\mathbf{u}$ for $\mathbf{x}$, which is a DC solution.

In circuit simulation one is often interested in the sinusoidal steady-state behaviour of circuits, i.e., the circuits are excited by a sinusoidal source, and we are looking for the solution $\mathbf{x}(t)$ that is reached after the effects of the initial conditions have been passed. This is known as alternating current or AC analysis. For this purpose, one considers the circuit equations in general form, and for simplicity we suppose that we have only one input at fixed frequency ω, e.g., $\mathbf{u} = \theta \sin(\omega t + \phi)$. θ is an amplitude, and ϕ is a phase. Thus $\mathbf{u}$ can be rewritten as the real part of a complex number:

$$\theta \sin(\omega t + \phi) = Re(\theta e^{i\omega t + \phi}) = Re(\Theta e^{i\omega t}). \tag{2.50}$$

Substituting (2.50) into (2.47), leads to the system

$$G\mathbf{x}(t) + C\frac{d}{dt}\mathbf{x}(t) = B\ Re(\Theta e^{i\omega t}). \tag{2.51}$$

Similarly, we consider a companion system, which differs from the previous one by its excitation:

$$G\mathbf{y}(t) + C\frac{d}{dt}\mathbf{y}(t) = B\ Im(\Theta e^{i\omega t}). \tag{2.52}$$

Summing up (2.51) and (2.52), we can formulate the system in a complex variable $\mathbf{z}(t) = \mathbf{x}(t) + i\mathbf{y}(t)$:

$$G\mathbf{z}(t) + C\frac{d}{dt}\mathbf{z}(t) = B\Theta e^{i\omega t}. \tag{2.53}$$

We seek solutions $\mathbf{z}(t)$ in the form

$$\mathbf{z}(t) = Z(\omega)e^{i\omega t}.$$

Substituting this solution in (2.53) leads to

$$G(Ze^{i\omega t}) + C\frac{d}{dt}(Ze^{i\omega t}) = B\Theta e^{i\omega t}.$$

After differentiation of the second term, one obtains

$$(G + i\omega C)Ze^{i\omega t} = B\Theta e^{i\omega t}.$$

Eliminating $e^{i\omega t}$, and solving the linear system, it follows that

$$Z(w) = (G + i\omega C)^{-1} B\Theta. \tag{2.54}$$

Therefore, the circuit solution is available as

$$\mathbf{x}(t) = Re(\mathbf{z}(t)) = Re(Ze^{i\omega t}).$$

Often designers want to study the response of the circuit over a range of frequencies. In this case the system (2.54) has to be solved for various values of ω.

Transient analysis is done by finding the solution of the circuit system (2.37) for the state vector $\mathbf{x}(t)$, at each time interval, over time period. If C is nonsingular, then one deals with ODE. Analytical solution of ODE with the initial condition (2.38) has a form (2.39). In case C is singular, one deals with a DAE, and consistent initial conditions with the DAE have to be found. Reduction of DAE to ODE is not a practical option for large circuits because the sparsity of C is typically destroyed by the required matrix decomposition, used for transforming the DAE into an ODE. Consequently, DAEs are numerically solved in their original form. The discussion of the required numerical methods are out of the scope of this thesis and can be found, for instance, in [10].

2.4.6 Circuit synthesis

In this section we will discuss how the circuit model (2.47)–(2.48) can be translated into an equivalent circuit. This procedure is called circuit synthesis. We will describe a synthesis in terms of admittance matrix (Y-parameters) since admittance matrix will be used later on in this thesis to synthesize reduced order models. Other methods for synthesis can be found, for instance, in [37] (synthesis in terms of impedance) and [42] (synthesis with controlled sources).

When the system (2.47)–(2.48) has input $\mathbf{u}(t)$ of voltages, and output $\mathbf{y}(t)$ of currents, then the obtained transfer function is called admittance function, or admittance. First, we will illustrate the Foster realization, on the following admittance matrix:

$$Y(s) = \frac{r_1}{s - p_1} + sr_2 + r_3, \tag{2.55}$$

where $r_1, r_2, r_3, p_1 \in \mathbb{R}$ are real residues and pole, respectively. Since the admittance of an RLC circuit shown in Figure 2.8 is given by

$$Y_{RLC}(s) = \frac{1}{R_1 + sL} + sC + \frac{1}{R_2} = \frac{\frac{1}{L}}{s + \frac{R_1}{L}} + \frac{1}{R_2},$$

the so-called Foster realization of (2.55) is

$$L = \frac{1}{r_1}, \quad R_1 = -\frac{p_1}{r_1}, \quad C = r_2, \quad R_2 = \frac{1}{r_3}.$$

If $r_1 > 0$, $r_2 > 0$, $r_3 > 0$ and $p_1 < 0$, then all elements in the realization are positive, and $Y(s)$ is positive real and describes a passive circuit.

Now we consider an admittance function with extra complex pole and residue:

$$Y(s) = \frac{r_1}{s - p_1} + sr_2 + r_3 + \left(\frac{r_4}{s - p_4} + \frac{\overline{r_4}}{s - \overline{p_4}} \right), \tag{2.56}$$

where $r_1, r_2, r_3\ p_1 \in \mathbb{R}$, and $r_4, p_4 \in \mathbb{C}$ are complex pole and residue. $\overline{p}_4$ and $\overline{r}_4$ denote their complex conjugates, respectively. Let $r_4 = \nu + i\mu$ and $p_4 = \alpha + i\beta$, then the Foster realization of (2.56) is an RLCG circuit, i.e., circuit consisting of resistor, inductor, capacitor, and conductance shown in Figure 2.9 with the following elements:

$$R_0 = \frac{1}{r_3}, \quad C_0 = r_2,$$

$$R_r = -\frac{p_1}{r_1}, \quad L_r = \frac{1}{r_1},$$

$$L_c = \frac{1}{2\nu}, \quad R_c = 2L_r, (L_r(\nu\alpha + \mu\beta) - \alpha), \quad G_c = -2L_cC_c(\nu\alpha + \mu\beta).$$

Multi-port symmetrical admittance matrix can be synthesized by the Foster realization as described in [85], [57], [58]. We will illustrate it on the example of a 2×2 admittance matrix

$$Y(s) = \begin{pmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{pmatrix}.$$

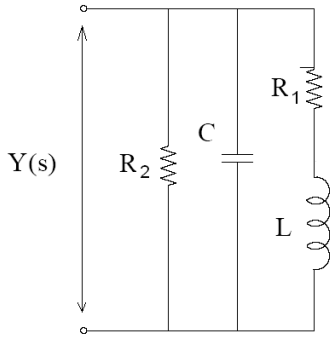

Figure 2.8: *Foster realization of the one-port admittance $Y(s)$ in (2.55).*

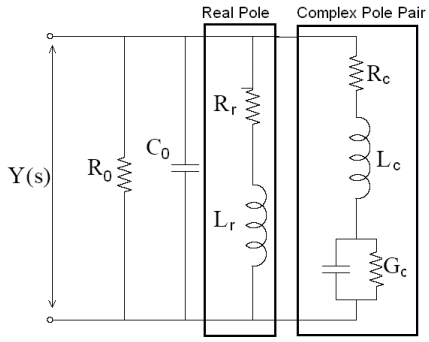

Figure 2.9: *Foster realization of the one-port admittance $Y(s)$ in (2.56).*

Such admittance can be realized by the use of Π-structure template shown in Figure 2.10, where each branch is realized by the one-port admittance. Based on the example of the 2×2 admittance matrix, the realization procedure can be extended to a k-port case.

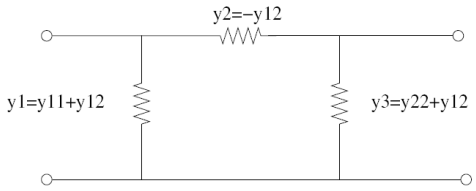

Figure 2.10: *A two-port realization.*

We note that the Foster realization of a k-port admittance matrix with p poles leads to a circuit with $O(k^2p)$ elements. In model order reduction one is interested not only in reducing the dimension of a system but also in decreasing the amount of circuit elements. Therefore, circuit synthesis is a crucial feature in model order reduction, and it will be used in the next chapters of this thesis.

2.5 Concluding remarks

In this chapter, we discussed existing general methods used for modeling of interconnect and substrate. In particular, we have reviewed method for modeling of interconnect used in the tool Fasterix, and we have presented the test structures which are further used in the thesis. We also presented a mathematical formulation for modeling of homogeneous substrate. The important observation here is that modeling of a homogeneous substrate requires to solve the Laplace equation. Since the result of extraction

of both, interconnect and substrate, is an electrical circuit consisting of resistors, inductors and capacitors, we have shown the main properties of the circuit such as stability, passivity, and important types of circuit analysis. We also have shown an idea of circuit synthesis by Foster realization which is a key aspect of model order reduction extensively used in the further Chapters.

Chapter 3

Model Order Reduction

Numerical simulation and analysis of large scale systems which arise, for instance, in the design of electrical circuits may be extremely expensive. The constructed state-space models may contain thousands of equations, making the system impossible to solve even when using modern technologies. In some other cases, the systems may be not as large for making simulations unfeasible; however, the models may need to be solved many times while varying system inputs. In such cases, compact representations of dynamical systems may be a point of great interest. The task of replacing a given dynamical system by a smaller model (with reduced complexity) is called model order reduction (MOR). In this chapter, we introduce the mathematical formulation of model order reduction for linear systems. We start with the projection framework along with several common approaches for the construction of projection basis. We discuss existing reduction techniques for multi-terminal networks, mainly concentrating on the reduction method used in the electromagnetic tool Fasterix and in methods for reduction of large resistor networks.

3.1 Model Order Reduction of linear systems

Let us consider an m-input, p-output time invariant system of the form

$$C\frac{d}{dt}\mathbf{x}(t) = -G\mathbf{x}(t) + B\mathbf{u}(t), \tag{3.1}$$

$$\mathbf{y}(t) = L^T\mathbf{x}(t), \tag{3.2}$$

where $C, G \in \mathbb{R}^{N\times N}$, $B \in \mathbb{R}^{N\times m}$, $L \in \mathbb{R}^{N\times p}$ are given matrices, $\mathbf{x}(t) \in \mathbb{R}^N$ is the state vector, $\mathbf{u}(t) \in \mathbb{R}^m$ is the vector of inputs, $\mathbf{y}(t) \in \mathbb{R}^p$ is the output vector, N is the state space dimension, m and p are the number of inputs and outputs, respectively.

When the order, N, of the system (3.1)–(3.2) is too large, solving various control problems within a reasonable computing time may be unfeasible or too expensive when performed often. Therefore, one can consider the possibility of approximating the system by a so-called reduced order system of order, n, with the condition $n \ll N$. We can denote such reduced order system by

$$\widetilde{C}\frac{d}{dt}\widetilde{\mathbf{x}}(t) = -\widetilde{G}\widetilde{\mathbf{x}}(t) + \widetilde{B}\mathbf{u}(t), \tag{3.3}$$

$$\widetilde{\mathbf{y}}(t) = \widetilde{C}^T\widetilde{\mathbf{x}}(t), \tag{3.4}$$

with $\widetilde{C}, \widetilde{G} \in \mathbb{R}^{n\times n}$, $\widetilde{B} \in \mathbb{R}^{n\times m}$, $\widetilde{L} \in \mathbb{R}^{n\times p}$. The system has the same input, $\mathbf{u}(t)$, as the original system but different output, $\widetilde{\mathbf{y}}(t)$. The goal of MOR is to find a reduced order model with the condition $n \ll N$, in such a way that the behaviour of the reduced order model is sufficiently close to that of the original system.

For instance, in the time domain, if we use the same input $\mathbf{u}(t)$ for both systems, we would require $\widetilde{\mathbf{y}}(t)$ to be closer to $\mathbf{y}(t)$. In other words, we require

$$||\mathbf{y} - \widetilde{\mathbf{y}}||$$

to be small for a certain norm. In the frequency domain, this is equivalent to imposing conditions on the frequency responses of both systems, i.e., we need to find a reduced order model such that the difference of the transfer functions of both models

$$||H(s) - \widetilde{H}(s)||, \tag{3.5}$$

is minimal for a given criterion. Here $H(s) = L^T(G + sC)^{-1}B$ and $\widetilde{H}(s) = \widetilde{L}^T(\widetilde{G} + s\widetilde{C})^{-1}\widetilde{B}$.

With these preliminaries, we can state that that the main problem is the following: given a large-scale linear time invariant system, rapidly compute an accurate reduced order model. Reduced means that the dimension, n, of the reduced order model is significantly smaller than the dimension of the original model, N. In addition to that, the cost of generating the reduced model should be significantly smaller than the cost of directly solving the original system the required amount of times. The reduced model should provide a reasonably accurate approximation of the original one. These conditions (accuracy, speed, dimension) can be conflicting goals. Depending on the application, it might also be very important to preserve certain properties of the system, such as stability and passivity, in order to make the reduced model physically insight full. In circuit simulation, the preservation of passivity is crucial, for the reduced models to be suitable

for their use in combination with other networks. Because a non-passive circuit can lead to unstable simulations when combined with other circuits, even in the case when all circuits are stable.

3.1.1 Projection framework for linear systems

Projection based model order reduction methods for a linear dynamical system (3.1)–(3.2) are based on approximating the solution $\mathbf{x}$ in a low-dimensional subspace, $\mathbf{x} = V\hat{\mathbf{x}}$, and reducing the number of equations with a left projection matrix W, in the following way

$$W^*CV\frac{d}{dt}\hat{\mathbf{x}} = -W^*GV\hat{\mathbf{x}} + W^*B\mathbf{u}, \tag{3.6}$$

$$\mathbf{y} = L^TV\hat{\mathbf{x}}, \tag{3.7}$$

is a reduced order system. Defining

$$\widetilde{C} = W^*CV \in \mathbb{R}^{n\times n}, \quad \widetilde{G} = W^*GV \in \mathbb{R}^{n\times n}, \tag{3.8}$$

$$\widetilde{B} = W^*B \in \mathbb{R}^{n\times m}, \quad \widetilde{L}^T = L^TV \in \mathbb{R}^{p\times n}, \tag{3.9}$$

one can construct a reduced system

$$\widetilde{C}\frac{d}{dt}\tilde{\mathbf{x}}(t) = -\widetilde{G}\tilde{\mathbf{x}}(t) + \widetilde{B}\mathbf{u}(t), \tag{3.10}$$

$$\tilde{\mathbf{y}}(t) = \widetilde{L}^T\tilde{\mathbf{x}}(t), \tag{3.11}$$

which consists of n equations and n unknowns. In the next subsections we will discuss several approaches for constructing the projection matrices V and W.

3.1.2 Modal approximation

The modal approximation technique is probably one of the first model reduction techniques developed for linear systems. Due to it simplicity it is still popular for some applications. Here we will explain the main idea of modal approximation. Let $H(s) = L^T(G + sC)^{-1}B$ be the transfer function of (3.1)–(3.2). Doing some algebra as in Section 2.4.3, one can rewrite the transfer function as a partial fraction expansion:

$$H(s) = \sum_{i=1}^{n} \frac{R_i}{s - \lambda_i}. \tag{3.12}$$

A general framework for modal approximation of the transfer function (3.12) is the following [48], [34]

1. Compute the poles λ_i and corresponding left and right eigenvectors $\mathbf{y}_i$ and $\mathbf{x}_i$;
2. Sort (λ_i, R_i) in decreasing $\frac{|R_i|}{|Re(\lambda_i)|}$ order;
3. Truncate at $\frac{|R_i|}{|Re(\lambda_i)|} < R_{min}$
4. Construct

$$H_k(s) = \sum_{i=1}^{k} \frac{R_i}{s - \lambda_i}. \tag{3.13}$$

Thus the idea behind the modal approximation is to take the part of the transfer function with the poles that are the closest to the imaginary axis and to throw away the others. It is also possible to consider modal approximation as a case of projection based framework. By taking $W = [\mathbf{y}_1, \ldots, \mathbf{y}_k]$, $V = [\mathbf{x}_1, \ldots, \mathbf{y}_k]$, the reduced order model can be defined according to (3.8)–(3.11).

If n is large, the step 1 of this framework is in general not feasible since full space eigenmethods such as QR and QZ [32] have time complexity $O(n^3)$. In practical situations an alternative for step 1, is the Subspace Accelerated DPA (SADPA) [72], [73]. SADPA computes k ($k \ll n$) most dominant poles and corresponding eigenvectors in an iterative way. The accuracy of the approximation can be controlled by the stopping convergence to additional poles when the relative error $\frac{||H(i\omega_i) - H_k(j\omega_i)||}{||H(j\omega_i)||}$ (measured over several frequencies ω_i) is smaller than a specified tolerance tol. [49]. An advantage of modal approximation is that the most dominant poles and residues are exactly those of the original system and, therefore, stability is preserved.

3.1.3 Moment matching

Generally speaking, moment matching can refer to any projection technique wherein projections are constructed in a way to ensure that the reduced model matches some number of values and derivatives of the transfer function of the original model at a prescribed set of frequencies. In this section we present the basic idea of moment matching using projection vectors.

The transfer function, $H(s) = L^T(G + sC)^{-1}B$, can be expanded in series around s_0 as

$$H(s) = \sum_{i=0}^{\infty} m_i (s - s_0)^i,$$

where $m_i = -L^T(G+s_0C)^{-i}C^i(G+s_0C)^{-1}B$ is the i-th moment of the transfer function at $s = s_0$ (or the i-th derivative at $s = s_0$). Since direct computation of moments is an ill-conditioned problem [77], implicit projection techniques based, for instance, on the Arnoldi method [81] and Lanczos method [35] can be used to make moment matching numerically robust. These methods use a Krylov subspace. A k-dimensional Krylov subspace corresponding to some matrix A and vector $\mathbf{v}$ is denoted as $\mathcal{K}_k(A,\mathbf{v})$ and is defined as

$$\mathcal{K}_k(A,\mathbf{v}) = span(\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}),$$

where *span* stands for the vector space generated by $\mathbf{v}, A\mathbf{v}, \ldots, A^{k-1}\mathbf{v}$. For instance, the Lanczos method can be considered as an approach for constructing biorthogonal matrices V and W, i.e., $W^*V = I$, that satisfy

$$colsp\{V\} = \mathcal{K}_k(A,\mathbf{v}_1), \quad colsp\{W\} = \mathcal{K}_k(A^T,\mathbf{w}_1),$$

where $colsp\{A\}$ denotes the vector space spanned by the columns of A; $\mathbf{v}_1$ and $\mathbf{w}_1$ are user-defined starting vectors.

A basis for a Krylov subspace can be quickly computed if A can be applied to $\mathbf{v}$ in a fast way, for instance, when A is sparse. This gives Krylov-based model reduction the potential for cost savings. However, to obtain the best reduced order model, one has to specify a right Krylov subspace. In other words, the form of the projection matrices V and W influence on the quality of approximation and costs of obtaining the reduced model. The following Theorem gives insight on how to choose the Krylov subspaces. The proof of the Theorem can be found, for instance, in [35].

Theorem 3. *If the columns of V span $\mathcal{K}_{k1}((G+s_0C)^{-1}C,(G+s_0C)^{-1}B)$ and the columns of W span $\mathcal{K}_{k2}((G+s_0C)^{-T}C^T,(G+s_0C)^{-T}L^T)$, then the reduced order transfer function $\widetilde{H}(s) = \widetilde{L}^{-1}(s\widetilde{C}+\widetilde{G})^{-1}\widetilde{B}$ matches the first k_1+k_2 moments of the original transfer function $H(s) = L^T(G+sC)^{-1}B$ about $s = s_0$.*

The general framework for moment matching via Krylov method is

1. Compute bases V and W such that their span correspond to the Krylov subspaces:

$$\mathcal{K}_m((G+s_0C)^{-1}C,(G+s_0C)^{-1}B),$$

$$\mathcal{K}_m((G+s_0C)^{-T}C^T,(G+s_0C)^{-T}L^T).$$

2. Use the projection matrices to obtain an mth order reduced order model as in (3.10)-(3.11).

All together, it is possible to match a total of $2m$ moments in the reduced model. Even though moment matching guarantees local accuracy of the reduced transfer function, it

does not guarantee global accuracy. In other words, there are no constraints on the behaviour of the reduced model far away from the expansion points about which moments are matched.

3.1.4 Balanced truncation

Balanced truncation, also referred to as truncated balanced realization (TBR), is a popular model order reduction approach for state space systems developed in the control community [65]. In TBR, projections V and W are chosen to correspond to the so-called Hankel singular vectors and corresponding Hankel singular values. The idea of TBR is to perform a change of coordinates $\widetilde{\mathbf{x}} = T\mathbf{x}$, where the states $\widetilde{\mathbf{x}}$ are ordered from the most important to least important, then by truncating the least important states, we can obtain the reduced model. A coordinate transformation, combined with a truncation can be viewed as a projection. We will consider balanced truncation for a system in the form

$$\frac{d}{dt}\mathbf{x} = A\mathbf{x} + B\mathbf{u}, \tag{3.14}$$

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}. \tag{3.15}$$

Such an ODE system is often referred as to the system (A, B, C, D). Let (A, B, C, D) be an nth order stable system. The first step in balanced truncation is to compute the unique positive definite solutions of the Lyapunov equations

$$PA^T + AP + BB^T = 0, \tag{3.16}$$

$$QA + A^TQ + C^TC = 0. \tag{3.17}$$

P and Q are the so-called controllability and observability grammians. Given P and Q, a balancing transformation T can be computed as $T = \Sigma^{1/2}U^TR^{-T}$, where $P = R^TR$ is the Cholesky factorization, and $U\Sigma^2U^T$ is the SVD of RQR^T. The state space transformation $\widetilde{\mathbf{x}} = T\mathbf{x}$ is a balancing transformation, i.e., P and Q are diagonal matrices such that,

$$P = T^{-1}P(T^{-1})^T = \Sigma,$$

$$Q = T^TQT = \Sigma.$$

The balanced system

$$\left(T^{-1}AT, T^{-1}B, CT, D\right) = \left(\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \begin{pmatrix} C_1 \\ C_2 \end{pmatrix}, D\right), \tag{3.18}$$

where $A_{11} \in R^{k\times k}$, $B_1 \in R^{k\times m}$, $C_1 \in R^{k\times p}$ and $k < n$ can be truncated up to required accuracy by using the bound

$$||H - \widetilde{H}||_\infty \leq 2 \sum_{i=k+1}^{m} \sigma_i,$$

where $\sigma_i = (\lambda_i(PQ))^{1/2}$ are the Hankel singular values. Here λ_i denotes i-th eigenvalue of PQ. The truncated reduced k-order system is

$$(\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}) = (A_{11}, B_1, C_1, D).$$

Although TBR provides a global error bound, it is rarely used for very large scale problems because the computational complexity required to solve the Lyapunov equations is $O(N^3)$. A possibility to apply balanced truncation to large sparse systems by combining SVD-type methods with rational Krylov approaches has been considered, for instance, in [36].

3.1.5 Preservation of stability and passivity

While describing RLC networks by (3.1)–(3.2), it often happens that the input and output matrices are equal, i.e., $B = L$. The following theorem shows that under certain conditions on C and G matrices, the system is passive.

Theorem 4. *Let $G, C \in \mathbb{R}^{N\times N}$, and $B, L \in \mathbb{R}^{N\times m}$. Assuming that G is symmetric positive semidefinite, i.e., $G \geq 0$ and C is symmetric positive definite, i.e., $C = C^T > 0$, and additionally $B = L$. Then, the system is stable and passive.*

The above theorem is also important from the point of view of model order reduction. In particular, it allows us to prove that Galerkin projections V and W ($V = W$) preserve stability and passivity of the reduced models. This is stated in the following theorem.

Theorem 5. *If the linear system*

$$C\frac{d}{dt}\boldsymbol{x}(t) = -G\boldsymbol{x}(t) + B\boldsymbol{u}(t), \tag{3.19}$$

$$\boldsymbol{y}(t) = L^T\boldsymbol{x}(t), \tag{3.20}$$

is such that $C = C^T > 0$, $G \geq 0$ and $B = L$, then the reduced model

$$V^*CV\frac{d}{dt}\hat{\boldsymbol{x}} = -V^*GV\hat{\boldsymbol{x}} + V^*B\boldsymbol{u}, \tag{3.21}$$

$$\boldsymbol{y} = L^TV\hat{\boldsymbol{x}}, \tag{3.22}$$

where V has full column rank, is stable and passive.

Proof. Suppose that the matrix G is positive semidefinite, i.e., $\mathbf{x}^T G\mathbf{x} \geq 0$ for all $\mathbf{x} \neq 0$. Then, the reduced matrix $\widetilde{G}$ satisfies

$$\widetilde{\mathbf{x}}^T \widetilde{G}\widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}^T \left(V^T G V\right) \widetilde{\mathbf{x}} = (\widetilde{\mathbf{x}} V^T) G (V\widetilde{\mathbf{x}}) = \mathbf{y}^T G \mathbf{y} \geq 0,$$

which is also positive semidefinite. By Theorem 4, the reduced model is stable and passive. □

We note that there are no constrains on the construction of V. The model order reduction method PRIMA [66] uses Galerkin projection (also known as congruence transform) and, therefore, preserves stability and passivity.

3.1.6 Synthesis of reduced order models

The synthesis procedure of the reduced order model (3.3)–(3.4) can be summarized as follows [48]:

- Compute the poles λ_i, which are eigenvalues of the generalized eigenvalue problem $-\widetilde{G}\mathbf{x} = \lambda_i \widetilde{C}\mathbf{x}$, and obtain the left and the right eigenvectors $\widetilde{\mathbf{y}}_i$ and $\widetilde{\mathbf{x}}_i$ by the QZ method. Note that the full space method (QZ) can be used since the system dimension is $n \ll N$.

- Use Foster realization to obtain either the admittance, or the impedance transfer function (2.46) as discussed in Section 2.4.6.

- Create a *netlist* of the circuit. Netlist is a file which contains the values of the circuit elements and their topological location in the network. The netlist can be put into a circuit simulator, e.g., PSTAR [2], to perform, for instance, frequency or time domain analysis.

In practice, the netlist of the reduced system is desired to be small, i.e., it should describe a dynamical system with few unknowns and be sufficiently sparse. In overall, it should contain fewer circuit elements than the original circuit. This would ensure that the reduced circuit is indeed cheaper to simulate that the original one. Of course, the dimension, n, of the reduced system cannot be smaller than the number of ports in the system. In other words, the ports have to preserved in the reduced system.

3.2 Model Order Reduction for multi-terminal networks

In general, networks arising during parasitic extraction contain large resistor (R) subnetworks and subnetworks with a mixture of components, e.g., subnetworks with resistors and inductors (RL), or subnetworks with resistors, inductors, and capacitors (RLC). Moreover, extracted networks can have a large amount of ports and nonlinear elements like diodes and transistors. Simulation of such complex networks may be very time consuming or unfeasible; therefore, model order reduction is required. Before applying MOR techniques, one may consider to decompose the networks into subnetworks and reduce these subnetworks separately. After that, the reduced subnetworks are connected and the nonlinear elements are added back. Such "divide and conquer" approach may lead to better result than applying reduction on the whole network at once.

Since networks may contain many ports, classical projection based model order reduction techniques discussed above have the disadvantage that they lead to dense reduced-order models. As a result, synthesized reduced networks may contain even more elements than the original ones. Therefore, special reduction techniques which preserve sparsity are required.

Recently, much effort has been focused on preserving sparsity of many-terminal R, RC and RLC networks [74], [49], [96], [50]. In the following sections we will discuss existing methods for reduction of large R networks and in Chapter 5 we will show a couple of strategies for improving reduction and sparsity of such networks.

In the next section, we will consider the reduction method for multi-terminal networks currently used in the EM tool Fasterix [64], [20], [1], [63]. We note that an original network in Fasterix is not given as a netlist and it is rather described by the relation between currents and voltages (2.1). As a result, the reduction method described below, cannot be directly applied for the reduction of an arbitrary RLC network. Nevertheless, for the purpose of completeness, we will briefly review this method and demonstrate a problem related to it, while extensive investigation of this method will be given in Chapter 4.

3.2.1 Model Order Reduction in Fasterix

As we have seen in Chapter 2, Fasterix translates electromagnetic properties of the interconnect system up to a certain maximum frequency into a circuit which is described

by the system of Kirchhoff equations (3.23)–(3.24).

$$(R + sL)\mathbf{I} - P\mathbf{V} = 0, \tag{3.23}$$

$$P^T\mathbf{I} + sC\mathbf{V} = \mathbf{J}, \tag{3.24}$$

where $s = -j\omega$, the resistance matrix $R \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, the inductance matrix $L \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, the incidence matrix $P \in \{1, 0, -1\}^{\mathcal{F}\times\mathcal{N}}$, and the capacitance matrix $C \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ are computed by Fasterix. $\mathbf{I} \in C^{\mathcal{F}}$ is the vector of currents flowing in the branches, $\mathbf{V} \in C^{\mathcal{N}}$ is the vector of voltages at the nodes, and $\mathbf{J} \in C^{\mathcal{N}}$ is the vector of currents flowing into the interconnection system. Note, that $\mathbf{J}$ flows only through the ports, i.e., upon permutation, $\mathbf{J}^T = \begin{pmatrix} \mathbf{J}_p^T & \mathbf{J}_i^T \end{pmatrix}$ and $\mathbf{J}_i^T = 0$. The matrices R, L are symmetric positive definite and do *not* correspond to MNA equations considered in Section 2.4 since R is not diagonal.

To obtain the admittance matrix which describes the input-output behaviour of interconnect, we subdivide all nodes into two subsets: ports (denoted as p) and internal nodes (i). Splitting similarly $\mathbf{V}$, P and C into corresponding blocks, we rewrite (3.23)–(3.24) as follows

$$\begin{pmatrix} R + sL & -P_i & -P_p \\ P_i^T & sC_{ii} & sC_{ip} \\ P_p^T & sC_{pi} & sC_{pp} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \mathbf{V}_i \\ \mathbf{V}_p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{J}_p \end{pmatrix}.$$

Eliminating $\mathbf{I}$ and $\mathbf{V}_i$, one obtains the admittance matrix $Y_p(s) : \mathbb{C} \to \mathbb{C}^{p\times p}$ of the *original circuit* which describes the behaviour of interconnect:

$$\mathbf{J}_p = \underbrace{\left(B_o^T(s)(G + sC)^{-1}B_i(s) + sC_{pp}\right)}_{Y_p(s)}\mathbf{V}_p, \tag{3.25}$$

where

$$B_o^T(s) = \begin{pmatrix} P_p^T & sC_{ip}^T \end{pmatrix}, \quad G = \begin{pmatrix} R & -P_i \\ P_i^T & 0 \end{pmatrix},$$

$$C = \begin{pmatrix} L & 0 \\ 0 & C_{ii} \end{pmatrix}, \quad B_i^T(s) = \begin{pmatrix} P_p^T & -sC_{ip}^T \end{pmatrix}.$$

The goal of the model order reduction technique in Fasterix (so-called *super node algorithm* (SNA)) is to generate a *reduced* RLC circuit, which has approximately the same behaviour at the ports as the original circuit described by $Y_p(s)$. Below we show the main steps of the SNA [20] [93].

1. Fasterix chooses a subset of internal nodes which will be retained in the final circuit. These nodes are called *super nodes*. Fasterix defines the super nodes in such

a way that between every point on the conductor and at least one of the super nodes, there exists a conducting path, no greater in length than a predefined small fraction of the minimum wavelength λ_{min}, such that

$$\lambda_{min} = \frac{1}{f_{max}\sqrt{\mu_0 \epsilon_{max}}}, \tag{3.26}$$

where f_{max} is the highest simulation frequency, ϵ_{max} is the highest permittivity and μ_0 is the permeability of free space. This step is done by Fasterix automatically.

Adding these super nodes in the subset of ports whic have to be preserved in the model, a conductance matrix of the resulting RLC circuit can be defined similar to $Y_p(s)$ in (3.27):

$$\widetilde{\mathbf{J}}_N = \underbrace{\left(\widetilde{B}_o^T(s)(\widetilde{G}+s\widetilde{C})^{-1}\widetilde{B}_i(s)+s\widetilde{C}_{NN}\right)}_{Y_N(s)}\widetilde{\mathbf{V}}_N, \tag{3.27}$$

where $Y_N : \mathbb{C} \to \mathbb{C}^{\mathcal{N}_1 \times \mathcal{N}_1}$, and $\mathcal{N}_1$ is the number of ports plus the number of super nodes. At this point, the super nodes play a role of ports in the model.

2. Depending on the frequency range for which the interconnect structure has to function, three different approximations of Y_N can be distinguished. For example, to carry out simulations in time domain, the full frequency range approximation for Y_N is required:

$$\widetilde{Y}_N(s) = P_N^T \Psi \left(\Psi^T (R+sL)\Psi\right)^{-1} \Psi^T P_N + sY_C,$$

where $\Psi \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}-\mathcal{N}_2}$ is such that its columns span the null space of P_N^T, and $Y_C \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_1}$ is necessary for capacitance contributions in the RLC circuit. Both Ψ and Y_C are computed by Fasterix.

3. Since direct realization of $\widetilde{Y}_N$ will lead to a large RLC circuit, Fasterix uses frequency fitting to approximate $\widetilde{Y}_N$ at $m+1$ frequency points, where $2 \leq m \leq 8$ (usually, $m = 4$). Thus, the final admittance matrix of RLC circuit is

$$\hat{Y}_N = \sum_{l=1}^{m} \frac{H_l}{s-\lambda_l} + s\hat{Y}_C,$$

where $\hat{Y}_C \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_1}$, $H_l \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_1}$, and λ_l are capacitance matrix, residue matrix, and poles computed by Fasterix.

4. Based on $\hat{Y}_N$ Fasterix constructs a netlist of RLC circuit which has approximately the same behaviour at the ports as interconnect system. The number of elements in the circuit is $O(m\mathcal{N}_1^2)$.

Thus the constructed RLC circuit (which we also called *reduced circuit*) has usually much

less elements than the original circuit defined by (3.23)–(3.24). Though there is no direct representation of the original circuit, we note that the inductance matrix, L, is usually dense, which means that the original circuit has many mutual inductances.

Though the SNA delivers compact extracted models, the problem is that the SNA sometimes delivers non-passive RLC circuits. Figure 3.1 demonstrates a time response of the original and reduced by the SNA circuit carried out in PSTAR [2]. It is clearly observed that the reduced circuit behaves unstable. This fact inspired us to investigate each step of the super node algorithm in detail and analyze the issue of passivity preservation. In Chapter 4 we will prove that the SNA does not guarantee passivity, and we will propose two modifications of the SNA that guarantee passive reduced circuits. Both modifications will be analyzed from the perspective of applicability for large original circuits.

We note that the system (3.23)–(3.24) can be rewritten in the form of the state space system (3.1)–(3.2) and, therefore, it can be further reduced by classical MOR techniques. This option has been studied in [41], [42], while comparison of the SNA with Krylov subspace method will be considered in Chapter 4 (Section 4.5.1).

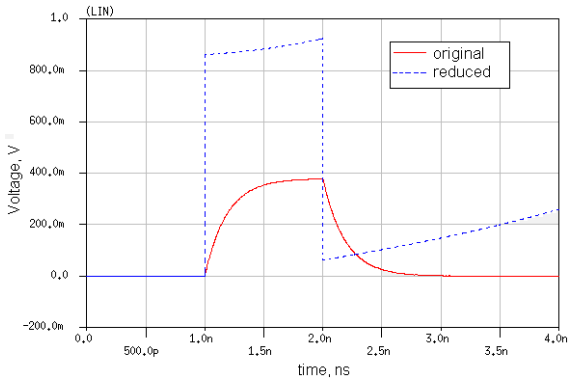

Figure 3.1: *Time response of the original circuit and the reduced one by the super node algorithm*

3.2.2 Model Order Reduction for resistor networks

If a network contains only resistors, equation (3.1) becomes $G\mathbf{x} = B\mathbf{u}$. For convenience we rewrite it as

$$G\mathbf{v} = B\mathbf{i}, \tag{3.28}$$

where $G \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite conductance matrix, $\mathbf{v} \in R^n$ is the vector of node voltages, and $\mathbf{i} \in R^n$ is the vector of currents injected into the ports (external nodes). Subdividing a set of nodes into external (ports) and internal, one can

rewrite (3.28) in block form:

$$\begin{pmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{pmatrix} \begin{pmatrix} \mathbf{v}_e \\ \mathbf{v}_i \end{pmatrix} = \begin{pmatrix} \hat{B} \\ 0 \end{pmatrix} \mathbf{i}_e, \tag{3.29}$$

where $\mathbf{v}_e \in \mathbb{R}^{n_e}$ and $\mathbf{v}_i \in \mathbb{R}^{n_i}$ are the vectors of voltages at external and internal nodes, respectively $(n = n_e + n_i)$, $\mathbf{i}_e \in R_e^n$ is the vector of currents injected into the external nodes, $\hat{B} \in \{-1, 0, 1\}^{n_e \times n_e}$ is the incidence matrix for the current injections, $G_{11} = G_{11}^T \in \mathbb{R}^{n_e \times n_e}$, $G_{12} \in \mathbb{R}^{n_e \times n_i}$, and $G_{22} = G_{22}^T \in \mathbb{R}^{n_i \times n_i}$.

A k-th current source between terminals a and b with current j leads to contributions $B_{a,k} = 1$, $B_{b,k} = -1$, and $\mathbf{i}_e(k) = j$. If current is only injected into a terminal a, then $B_{a,k} = 1$ and $\mathbf{i}_e(k) = j$.

The problem of reduction of a large resistor network is defined as follows [74]. Given a very large resistor network described by (3.28), find an equivalent network that:

(a) has the same terminals,

(b) has exactly the same path resistances between terminals,

(c) has $\hat{n}_i \ll n_i$ internal nodes,

(d) has $\hat{N} \ll N$ resistors,

(e) is realizable as a netlist.

If one straightforwardly eliminates all internal nodes, then one obtains an equivalent network described by $\widetilde{G}\mathbf{v}_e = B\mathbf{i}_e$ with conductance matrix

$$\widetilde{G} = G_{11} - G_{12}G_{22}^{-1}G_{12}^T. \tag{3.30}$$

This reduced circuit satisfies conditions (a)-(c), but in general, it will violate (d) and (e). Indeed, if the number of terminals, n_e is large, then the number of resistors $\hat{N} = (n_e^2 - n_e)/2$ in the dense reduced network is in general much larger than the number of resistors, N, in the sparse original network, leading to increased memory and CPU requirements.

Approaches for reduction of resistor networks have been considered, for instance, in [62], with the use of large-scale graph partitioning packages such as METIS [54], but these approaches have only been applied to small networks. In the next two sections we will discuss existing state-of-the art methods for multi-terminal R and RLC networks.

3.2.3 ReduceR - efficient reduction of resistor networks

In [74] an approach, *ReduceR*, for the exact reduction of large resistor networks has been suggested. It is based on two concepts. The first concept is used to recover the structure of a resistor network as an undirected graph. The second concept is the elimination of internal nodes, known as the Schur complement, together with a reordering algorithm, Approximate Minimum Degree (AMD) [7]. AMD is usually used as a preprocessing step for the Cholesky factorization, to determine the order in which nodes have to be eliminated, to reduce "fill-in". The fill-in reducing property of this method guarantees the sparsity of the reduced network.

Algorithm 1 shows the main steps of *ReduceR*. In the first phase (steps 1-14), a given network is partitioned in smaller individual parts, known as strongly connected components (scc), which can be reduced individually. Computing the scc of an undirected graph is a known graph problem, and it requires $O(|V| + |E|)$ operations, where $|V|$ is the number of vertices, and $|E|$ is the number of edges [74]. Implementation of an algorithm to compute scc can be found in [31].

In the second phase (steps 2-14) two-connected components are obtained. A connected component is called two-connected if it becomes disconnected by removing one vertex. A vertex with such property is called an articulation node. Preserving articulation nodes in the network and reducing two-connected components independently delivers the next reduction. Computing two-connected components requires $O(|V| + |E|)$ time, and efficient implementation of this graph algorithm can be found in [31].

In the third phase (steps 15-17) the partially reduced network is reduced even further by a global approach, which is based on AMD. In this phase it is determined which internal nodes have to be eliminated. For resistor networks elimination of a set of nodes is equivalent to calculate the Schur complement [32] of the principal submatrix of the conductance matrix. By doing this, no approximation error is made (up to round-off errors). The idea is to eliminate those internal nodes which cause the least fill-in and, for this goal, AMD is used. To find the optimal reduction, the internal nodes are eliminated one by one in the order computed by AMD, while keeping track of the reduced system with fewest resistors. There is also option to use constrained AMD [22] to reorder unknowns such that the terminals will be among the last to eliminate.

It should be noted that *ReduceR* is aimed to deliver resulting reduced matrix as sparse as possible since the ordering of eliminating nodes is chosen to minimize fill-in. The path resistances between external nodes remain equal to the path resistances in the original network up to roundoff errors.

Algorithm 1 ReduceR

INPUT: conductance matrix G, list of terminals
OUTPUT: reduced conductance matrix G
1: Compute strongly connected components G_i of G by algorithm in [31]
2: **for** every strongly connected component G_i in G with one or more terminals **do**
3: Compute the two-connected components $G_i^{(2)}$ of G_i
4: **for** every two-connected component $G_i^{(2)}$ of G_i **do**
5: **if** $G_i^{(2)}$ has no terminals and exactly one articulation node a **then**
6: Remove all resistors and nodes in $G_i^{(2)}$ from G_i, but keep articulation node a
7: **else if** $G_i^{(2)}$ has no terminals and exactly two articulation nodes a and b **then**
8: Replace the network between a and b by a single equivalent resistor
9: **else if** $G_i^{(2)}$ has exactly one terminal t and exactly one articulation node a **then**
10: Replace the network between a and t by a single equivalent resistor
11: **else**
12: Keep $G_i^{(2)}$ (will be reduced in step 15-16)
13: **end if**
14: **end for**
15: Reorder rows and columns of G_i using AMD
16: Eliminate internal nodes one by one, keeping track of reduced system with the fewest resistors
17: Replace G_i by its reduced equivalent in G
18: **end for**

3.2.4 Sparse Implicit Projection (SIP)

In [96], the Sparse Implicit Projection (SIP) algorithm has been developed for the reduction of many-terminal RLC networks. Mathematically, SIP is a projection method. It inherits accuracy through moment matching at multiple expansion points and preserves passivity. Similar to *ReduceR*, SIP implementation is based on sparse matrix manipulations together with an elimination approach. This makes it more efficient for large models than other projection based methods. For more details about SIP, the reader is referred to [96].

In this subsection we will show that for pure resistor networks, the node elimination procedure of *ReduceR* is equivalent to the node elimination procedure of SIP.

In the implementation of SIP, a standard Cholesky decomposition is performed recursively [32]. At the beginning we have $G^{(1)} = G$ and at each step a new variable is eliminated. At step i the matrix G has the form:

$$G^{(i)} = \begin{pmatrix} I_{i-1} & & \\ & a_{i,i} & \mathbf{b}_i \\ & \mathbf{b}_i^T & B^{(i)} \end{pmatrix},$$

where I_{i-1} is identity matrix of dimension $i-1$. On the other hand, $G^{(i)}$ can be rewritten as a product of low triangular matrix L^i and $G^{(i+1)}$ as follows

$$G^{(i)} = L^{(i)} G^{(i+1)} (L^{(i)})^T,$$

where

$$G^{(i+1)} = \begin{pmatrix} I_{i-1} & & \\ & 1 & 0 \\ & 0 & B^{(i)} - \frac{1}{a_{i,i}} \mathbf{b}_i \mathbf{b}_i^T \end{pmatrix}.$$

After n steps we have $G^{(n+1)} = I$. It can be seen that each matrix $B^{(i)} - \frac{1}{a_{i,i}} \mathbf{b}_i \mathbf{b}_i^T$ is, in fact, the Schur complement which is used in *ReduceR* at step 16. SIP computes $B^{(i)} - \frac{1}{a_{i,i}} \mathbf{b}_i \mathbf{b}_i^T$ for $i = 1, \ldots n_i$, where n_i is the number of internal nodes. Thus, elimination procedure of internal nodes in SIP is equivalent to elimination procedure of internal nodes in *ReduceR*. Note that AMD is also incorporated in the algorithm to minimize fill-in of the conductance matrix.

On the other hand, SIP is a projection technique. A projection is not constructed explicitly. To show it, one can rewrite the Schur complement (3.30) as a congruence transformation [96]

$$S = M^T G M,$$

where the projection matrix, M, is a product of many individual projections:

$$M = M^{(1)} M^{(2)} \dots M^{(n_i)},$$

which are obtained during elimination of nodes during the Cholesky factorization

$$G^{(i+1)} = (M^{(i)})^T G^{(i)} M^{(i)},$$

where $M^{(i)} = (L^{(i)})^{-T}$, $i = 1, \dots n_i$.

In case of reduction of RLC networks, it can be shown that the transfer function of the reduced order system, $\widetilde{H}(s) = \widetilde{B}(\widetilde{G} + s\widetilde{C})^{-1}\widetilde{B}^T$, with $\widetilde{G} = M^T G M$, $\widetilde{C} = M^T C M$, $\widetilde{B} = M^T E$ matches two moments of the original transfer function $H(s) = B(G + sC)^{-1}B^T$. To increase accuracy, multi-point reduction scheme [96] can be used; however, it is out of the scope of this section since we are concerned with reduction of resistor networks.

3.3 Concluding remarks

In this chapter we discussed the goals and some popular existing methods of model order reduction. We mentioned classical projection based methods, such as modal approximation, balanced truncation, moment matching. Then we concentrated on the reduction method used in the tool Fasterix and methods for reduction of large resistor networks (*ReduceR* and SIP) which are important ingredients for the methods developed in this thesis. We have shown the equivalence of *ReduceR* and SIP in the particular case of resistor networks and mentioned the connection with a projection based framework. It was shown that preservation of sparsity is a crucial condition for reduction of many-terminal networks.

Chapter 4

Model Order Reduction in Fasterix

The super node algorithm (SNA) is a model order reduction technique used in the electromagnetic tool Fasterix. The super node algorithm performs model order reduction based on physical principles. Although the algorithm provides us with compact models, its passivity has not thoroughly been studied yet. The loss of passivity is a serious problem because simulations of the reduced network may encounter artificial behavior which render the simulations useless. In this chapter we first show a derivation of Kirchhoff equations, which describe the interconnect system, and we prove an important property of the resistance matrix. Second, we review the super node algorithm and discuss the application of projection based reduction methods. Moreover, we find the reason of delivering non-passive models and propose two modified versions of the algorithm which guarantee passivity. In one of them, a passivity enforcement procedure is applied after the frequency fitting step within the super node algorithm. This allows to preserve passivity while keeping the main advantages of the algorithm. Finally numerical examples validate the proposed approach.

4.1 Introduction

Computer techniques have revolutionized the way in which electromagnetic problems are analyzed. In this chapter we concentrate on Fasterix [64], [20], [1], [63], an EM tool for analyzing electromagnetic (EM) behavior of interconnect systems. Fasterix is used at Philips and NXP Semiconductors and has proved very successful for interconnect

systems as printed circuit boards, the main application for which it was developed.

As a first step in Fasterix a geometry preprocessor subdivides a given conductor of arbitrary geometry into quadrilateral elements and derives directly a lumped model. In this chapter we refer to it as *original circuit*. Such circuit when observed from its ports, is equivalent to the interconnect system. The values of the mutual inductances, capacitors and resistors of this circuit are extracted using the boundary element method and may be of the order of many tens of millions. Thus the direct use of this circuit is inefficient because computer memory and CPU limitations imply that the interconnect system cannot realistically be simulated. As a model reduction step, Fasterix employs a *compressed equivalent circuit* (*reduced circuit*) which has a much smaller number of nodes and reducing the circuit analysis computations by many orders of magnitude. Nodes, at which the reduced circuit is built, are called *super nodes*. The *super node algorithm* (SNA) is a procedure in Fasterix to obtain a reduced circuit from the the original one.

The advantage of the SNA is that it is inspired by the physical insight into the models, and produces reduced circuits depending on the maximum predefined frequency. In Fasterix the original and reduced circuits are generated once, i.e., they do not have to be recomputed for each time or frequency. Due to it, Fasterix has proved to be orders of magnitude more efficient, than typical tools based on finite element method and boundary element method [20].

Although the algorithm provides us with compact models, some of them suffer from instabilities which can be observed during time domain simulations [63], [42].

In this chapter we first derive Kirchhoff equations from Maxwell's equations which are a starting point for applying the SNA. Second, we analyze the SNA and explain why it delivers non-passive circuits. We also discuss the use of the projection MOR techniques for reduction and the necessity to modify the SNA to preserve passivity of the reduced models. Furthermore we propose a few modified versions of the SNA, which guarantee passivity. Both versions will be analyzed on applicability to reduce large models and will be validated by some numerical examples.

4.2 Derivation of Kirchhoff equations

In this section we will derive the Kirchhoff equations which describe the behavior of the original circuit and which are a staring point for applying any reduction technique. For this purpose we derive Electric Field Integral Equations (EFIE) from Maxwell's equations and then formulate an equivalent boundary value problem. Next, we present a variational formulation of this problem and the function spaces that contain its weak solutions. Based on it, we will derive a linear set of equations, which correspond to the

Kirchhoff equations, the solutions of which represent the currents, charges and potentials in an electrical circuit. The main references for this section are [93] and [20].

4.2.1 Electric field integral equation

Maxwell's equations for harmonic fields are given by

$$\nabla \times \mathbf{E} = i\omega\mathbf{B} \quad \text{(Faraday's law)}, \tag{4.1}$$
$$\nabla \times \mathbf{H} = \mathbf{J} - i\omega\mathbf{D} \quad \text{(Ampere's law)}, \tag{4.2}$$
$$\nabla \cdot \mathbf{B} = 0 \quad \text{(Gauss's law)}, \tag{4.3}$$
$$\nabla \cdot \mathbf{D} = \rho, \tag{4.4}$$

where $\mathbf{E}$ is the electric field, $\mathbf{H}$ is the magnetic field, $\mathbf{B}$ is the magnetic induction, $\mathbf{D}$ is the electric displacement, $\mathbf{J}$ current density, ρ is charge density. Associated with Maxwell's equations, we have a continuity equation:

$$\nabla \cdot \mathbf{J} - i\omega\rho = 0, \tag{4.5}$$

which can be obtained from (4.2) and (4.4). We will consider linear, homogenous, and isotropic media, such that

$$\mathbf{D} = \epsilon\mathbf{E}, \tag{4.6}$$
$$\mathbf{B} = \mu\mathbf{H}, \tag{4.7}$$
$$\mathbf{J} = \sigma\mathbf{E}, \tag{4.8}$$

where the scalars ϵ (permittivity), μ (permeability) and σ (conductivity) are assumed to be constant. Equations (4.1)–(4.5) give a relation between electric and magnetic fields. We will obtain a smaller number of second-order equations, called Helmholtz equations which are equivalent to Maxwell's equations.

Since $\nabla \cdot \mathbf{B} = 0$, $\mathbf{B}$ can be defined in terms of a vector potential $\mathbf{A}$:

$$\mathbf{B} = \nabla \times \mathbf{A}. \tag{4.9}$$

Then equation (4.1) can be rewritten as

$$\nabla \times (\mathbf{E} - i\omega\mathbf{A}) = 0.$$

The argument of rotation ($\nabla\times$) can be presented as a gradient of the electric potential ϕ:

$$(\mathbf{E} - i\omega\mathbf{A}) = -\nabla\phi,$$

or

$$\mathbf{E} = i\omega\mathbf{A} - \nabla\phi. \tag{4.10}$$

Substituting (4.10) in (4.6) one obtains

$$\mathbf{D} = \epsilon\mathbf{E} = \epsilon(i\omega\mathbf{A} - \nabla\phi). \tag{4.11}$$

Substituting (4.11) in (4.4) one obtains

$$i\omega\epsilon\nabla\cdot\mathbf{A} - \nabla\cdot(\epsilon\nabla\phi) = \rho. \tag{4.12}$$

Since

$$\nabla\times\mathbf{B} = \mu(\nabla\times\mathbf{H}) = \mu(\mathbf{J} - i\omega\mathbf{D}), \tag{4.13}$$

$$\nabla\times\mathbf{B} = \nabla\times(\nabla\times\mathbf{A}), \tag{4.14}$$

combining the right parts of (4.13) and (4.14) and taking into account (4.11) together with the property $\nabla\times(\nabla\times\mathbf{A}) = \nabla(\nabla\cdot\mathbf{A}) - \triangle\,\mathbf{A}$, one obtains

$$\nabla(\nabla\cdot\mathbf{A}) - \triangle\,\mathbf{A} - i\omega\mu\epsilon\nabla\phi - \omega^2\mu\epsilon\mathbf{A} = \mu\mathbf{J}. \tag{4.15}$$

Obtained equations (4.12) and (4.15) are still coupled equations. To uncouple them, we will use definitions of $\mathbf{A}$ and ϕ. Since $\mathbf{B}$ is defined in terms of $\mathbf{A}$ in (4.9), the vector potential is arbitrary to the extent that the gradient of some scalar function Λ can be added, i.e.,

$$\mathbf{A}' = \mathbf{A} + \nabla\Lambda. \tag{4.16}$$

and similarly

$$\phi' = \phi + i\omega\Lambda. \tag{4.17}$$

To express $\mathbf{A}$ and ϕ uniquely, an extra condition must be added, for instance, the Lorentz gauge condition [51]:

$$\nabla\cdot\mathbf{A} - i\omega\mu\epsilon\phi = 0. \tag{4.18}$$

Substituting (4.18) into (4.15) and (4.12) one obtains the inhomogeneous Helmholtz equations which uncouple equations (4.15) and (4.12) for $\mathbf{A}$ and ϕ:

$$(\triangle + k^2)\mathbf{A} = -\mu\mathbf{J}, \tag{4.19}$$

$$\nabla\cdot(\epsilon\nabla\phi) + \epsilon k^2\phi = -\rho, \tag{4.20}$$

where $k = \omega\sqrt{\mu\epsilon}$.

The solutions of the Helmholtz equations (4.19)–(4.20) are [93]

$$\mathbf{A}(\mathbf{x},\omega) = \int_\Omega \mathbf{G}_A(\mathbf{x}',\mathbf{x};k)\ \mu\ \mathbf{J}(\mathbf{x}',\omega)\ d\mathbf{x}', \tag{4.21}$$

$$\phi(\mathbf{x},\omega) = \int_\Omega \mathbf{G}_\phi(\mathbf{x}',\mathbf{x};k)\frac{\rho(\mathbf{x}',\omega)}{\epsilon}\ d\mathbf{x}', \tag{4.22}$$

where Ω denotes the domain of the conductor, and

$$\mathbf{G}_\phi = \frac{e^{ik|\mathbf{x}'-\mathbf{x}|}}{4\pi|\mathbf{x}'-\mathbf{x}|}, \quad \mathbf{G}_\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \frac{e^{ik|\mathbf{x}'-\mathbf{x}|}}{4\pi|\mathbf{x}'-\mathbf{x}|} \tag{4.23}$$

are the fundamental solutions of (4.19)–(4.20) for a homogeneous medium. Substituting (4.21) into Ohm's law

$$\mathbf{J} = \sigma\mathbf{E} = \sigma\left(i\omega\mathbf{A} - \nabla\phi\right), \tag{4.24}$$

and taking into account (4.5) and (4.22), we obtain the following system of equations

$$\frac{\mathbf{J}}{\sigma} + \nabla\phi - i\omega\int_\Omega \mathbf{G}_\mathbf{A}\ \mu\ \mathbf{J}\ d\mathbf{x}' = 0, \tag{4.25}$$

$$\nabla\cdot\mathbf{J} - i\omega\rho = 0, \tag{4.26}$$

$$\phi - \int_\Omega \mathbf{G}_\phi\frac{\rho}{\epsilon}\ d\mathbf{x}' = 0, \tag{4.27}$$

which are collectively referred as the mixed potential EFIE.

4.2.2 The boundary value problem

Based on the mixed potential EFIE (4.25)–(4.27), the following *boundary value problem* can be formulated

$$\frac{\mathbf{J}}{\sigma} + \nabla\phi - i\omega\int_\Omega \mathbf{G}_\mathbf{A}\ \mu\ \mathbf{J}\ d\mathbf{x}' = 0, \tag{4.28}$$

$$\nabla\cdot\mathbf{J} - i\omega\rho = 0, \tag{4.29}$$

$$\phi - \int_\Omega \mathbf{G}_\phi\frac{\rho}{\epsilon}\ d\mathbf{x}' = 0, \tag{4.30}$$

The boundary conditions are

$$\phi(x) = \mathbf{V}_{\text{fixed}}, \quad \mathbf{x}\in\Gamma_V, \tag{4.31}$$

$$\mathbf{J}\cdot\mathbf{n} = 0, \quad \mathbf{x}\in\Gamma, \tag{4.32}$$

where $\mathbf{n}$ is the unit normal vector to Γ, and ϕ, $\mathbf{J}$, ρ are functions belonging to the following function spaces

$$\rho \in L^2(\Omega) = \left\{ \mathbf{u} \in L^2(\Omega) | \int_\Omega \mathbf{u}^2 d\mathbf{x} < \infty \right\},$$

$$\phi \in H^1(\Omega) = \left\{ \mathbf{u} \in L^2(\Omega) | \nabla \mathbf{u} \in L^2(\Omega)^3 \right\},$$

$$\mathbf{J} \in H^{div}(\Omega) = \left\{ \mathbf{v} \in L^2(\Omega)^3 | \nabla \cdot \mathbf{v} \in L^2(\Omega) \right\}.$$

Ω denotes finite conductor regions, Γ denotes the boundary of these regions, Γ_V is a part of Γ which is restricted to the ports, $\mathbf{n}$ is the unit normal vector perpendicular to the boundary surface. The physical meaning of the boundary conditions is that no current flows through the boundary Γ, except the boundary Γ_V of the ports, where potential is applied.

4.2.3 Variational formulation and discretization

Multiplying (4.28)–(4.30) by the test functions

$$\widetilde{\rho} \in L^2(\Omega),$$

$$\widetilde{\phi} \in H^1(\Omega),$$

$$\widetilde{\mathbf{J}} \in H_0^{div}(\Omega) = \left\{ \mathbf{v} = L^2(\Omega)^3 | \nabla \cdot \mathbf{v} \in L^2(\Omega); \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \Gamma \right\},$$

and integrating over the domain Ω of the conductors one obtains a *variational formulation* of the boundary value problem

$$\int_\Omega \left(\frac{\mathbf{J}}{\sigma} + \nabla\phi - i\omega \int_\Omega \mathbf{G_A}\ \mu\ \mathbf{J}\ d\mathbf{x}' \right) \widetilde{\mathbf{J}}\ d\mathbf{x} = 0, \tag{4.33}$$

$$\int_\Omega (\nabla \cdot \mathbf{J} - i\omega\rho)\ \widetilde{\phi}\ d\mathbf{x} = 0, \tag{4.34}$$

$$\int_\Omega \left(\phi - \int_\Omega \mathbf{G}_\phi \frac{\rho}{\epsilon}\ d\mathbf{x}' \right) \widetilde{\rho}\ d\mathbf{x} = 0. \tag{4.35}$$

If functions J, ϕ and ρ satisfy the variational formulation (4.33)–(4.35) for any $\widetilde{\rho} \in L^2(\Omega)$ $\widetilde{\phi} \in H^1(\Omega)$ and $\widetilde{\mathbf{J}} \in H_0^{div}(\Omega)$, then J, ϕ and ρ satisfy (4.28)–(4.30). Integrating by parts the second term in (4.33) and substituting the boundary condition $\widetilde{\mathbf{J}} \cdot \mathbf{n} = 0$, one obtains a *weak formulation* of the boundary value problem:

$$\int_\Omega \left(\frac{\mathbf{J}}{\sigma} \cdot \widetilde{\mathbf{J}} + \phi \nabla \cdot \widetilde{\mathbf{J}} - i\omega \int_\Omega \mathbf{G_A}\ \mu\ \mathbf{J}\ d\mathbf{x}'\ \widetilde{\mathbf{J}} \right) d\mathbf{x} = 0, \quad \text{for all} \quad \widetilde{\mathbf{J}} \in H_0^{div}, \tag{4.36}$$

$$\int_\Omega (\nabla \cdot \mathbf{J} - i\omega\rho)\ \widetilde{\phi}\ d\mathbf{x} = 0, \quad \text{for all} \quad \widetilde{\phi} \in L^2(\Omega), \tag{4.37}$$

$$\int_\Omega \left(\phi - \int_\Omega \mathbf{G}_\phi \frac{\rho}{\epsilon} d\mathbf{x}' \right) \widetilde{\rho}\ d\mathbf{x} = 0 \quad \text{for all} \quad \widetilde{\rho} \in L^2(\Omega), \tag{4.38}$$

where

$$\rho, \phi, \in L^2(\Omega),$$
$$\mathbf{J} \in H_0^{div}(\Omega).$$

Integrals in (4.28)–(4.30) are 3D integrals over the volume Ω of the conductors. They may be replaced by the 2D integrals over the surface due to the assumptions that the conductors are planar and very thin; therefore, the quantities $\mathbf{J}$, ϕ and ρ are constant in the direction perpendicular to the conductors.

To find approximate solutions of (4.36)–(4.38), the function spaces are approximated by finite dimensional subspaces. Let Ω_h be the domain of the thin conductors. Ω_h can be subdivided into quadrilateral elements Ω_j, $j = 1, \ldots, \mathcal{N}$. Let $\mathcal{F}$ denote the number of edges of the quadrilateral elements, excluding edges in the boundary, and let $\mathcal{N}$ denote the number of quadrilateral elements. Thus the *discrete formulation* of the problem (4.36)–(4.38) is to find the functions $(\phi_h, \mathbf{J}_h, \rho_h)$ such that

$$\int_{\Omega_h} \left(\frac{\mathbf{J}}{\sigma} \cdot \widetilde{\mathbf{J}}_h - \phi_h \nabla \cdot \widetilde{\mathbf{J}}_h - i\omega \int_{\Omega_h} \mathbf{G_A}\ \mu\ \mathbf{J}_h\ d\mathbf{x}' \widetilde{\mathbf{J}}_h \right) d\mathbf{x} = 0, \quad \text{for all} \quad \widetilde{\mathbf{J}}_h \in H_{h,0}^{div}, \tag{4.39}$$

$$\int_{\Omega_h} (\nabla \cdot \mathbf{J}_h - i\omega\rho_h)\, \widetilde{\phi}_h\ d\mathbf{x} = 0, \quad \text{for all} \quad \widetilde{\phi}_h \in U_h, \tag{4.40}$$

$$\int_{\Omega_h} \left(\phi_h - \int_{\Omega_h} \mathbf{G}_\phi \frac{\rho_h}{\epsilon}\ d\mathbf{x}' \right) \widetilde{\rho}_h\ d\mathbf{x} = 0, \quad \text{for all} \quad \widetilde{\rho}_h \in W_h, \tag{4.41}$$

where U_h, W_h and $H_{h,0}^{div}$ are finite dimensional subspaces of the infinite dimensional function spaces L^2 and H_0^{div}. Thus functions ϕ_h, $\mathbf{J}_h$ and ρ_h can be expanded in terms of the basis functions. The scalar potential is approximated as

$$\phi_h(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}} V_j b_j(\mathbf{x}), \tag{4.42}$$

where V_j denotes the potential of the j-th element and $b_j(\mathbf{x})$ is

$$b_j(\mathbf{x}) = \begin{cases} 1 & \text{for } \mathbf{x} \in \Omega_j, \\ 0 & \text{elsewhere.} \end{cases}$$

The surface charge density is approximated as

$$\rho_h(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}} Q_j c_j(\mathbf{x}),$$

where Q_j denotes the charge of the j-th element and $c_j(\mathbf{x})$ is basis the function on the j-th element adopted to include the singularity of the charge density near the conductor boundary [93].

The surface current density is expanded as

$$\mathbf{J}_h(\mathbf{x}) = \sum_{l=1}^{\mathcal{F}} I_l \widetilde{\mathbf{w}}_l(\mathbf{x}),$$

where I_l is the current through edge l, and $\widetilde{\mathbf{w}}_l(\mathbf{x})$ is defined as

$$\widetilde{\mathbf{w}}_l(\mathbf{x}) = \begin{cases} f(\mathbf{x})\mathbf{w}_l(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega_i \cup \Omega_j \text{ and } \mathcal{F}_l = \Omega_i \cap \Omega_j \\ 0 & \text{otherwise} \end{cases},$$

where $f(\mathbf{x})$ is defined in [93] and it satisfies $\int_{\Omega_j} f(\mathbf{x})d\mathbf{x} = |\Omega_j|$, where $|\Omega_j|$ denotes the area of Ω_j. Edge functions $\mathbf{w}_l$ are defined later in (4.59)–(4.60). The important note about $\mathbf{w}_l$ is that they are constructed in such a way that the current through Ω_j is conserved. In other words, the current flowing inside Ω_j equals the current flowing outside Ω_j, i.e., $\sum_{k=1}^{4} I_k = 0$. This property will also hold for the domain Ω_h.

Substituting expansions of ϕ_h, $\mathbf{J}_h$ and ρ_h in the weak formulation (4.36)–(4.38), one obtains the following linear system of equations

$$\sum_{l=1}^{\mathcal{F}} \left(R_{kl} - i\omega L_{kl}\right) I_l - \sum_{j=1}^{\mathcal{N}} P_{kj} V_j = 0, \tag{4.43}$$

$$i\omega \sum_{j=1}^{\mathcal{N}} M_{ij} Q_j - \sum_{l=1}^{\mathcal{F}} P_{li} I_l = 0, \tag{4.44}$$

$$\sum_{j=1}^{\mathcal{N}} \left(M_{ij} V_j - D_{ij} Q_j\right) = 0, \tag{4.45}$$

where

$$R_{kl} = \int_{\Omega_h} \frac{1}{\sigma} \widetilde{\mathbf{w}}_l(\mathbf{x}) \cdot \widetilde{\mathbf{w}}_k(\mathbf{x}) d\mathbf{x}, \tag{4.46}$$

$$L_{kl} = \int_{\Omega_h} \widetilde{\mathbf{w}}_l(\mathbf{x}) \cdot \left\{ \int_{\Omega_h} \mathbf{G}_A(\mathbf{x}, \mathbf{x}') \ \mu \ \widetilde{\mathbf{w}}_k(\mathbf{x}') d\mathbf{x}' \right\} d\mathbf{x}, \tag{4.47}$$

$$P_{kj} = \int_{\Omega_j} b_j(\mathbf{x}) \nabla \cdot \widetilde{\mathbf{w}}_k(\mathbf{x}) d\mathbf{x}, \tag{4.48}$$

$$M_{ij} = \int_{\Omega_j} c_j(\mathbf{x}) b_i(\mathbf{x}) d\mathbf{x}, \tag{4.49}$$

$$D_{ij} = \int_{\Omega_j} c_j(\mathbf{x}) \left\{ \int_{\Omega_i} C_\phi(\mathbf{x}, \mathbf{x}') \frac{c_i(\mathbf{x}')}{\epsilon} d\mathbf{x}' \right\} d\mathbf{x}, \tag{4.50}$$

$k, l = 1, \ldots, \mathcal{F}$ and $i, j = 1, \ldots, \mathcal{N}$. Equations (4.43)–(4.45) are the set of $\mathcal{F} + 2\mathcal{N}$ equations, which can be rewritten as:

$$(R - i\omega L)\mathbf{I} - P\mathbf{V} = 0, \tag{4.51}$$

$$-P^T\mathbf{I} + i\omega M\mathbf{Q} = 0, \tag{4.52}$$

$$M^T\mathbf{V} - D\mathbf{Q} = 0. \tag{4.53}$$

For elements Ω_j which include the boundary of connection ports, i.e., $\Omega_j \subset \Gamma_V$, we have a boundary condition

$$\mathbf{V}_j = \mathbf{V}_{\text{fixed,j}}, \tag{4.54}$$

which means that the voltage is applied at the ports of the conductors. Thus the elements and edges after discretization of conductors can be associated with the nodes and branches of a graph. It means that the conductors can be interpreted as a circuit described by (4.51)–(4.54) which we will call original circuit. Eliminating $\mathbf{Q}$ we obtain a system of $\mathcal{F} + \mathcal{N}$ equations with the unknowns $\mathbf{V}$ and $\mathbf{I}$, which have the form of Kirchhoff's voltage and Kirchhoff's current laws:

$$(R - i\omega L)\mathbf{I} - P\mathbf{V} = 0, \tag{4.55}$$

$$-P^T\mathbf{I} + i\omega C\mathbf{V} = 0, \tag{4.56}$$

$$\mathbf{V}_j = \mathbf{V}_{\text{fixed,j}}, \tag{4.57}$$

where $R \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$ is the resistance matrix (sparse), $L \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$ is the inductance matrix (dense), $P \in \{1, 0, -1\}^{\mathcal{F}\times\mathcal{N}}$ is an incidence matrix, $C = MD^{-1}M^T \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the capacitance matrix, $\mathbf{I} \in \mathbb{C}^{\mathcal{F}}$ is the vector of currents flowing through the branches, $\mathbf{V} \in \mathbb{C}^{\mathcal{N}}$ is the vector of voltages at the nodes. Matrices R, L are symmetric positive definite and C is symmetric positive semidefinite. For a given conductor, the entries of R, L, C and P are computed by Fasterix.

To summarize, Kirchhoff equations (4.55)–(4.56) come from discretization of conductors and describe the original circuit. We note that presented derivation of Kirchhoff equations is not unique, other derivation is possible, for instance, in [64]. Since a direct use of the original circuit is inefficient, the model order reduction is required. Before going into details of a reduction technique used in Fasterix, we will show important properties of some matrices used in Kirchhoff equations.

4.2.4 Properties of R and P matrices

In the following lemma we will prove an important property of R matrix used in Kirchhoff equations (4.55)–(4.56).

Lemma 1. *Matrix R with elements R_{kl} in (4.46) is a positive definite matrix.*

For simplicity we consider the case of a single quadrilateral element Ω_j inside of the domain Ω_h with the edges of unit length. The proof can be extended for the case of a few quadrilateral elements of arbitrary shape.

Proof. Elements of R matrix are defined as follows

$$R_{kl} = \int_{\Omega_h} \frac{1}{\sigma} \widetilde{\mathbf{w}}_l(\mathbf{x}) \widetilde{\mathbf{w}}_k(\mathbf{x}) d\mathbf{x} = \int_{\Omega_h} \frac{1}{\sigma} f^2(\mathbf{x}) \mathbf{w}_l(\mathbf{x}) \mathbf{w}_k(\mathbf{x}) d\mathbf{x}, \quad k,l = 1,\ldots,\mathcal{F}, \tag{4.58}$$

where $f(\mathbf{x}) = 1$ and the edge functions are [94]

$$\mathbf{w}_1 = (1-s_2)\mathbf{v}_2 / |\mathbf{v}_1 \times \mathbf{v}_2|, \quad \mathbf{w}_2 = -s_1 \mathbf{v}_1 / |\mathbf{v}_1 \times \mathbf{v}_2|, \tag{4.59}$$

$$\mathbf{w}_3 = -s_2 \mathbf{v}_2 / |\mathbf{v}_1 \times \mathbf{v}_2|, \quad \mathbf{w}_4 = (1-s_1)\mathbf{v}_1 / |\mathbf{v}_1 \times \mathbf{v}_2|, \tag{4.60}$$

and

$$\mathbf{v}_1 = (\mathbf{x}_2 - \mathbf{x}_1) + s_2(\mathbf{x}_1 - \mathbf{x}_2 + \mathbf{x}_3\text{-}\mathbf{x}_4),$$
$$\mathbf{v}_2 = (\mathbf{x}_4 - \mathbf{x}_1) + s_1(\mathbf{x}_1 - \mathbf{x}_2 + \mathbf{x}_3 - \mathbf{x}_4).$$

The mapping from a unit square with isoparametric coordinates s_1 and s_2 to the quadrilateral element with coordinates $\mathbf{x}_1 \ldots \mathbf{x}_4$ is shown in Figure 4.1. This mapping is given by

$$\mathbf{x}(s_1, s_2) = (1-s_2)\left[(1-s_1)\mathbf{x}_1 + s_1\mathbf{x}_2\right] + s_2\left[(1-s_1)\mathbf{x}_4 + s_1\mathbf{x}_3\right]. \tag{4.61}$$

For an arbitrary quadrilateral element we have

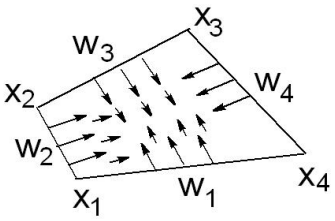

Figure 4.1: *Edge functions associated with a quadrilateral element.*

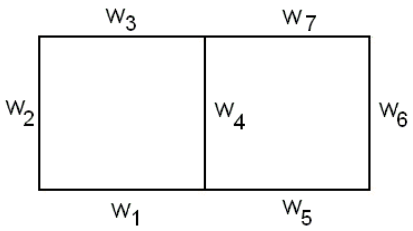

Figure 4.2: *Edge functions associated with two neighbor quadrilateral elements.*

$$R_{kl} = \int_0^1 \int_0^1 \frac{1}{\sigma} f^2(\mathbf{x}(s_1, s_2)) \mathbf{w}_l(\mathbf{x}(s_1, s_2)) \cdot \mathbf{w}_k(\mathbf{x}(s_1, s_2)) \left| \frac{\partial \mathbf{x}}{\partial s_1} \times \frac{\partial \mathbf{x}}{\partial s_2} \right| ds_1 ds_2,$$

where $\frac{\partial \mathbf{x}}{\partial s_1} = \mathbf{v}_1$ and $\frac{\partial \mathbf{x}}{\partial s_2} = \mathbf{v}_2$. Since the quadrilateral element has no edges at the boundary, then $f(\mathbf{x}(s_1, s_2)) = 1$. For example, the element R_{12} becomes

$$R_{12} = \frac{1}{\sigma} \int_0^1 \int_0^1 -s_1(1 - s_2) \frac{\mathbf{v}_2 \cdot \mathbf{v}_1}{|\mathbf{v}_1 \times \mathbf{v}_2|^2} |\mathbf{v}_1 \times \mathbf{v}_2| \, ds_1 ds_2.$$

Since we consider a single quadrilateral element Ω_j with unite edges, then the edge functions have a simple structure:

$$\begin{aligned}
\mathbf{w}_1 &= (1 - s_2)\mathbf{y}^0 / |\mathbf{v}_1 \times \mathbf{v}_2|, \\
\mathbf{w}_2 &= -s_1 \mathbf{x}^0 / |\mathbf{v}_1 \times \mathbf{v}_2|, \\
\mathbf{w}_3 &= -s_2 \mathbf{y}^0 / |\mathbf{v}_1 \times \mathbf{v}_2|, \\
\mathbf{w}_4 &= (1 - s_1)\mathbf{x}^0 / |\mathbf{v}_1 \times \mathbf{v}_2|,
\end{aligned}$$

where $|\mathbf{v}_1 \times \mathbf{v}_2| = 1$, and $\mathbf{x}^0$ and $\mathbf{y}^0$ are the canonical basis vectors. Therefore, the properties of R can be defined by the integrals over Ω_h from the elements of the following matrix

$$\frac{1}{\sigma} \begin{pmatrix} (1 - s_2)^2 & 0 & -s_2(1 - s_2) & 0 \\ 0 & s_1^2 & 0 & -s_1(1 - s_1) \\ -s_2(1 - s_2) & 0 & s_2^2 & 0 \\ 0 & -s_1(1 - s_1) & 0 & (1 - s_1)^2 \end{pmatrix}. \tag{4.62}$$

After integration over canonical square we get a matrix with constant factors

$$R = \frac{1}{\sigma} \begin{pmatrix} 1/3 & 0 & 1/4 & 0 \\ 0 & 1/3 & 0 & 1/4 \\ 1/4 & 0 & 1/3 & 0 \\ 0 & 1/4 & 0 & 1/3 \end{pmatrix}, \tag{4.63}$$

which is positive definite. □

In fact (4.63) corresponds to a stamp which can be used for the construction of R in case of more than one quadrilateral element. For instance, two neighbor elements with edges of unit length, presented in Figure 4.2, have a common edge with the edge function $\mathbf{w}_4$.

In this case R becomes as follows

$$R = \frac{1}{\sigma}\begin{pmatrix} 1/3 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 1/4 & 0 & 0 & 0 \\ 1/4 & 0 & 1/3 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 2/3 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/4 \\ 0 & 0 & 0 & 1/4 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/4 & 0 & 1/3 \end{pmatrix},$$

which is also positive definite matrix.

The following lemma is important because it shows that P can be treated as an incidence matrix. The proof of the lemma can be found in [93].

Lemma 2. *Let J be the Jacobian defined by the transformation (4.61) for $\boldsymbol{x} \in \Omega_j$, $j = 1, \ldots, \mathcal{N}$. For one of the following conditions*
1) $f(\boldsymbol{x}) \equiv 1$,
2) $\nabla f \cdot \boldsymbol{w}_k = 0$, and $J = |\Omega_j|$,
the matrix P has the form

$$P_{kj} = \begin{cases} \pm 1, & \text{if } k\text{-th edge belongs to } \Omega_j \\ 0, & \text{otherwise.} \end{cases}$$

4.3 Original circuit used in Fasterix

As we saw in Section 4.2, Fasterix translates electromagnetic properties of the interconnect system up to a certain maximum frequency into an original circuit which is described by the system of Kirchhoff equations (4.55)–(4.56). In other words, discretization of the domain of the conductor into quadrilateral elements and specific choice of basic functions results in a system which has the appearance of Kirchhoff equations. The variables of the system represent currents and potentials at the $\mathcal{N}$ nodes and the $\mathcal{F}$ branches:

$$(R + sL)\mathbf{I} - P\mathbf{V} = 0, \tag{4.64}$$

$$P^T\mathbf{I} + sC\mathbf{V} = \mathbf{J}, \tag{4.65}$$

where $R \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$ is the resistance matrix, $L \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$ is the inductance matrix, $P \in \{1, 0, -1\}^{\mathcal{F}\times\mathcal{N}}$ is an incidence matrix, $C \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is the capacitance matrix, $\mathbf{I} \in C^{\mathcal{F}}$ is a vector of currents flowing in the branches, $\mathbf{V} \in C^{\mathcal{N}}$ is a vector of voltages at the nodes. The difference between (4.64)–(4.65) and (4.55)–(4.56) is the vector $\mathbf{J} \in C^{\mathcal{N}}$ which collects the terminal currents (unknowns) flowing into the interconnection system. Value

s is a complex number with negative imaginary part, i.e., $s = -j\omega$. Matrices R, L are symmetric positive definite, and C is symmetric positive semidefinite. We recall that (4.64)–(4.65) describe the interconnect system up to the certain frequency but do *not* correspond to MNA equations since R is not diagonal.

To obtain the admittance matrix which describes the input-output behavior of the original circuit, we suppose that the currents $\mathbf{J}$ flows only through the ports of the system (i.e., upon permutation $\mathbf{J}^T = \begin{pmatrix} \mathbf{J}_p^T & \mathbf{J}_i^T \end{pmatrix}$ and $\mathbf{J}_i^T = 0$). Here index p stands for external nodes (ports) and i stands for internal nodes. Splitting $\mathbf{V}$, P and C into corresponding blocks, we rewrite (4.64)–(4.65) as follows

$$\begin{pmatrix} R + sL & -P_i & -P_p \\ P_i^T & sC_{ii} & sC_{ip} \\ P_p^T & sC_{pi} & sC_{pp} \end{pmatrix} \begin{pmatrix} \mathbf{I} \\ \mathbf{V}_i \\ \mathbf{V}_p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{J}_p \end{pmatrix}. \tag{4.66}$$

Eliminating $\mathbf{I}$ and $\mathbf{V}_i$ from (4.66), one obtains the admittance matrix $Y(s) : \mathbb{C} \to \mathbb{C}^{p \times p}$ of the original non-reduced circuit:

$$\mathbf{J}_p = \underbrace{\left(B_o^T(s)(\widetilde{G} + s\widetilde{C})^{-1} B_i(s) + sC_{pp} \right)}_{Y(s)} \mathbf{V}_p, \tag{4.67}$$

where

$$\widetilde{B}_o^T(s) = \begin{pmatrix} P_p^T & sC_{ip}^T \end{pmatrix}, \quad \widetilde{G} = \begin{pmatrix} R & -P_i \\ P_i^T & 0 \end{pmatrix},$$

$$\widetilde{C} = \begin{pmatrix} L & 0 \\ 0 & C_{ii} \end{pmatrix}, \quad \widetilde{B}_i^T(s) = \begin{pmatrix} P_p^T & -sC_{ip}^T \end{pmatrix}.$$

We will consider $Y(s)$ as the admittance of the original circuit which can be further used for comparison with reduced order models. The important observation is that $Y(s)$ is not in the first-order form because $B_i(s)$ and $B_o(s)$ are given as the functions dependent on $s = -i\omega$.

4.4 Super node algorithm

In this section we review a reduction technique, the super node algorithm. We also show a real example which demonstrates a convergence problem resulted from the transient simulation of the reduced circuit and make a remark on the use of projection based model order reduction methods as an alternative for performing the reduction.

4.4.1 Admittance matrix of the super nodes circuit

As it was discussed above, the original circuit described by (4.64)–(4.65) contains $\mathcal{N}$ nodes (ports and internal nodes) and $\mathcal{F}$ branches. In Fasterix the reduced circuit is built on the so-called super nodes, i.e., ports and a subset of internal nodes. Fasterix defines the super nodes in such a way that there exists a conducting path, no greater in length than a predefined small fraction of the minimum wavelength λ_{min} between every point on the conductors and at least one of the super nodes [64]:

$$\lambda_{min} = \frac{1}{\Theta\sqrt{\epsilon_r \epsilon_0 \mu_0}}, \tag{4.68}$$

where Θ is the highest simulation frequency, ϵ_r is the relative permittivity of the material, ϵ_0 is the permittivity of free space, and μ_0 is the permeability of free space. The fraction of wavelength normally used to define an electrically short distance is $\lambda_{min}/10$, and this has been shown to give a good accuracy [64]. We will use the following notation: N denotes super nodes and N' denotes all other nodes. Thus the vectors $\mathbf{V}$, $\mathbf{J}$ and the matrices P, C in (4.64)–(4.65) can be partitioned into blocks. Therefore, (4.64)–(4.65) can be rewritten as

$$\underbrace{\begin{pmatrix} R+sL & -P_{N'} & -P_N \\ P_{N'}^T & sC_{N'N'} & sC_{N'N} \\ P_N^T & sC_{NN'} & sC_{NN} \end{pmatrix}}_{A} \begin{pmatrix} \mathbf{I} \\ \mathbf{V}_{N'} \\ \mathbf{V}_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathbf{J}_N \end{pmatrix}, \tag{4.69}$$

where $\mathbf{V}_N \in C^{\mathcal{N}_1}$, $\mathbf{V}_{N'} \in C^{\mathcal{N}_2}$, $\mathbf{J}_N \in C^{\mathcal{N}_1}$, $\mathbf{J}_{N'} \in C^{\mathcal{N}_2}$, $P_N \in \{1,0,-1\}^{\mathcal{F}\times\mathcal{N}_1}$, $P_{N'} \in \{1,0,-1\}^{\mathcal{F}\times\mathcal{N}_2}$, $C_{NN} \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_1}$, $C_{NN'} \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_2}$, $C_{N'N} \in \mathbb{R}^{\mathcal{N}_2\times\mathcal{N}_1}$, $C_{N'N'} \in \mathbb{R}^{\mathcal{N}_2\times\mathcal{N}_2}$. Matrix $P_{N'}$ has full column rank. It is supposed that no current flows into the nodes from the subset N', i.e., $\mathbf{J}_{N'} = 0$.

If we consider the voltage in the super nodes as an input $\mathbf{V}_N$, and currents flowing into the system through them as an output $\mathbf{J}_N$, we come to the following system:

$$\left(\underbrace{\begin{pmatrix} R & -P_{N'} \\ P_{N'}^T & 0 \end{pmatrix}}_{\hat{G}} + s \underbrace{\begin{pmatrix} L & 0 \\ 0 & C_{N'N'} \end{pmatrix}}_{\hat{C}} \right) \mathbf{x} = \underbrace{\begin{pmatrix} P_N \\ -sC_{N'N} \end{pmatrix}}_{B_i(s)} \mathbf{V}_N, \tag{4.70}$$

$$\mathbf{J}_N = \underbrace{\begin{pmatrix} P_N^T & sC_{N'N}^T \end{pmatrix}}_{B_o^T(s)} \mathbf{x} + sC_{NN}\mathbf{V}_N, \tag{4.71}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{I}^T & , \mathbf{V}_{N'}^T, \end{pmatrix}^T$. It should be noted that G in (4.70) is positive real and C

is positive semidefinite. Eliminating $\mathbf{x}$ from (4.70) and substituting it into (4.71), one obtains the following linear relation between $\mathbf{J}_N$ and $\mathbf{V}_N$, i.e.,

$$\mathbf{J}_N = \underbrace{\left(B_o^T(s)(\hat{G}+s\hat{C})^{-1}B_i(s)+sC_{NN}\right)}_{Y_1(s)}\mathbf{V}_N, \tag{4.72}$$

where $Y_1(s):\mathbb{C}\to\mathbb{C}^{\mathcal{N}_1\times\mathcal{N}_1}$ is the admittance matrix of the reduced circuit on the base of all super nodes. It has the same structure as $Y(s)$ in (4.67) but different dimension. To carry out frequency (time) domain analysis of the reduced circuit, Fasterix simplifies $Y_1(s)$ by using frequency approximations. After that Fasterix synthesizes an RCL circuit. In the next sections we will give a framework of these steps.

4.4.2 Approximations of $Y_1(s)$

Depending on the frequency range of interest, three different approximations for $Y_1(s)$ can be distinguished, namely, low, high and full frequency range approximations [20]. We will be mainly concentrated on the last one since it is necessary for construction of the reduced circuit to carry out transient analysis. Before doing it, we show a derivation of a preliminary approximation of $Y_1(s)$. Fasterix chooses discretization step, h, such that $k_0h \ll 1$, where k_0 is a free space wavenumber:

$$k_0h=\omega\sqrt{\epsilon_0\mu_0}h=2\pi\frac{\omega}{\Theta\Lambda}h\ll 1,\quad \text{for each } \omega\in[0,\Theta],$$

where Θ denotes the maximum frequency of operation and Λ denotes the wavelength associated with the maximum frequency of operation. When the mesh size, h, is sufficiently small, elements $i\omega L_{jk}\approx\sqrt{\frac{\mu_0}{\epsilon_0}}ik_0h$ and $i\omega C_{mn}\approx\sqrt{\frac{\mu_0}{\epsilon_0}}ik_0h$ are frequency independent [20]. Thus, the vectors $\mathbf{V}_{N'}$ and $\mathbf{I}$ can be expanded in powers of (ik_0h) as follows

$$\mathbf{V}_{N'}\approx\mathbf{V}_0+\mathbf{V}_1, \tag{4.73}$$
$$\mathbf{I}\approx\mathbf{I}_0+\mathbf{I}_1, \tag{4.74}$$

where $\mathbf{V}_0$ and $\mathbf{I}_0$ have the order $O(1)$ and $\mathbf{V}_1$ and $\mathbf{I}_1$ have the order $O(ik_0h)$. By substituting expansions (4.73)–(4.74) into (4.70) and gathering the terms with the same powers, one obtains the two sets of equations:

$$(R+sL)\mathbf{I}_0-P_{N'}\mathbf{V}_0=P_N\mathbf{V}_N, \tag{4.75}$$
$$-P_{N'}^T\mathbf{I}_0=0, \tag{4.76}$$

$$(R + sL)\mathbf{I}_1 - P_{N'}\mathbf{V}_1 = 0, \tag{4.77}$$

$$-P_{N'}^T\mathbf{I}_1 = s(C_{N'N'}\mathbf{V}_0 + C_{N'N}\mathbf{V}_N). \tag{4.78}$$

By substituting expansions (4.73)–(4.74) into (4.71), the preliminary approximation of admittance matrix $Y_1(s)$ becomes

$$Y_2(s) = P_N^T(\mathbf{I}_0 + \mathbf{I}_1) + sC_{NN'}\mathbf{V}_0 + sC_{NN}\mathbf{V}_N + O((ik_0h)^2). \tag{4.79}$$

The error between $Y_1(s)$ and $Y_2(s)$ is of the order $O((ik_0h)^2)$.

High frequency range approximation

This type of approximation was created to construct a reduced circuit to carry out simulations in frequency domain. The high frequency range approximation of $Y_1(s)$ is constructed as

$$Y_3(s) = s^{-2}Y_R + s^{-1}Y_L + Y_G + sY_C, \tag{4.80}$$

where $Y_R, Y_L, Y_G, Y_C \in \mathbb{R}^{\mathcal{N}_1 \times \mathcal{N}_1}$. Details about calculation of these matrices can be found in Appendix A.1. Note that representation (4.80) is not suitable for time domain simulations since the inverse Laplace transform of the first summand is a ramp function:

$$L^{-1}\left\{\frac{1}{s^2}\right\} = tu(t), \tag{4.81}$$

which is unbounded. Therefore, (4.80) does not correspond to a bounded-input bounded-output stable system.

Full frequency range approximation

This type of approximation is required for construction of the reduced circuit to carry out time domain simulations. Therefore, it is important to investigate it in order to find a reason of unstable simulations. In [93] the full frequency range approximation of $Y_1(s)$ is constructed as

$$Y_3(s) = Y_{RL}(s) + sY_C. \tag{4.82}$$

The second summand, Y_C, stands for the capacitance admittance matrix from the high frequency range approximation. The first summand, Y_{RL}, stands for the resistor-inductance contribution and can be presented in the pole-residue form:

$$Y_{RL}(s) = P_N^T\Psi\left(\Psi^T(R + sL)\Psi\right)^{-1}\Psi^T P_N =$$

$$\sum_{i=1}^{n} \frac{\left(\Psi^T P_N \mathbf{x}_i\right)\left(\mathbf{y}_i^* P_N^T \Psi\right)}{(s-\lambda_i)}, \quad n = \mathcal{F} - \mathcal{N}_2. \tag{4.83}$$

λ_i are the eigenvalues of the matrix pencil $(\Psi^T L\Psi, -\Psi^T R\Psi)$. The matrix $\Psi \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}-\mathcal{N}_2}$ is such that its columns span the null space of $P_{N'}^T$:

$$P_{N'}^T \Psi = 0, \quad \Psi^T P_{N'} = 0.$$

It should be noted that $\Psi^T L\Psi$ and $\Psi^T R\Psi$ are real symmetric matrices (case of Hermitian matrices), and $-\Psi^T R\Psi$ is negative definite. Therefore, $\lambda_i \in \mathbb{R}$ and $\lambda_i < 0$. Moreover, left and right eigenvectors $\mathbf{x}_i$, $\mathbf{y}_i \in \mathbb{R}^{\mathcal{N}_1}$ are equal. It is enough to conclude that $Y_{RL}(s)$ describes a stable system.

Algorithm 2 shows how to compute $Y_{RL}(s)$ in (4.83). Calculation of the generalized eigenvalues can be performed by the QZ method (complexity $O(\mathcal{N}_1^3)$) [32]. However, in order to build a reduced circuit described by $Y_3(s)$ and to be able to carry out simulations in time domain, $Y_3(s)$ has to be realized as an RLC circuit. A netlist of such circuit can be used in a circuit simulator program. In the next section we review the procedure for realization used in Fasterix.

Algorithm 2 Computation of poles and residues of $Y_{RL}(s)$.

INPUT: $R \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, $L \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, $P \in \{1,0,-1\}^{\mathcal{F}\times\mathcal{N}}$, $\mathcal{N}_1, \mathcal{N}_2$;
OUTPUT: $H_i \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_1}$, $\lambda_i \in \mathbb{R}$, $i = 1,\ldots,\mathcal{F}-\mathcal{N}_2$;
1. Define $P_N \in \{1,0,-1\}^{\mathcal{F}\times\mathcal{N}_1}$, $P_{N'} \in \{1,0,-1\}^{\mathcal{F}\times\mathcal{N}_2}$;
2. Calculate $\Psi \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}-\mathcal{N}_2}$ such that $P_{N'}^T\Psi = 0$ and $\Psi^T P_{N'} = 0$;
3. $A := \Psi^T R\Psi$, $B := \Psi^T L\Psi$;
4. Solve generalized eigenvalue problem $-A\mathbf{x} = \lambda B\mathbf{x}$;
5. Compute $H_i := \left(\Psi^T P_N \mathbf{x}_i\right)\left(\mathbf{x}_i^* P_N^T \Psi\right)$, for $i = 1,...\mathcal{F}-\mathcal{N}_2$.

4.4.3 Frequency fitting and realization

First, we introduce the concept of branch admittance for a given admittance matrix $Y(s)$. The branch admittance between a node i and the ground is defined by

$$y_{ii} = \sum_{j=1}^{n} Y_{ij}. \tag{4.84}$$

The branch admittance between the nodes i node j is

$$y_{ij} = -Y_{ij}, \quad i \neq j. \tag{4.85}$$

Let $y_3(s)$ denote the branch admittance of $Y_3(s) = Y_{RL}(s) + sY_C$. Since calculation of all generalized eigenvalues, λ_i, of $Y_{RL}(s)$ in (4.83) may be a time consuming process (QZ method has complexity $O(\mathcal{N}_1^3)$; therefore, for $\mathcal{N}_1 > 10^3$ this method is very time consuming), Fasterix approximates $y_3(s)$ with $m < n$ terms. For this goal, a set of $m+1$ match frequencies, s_k, is chosen. The set consists of some large negative values between $-\Theta$ and $-\max(\lambda_i)$, and some small negative values between $-\min(\lambda_i)$ and 0. Θ denotes the maximum frequency for which the conductors have to function. Solving the following set of $m+1$ equations

$$s_k y_{C,ij} + \sum_{l=1}^{m} \frac{\widetilde{H}_{l,ij}}{(s_k - \lambda_l)} = y_{3,ij}(s_k), \quad k = 1, ...m+1. \tag{4.86}$$

for the coefficients $y_{C,ij}$ and $\widetilde{H}_{l,ij}$, $l = 1, \ldots, m$ is equivalent to determine the approximation of $y_{3,ij}(s)$ with $m < n$ terms (usually, $2 \leq m \leq 8$). Note that the set of equations (4.86) has to be solved for $i, j = 1, \ldots \mathcal{N}_1$.

The synthesized circuit consists of $\mathcal{N}_1$ super nodes. Between every pair of circuit nodes, there is an RLC-branch, and between each super node and the ground, there is the RLC-branch as well. Each RLC branch consists of m parallel connections of a resistor R in a series with an inductor L, and in parallel with a capacitor C (Figure 4.4):

$$R_l = -\lambda_l \widetilde{H}_{l,ij}^{-1}, \quad L_l = \widetilde{H}_{l,ij}^{-1}, \quad C = y_{C,ij}, \quad l = 1, \ldots m, \quad i, j = 1, \ldots \mathcal{N}_1. \tag{4.87}$$

Realization of the circuit in (4.87) corresponds to the Foster multi-port network realiza-

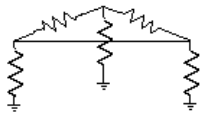

Figure 4.3: *Schematic representation of the reduced circuit consisting of three super nodes.*

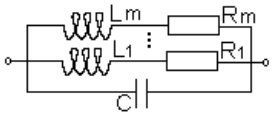

Figure 4.4: *An RLC branch of the reduced circuit.*

tion [58], [85] which we discussed in Section 2.4.6. Example of a circuit based on three super nodes is shown in Figure 4.3, where each branch has a form as in Figure 4.4. In practice Fasterix omits resistors and inductors between each super node and the ground since they usually have very large values.

4.4.4 Summary of the super node algorithm

Algorithm 3 shows the main steps of the SNA. Computing Y_C (step 2) requires solving four sets of equations which can be found in Appendix A.1. In Fasterix computation of generalized eigenvalues (step 3) is done according to the algorithm developed by Parlett and Reid [68], [92]. We recall that the reduced circuit obtained by the Algorithm 3, when

Algorithm 3 The Super Node Algorithm

INPUT: $R \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, $L \in \mathbb{R}^{\mathcal{F}\times\mathcal{F}}$, $C \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$, $P \in \{1, 0, -1\}^{\mathcal{F}\times\mathcal{N}}$, $\mathcal{N}_1, \mathcal{N}_2$;
OUTPUT: Netlist of the reduced circuit;
1. Perform steps 1-3 of the Algorithm1;
2. Compute $Y_C \in \mathbb{R}^{\mathcal{N}_1\times\mathcal{N}_1}$;
3. Choose m. Calculate $m-1$ the smallest and the largest eigenvalues λ of the generalized eigenvalue problem $-A\mathbf{x} = \lambda B\mathbf{x}$;
4. Set match frequencies, such that $s_1 = 0$, $s_{m+1} = -\Theta$, $s_k = -\lambda_k$, $k = 2, \dots m$;
5. Solve the set of $m+1$ equations in (4.86) for $y_{C,ij}$, $\widetilde{H}_{l,ij}$, $i, j = 1, \dots, \mathcal{N}_1$;
6. Construct a netlist according to (4.87).

observed from its ports, is equivalent to the interconnect system up to the maximum frequency Θ. In the next section we show a numerical example which validates the problem with the SNA.

4.4.5 Numerical example

We will consider the two parallel striplines model which is shown in Figure 2.4 (Section 2.2.3). For the maximum frequency 1 MHz, Fasterix generates a mesh with 10 elements. Four elements of the mesh are quadrilateral and six have the form of edges. Corresponding matrices $R \in \mathbb{R}^{8\times8}$, $L \in \mathbb{L}^{8\times8}$ and $C \in \mathbb{C}^{10\times10}$ in (4.64)–(4.65) are sparse. To build the reduced circuit, Fasterix chooses 6 super nodes ($\mathcal{N}_1 = 6$), shown in Figure 2.4 black dots, and applies the SNA. In (4.82) Y_C is indefinite with one negative eigenvalue, and poles λ_i are real and stable. Five frequencies in the range from 0 till 6 MHz were chosen by Fasterix for frequency fitting. Data for the original and reduced circuits are presented in Table 4.1. It can be seen that the number of elements in the reduced circuit is slightly larger than in the original one. In this case it is not significant because initially the model is very small. For carrying out simulations we used PSTAR [2] which is Philips (NXP) circuit simulator. Figure 4.5 shows a comparison between the original admittance matrix and the admittance matrix obtained after the SNA for the entry (IN_2, OUT_2). One can observe a good agreement on a wide range of frequencies. However, it is not the case for the transient analysis.

Table 4.1: *Data for the original and reduced circuits of the two printed striplines model from Fasterix*

model	ports	R	L	C	L_{mutual}	time
original	5	8	8	10	20	0.01
reduced (SNA)	5	60	60	21	0	0.01

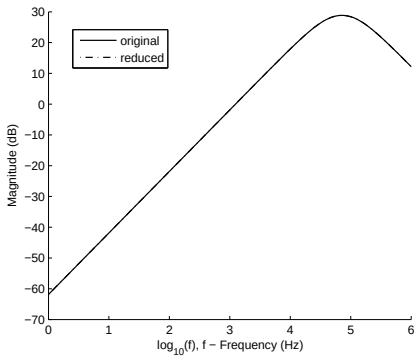

Figure 4.5: *Comparison in frequency domain of admittances of the original and reduced circuits by the SNA for the entry* (IN_2, OUT_2).

For the transient analysis, a trapezoidal pulse having rise/fall times of 1 ps and pulse width of 1 ns is applied to the pins of the lower strip. A 50 Ω resistor R_{out} is connected between two ports of the upper strip. The voltage is measured over R_{out} and regarded as output. We recall that the transient response of the reduced circuit, presented in Figure 3.1 (Section 3.2.1), is unstable which can be explained by the non-passivity of the reduced model. Moreover, a pole-zero analysis of the reduced circuit, made in PSTAR, detected a few unstable poles which means that the circuit is not passive.

In the next section we will explain why the SNA delivers non-passive models and present a way to overcome this problem.

4.5 Positive realness and passivity

An important property of RLC circuits is passivity. Roughly speaking, a system is passive if it does not generate energy. As we discussed in Section 2.4.4, passivity is closely related to the positive realness of the transfer function [8], [66]. We recall that the transfer function $Y(s) : \mathbb{C} \to \mathbb{C}^{n\times n}$ (in our case admittance function), is positive real if the

following three conditions are satisfied:

1. $Y(s)$ is analytic for all s with $Re(s) > 0$ (i.e., there are no poles λ with $Re(\lambda) > 0$),
2. $Y^*(s) = Y(\bar{s})$ for all s with $Re(s) > 0$,
3. $Y(s) + Y^*(s) \geq 0$ for all s with $Re(s) > 0$ (i.e., $Y(s) + Y^*(s)$ is nonnegative definite for all s with $Re(s) > 0$).

Here $\bar{s}$ denotes the complex conjugate of s and * denotes the complex conjugate transpose.

In Theorem 7 which is presented below we will prove that the admittance matrix, $Y_1(s)$, defined in (4.72) is positive real. To proceed with the proof, we need to introduce some preliminary results.

Theorem 6. *A matrix $A \in \mathbb{C}^{n\times n}$ is positive real (nonnegative real) if and only if the Hermitian part of A, i.e., $\frac{1}{2}(A + A^*)$ is symmetric positive definite (positive semidefinite), i.e. $\frac{1}{2}(A + A^*) > 0$ ($\frac{1}{2}(A + A^*) \geq 0$).*

This theorem is a well-known result and the proof for it can be found, for instance, in [42]. On the other hand, the following lemma is a result which we proved by ourself and we did not find any statement nor proof in literature sources.

Lemma 3. *Let $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \in \mathbb{C}^{n\times n}$ be nonnegative real matrix. The Schur complement $A_s = A_{22} - A_{21}A_{11}^{-1}A_{12}$ of the block A_{11} of A is nonnegative real.*

Proof. From the fact that A is a nonnegative real matrix, it follows that A_{11} is nonnegative real as well. Now we notice that

$$\begin{bmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{bmatrix} (A + A^*) \begin{bmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{bmatrix}^* = \tag{4.88}$$

$$\begin{bmatrix} A_{11} + A_{11}^* & A_{12} - A_{11}A_{11}^{-*}A_{21}^* \\ (A_{12} - A_{11}A_{11}^{-*}A_{21}^*)^* & A_s + A_s^* \end{bmatrix},$$

where $A_s = A_{22} - A_{21}A_{11}^{-1}A_{12}$. The left side of (4.88) is positive semidefinite (note, $A + A^*$ is multiplied on the left and on the right sides by the invertible matrix; by the Sylvester law of Inertia [32], the result is positive semidefinite matrix). Thus $A_s + A_s^*$ is positive semidefinite as a principal minor of the positive semidefinite matrix. By Theorem 6, A_s is nonnegative real. □

With the help of the preliminary results, we will prove Theorem 7.

Theorem 7. *The admittance $Y_1(s)$ in (4.72) is positive real.*

Proof. First, $Y_1(s)$ is analytic for all s with $Re(s) > 0$. Indeed, all poles defined by the generalized eigenvalue problem $-\hat{G}\mathbf{x} = \hat{C}\lambda\mathbf{x}$ ($\hat{G}$ is positive real, $\hat{C}$ is positive semidefinite) are nonnegative. It can be easily checked that the second condition of positive realness holds true. To show that the third condition is satisfied, we notice that $Y_1(s)$ is the Schur complement of the block

$$\begin{pmatrix} R + sL & -P_{N'} \\ P_{N'}^T & sC_{N'N'} \end{pmatrix}$$

of the matrix $A(s)$ defied in (4.69), which is nonnegative real for each s with $Re(s) > 0$:

$$A + A^* = \begin{pmatrix} 2(R + Re(s)L) & 0 & 0 \\ 0 & 2Re(s)C_{N'N'} & 2Re(s)C_{N'N} \\ 0 & 2Re(s)C_{NN'} & 2Re(s)C_{NN} \end{pmatrix} \geq 0 \tag{4.89}$$

By Lemma 3, $Y_1(s)$ is nonnegative real for each s with $Re(s) > 0$. Thus, $Y_1(s)$ describes a passive system. □

Similarly it can be proved that the admittance matrix $Y(s)$ of the original circuit in (4.67) is positive real. From the other hand, positive realness of $Y_3(s) = Y_{RL} + sY_C$ in (4.82) is not guaranteed. What stays positive real is the admittance matrix Y_{RL}. This we will prove in Lemma 4 which is presented below. However, in practice Y_C might be indefinite, i.e., Y_C has positive and negative eigenvalues. Since the frequency fitting step in the SNA applied to $Y_3(s)$ may destroy positive realness, passivity of the reduced circuit is not guaranteed.

Lemma 4. *The admittance matrix $Y_{RL}(s)$ in (4.83) is positive real.*

Proof. In Section 4.4.2 it was shown that all poles λ_i of Y_{RL} which are eigenvalues of the matrix pencil $\left(\Psi^T L\Psi, -\Psi^T R\Psi\right)$, are negative. Therefore, the system is stable. It is trivial to check out the second condition for positive realness. Let $B^T = P_N^T\Psi$, $\widetilde{R} = \Psi^T R\Psi$, and $\widetilde{L} = \Psi^T L\Psi$. We will show that the third condition for positive realness is satisfied:

$$Y_{RL}^*(s) + Y_{RL}(s) = B^T\left(\widetilde{R} + s\widetilde{L}\right)^{-*} B + B^T\left(\widetilde{R} + s\widetilde{L}\right)^{-1} B = \tag{4.90}$$

$$= B^T\left(\widetilde{R} + s\widetilde{L}\right)^{-*}\left(\left(\widetilde{R} + s\widetilde{L}\right) + \left(\widetilde{R} + s\widetilde{L}\right)^*\right)\left(\widetilde{R} + s\widetilde{L}\right)^{-1} B =$$

$$= \mathbf{y}^*\left(\left(\widetilde{R} + s\widetilde{L}\right) + \left(\widetilde{R} + s\widetilde{L}\right)^*\right)\mathbf{y},$$

with $\mathbf{y} = (\widetilde{R} + s\widetilde{L})^{-1}B$. Thus it is sufficient to prove the positive realness of $W(s) = \widetilde{R} + s\widetilde{L}$. For $s = \sigma + i\omega$ with $\sigma > 0$ we have:

$$W^*(s) + W(s) = (\widetilde{R} + s\widetilde{L})^* + \widetilde{R} + s\widetilde{L} = 2\widetilde{R} + 2\sigma\widetilde{L},$$

which is nonnegative definite. Thus, $Y_{RL}(s)$ is positive real. □

4.5.1 Comparison with projection based reduction methods

In the previous section we have proved that the super node algorithm does not preserve passivity. However, there exist some model order reduction techniques which preserve stability and passivity [77]. Therefore, the question is whether one can apply such methods instead of the super node algorithm to deliver passive reduced models.

Since $Y(s)$ in (4.67) is not in the first-order form, we cannot directly apply general MOR techniques. By adding extra equations to the system (4.70)–(4.71), as it is done in [41], one can write the voltage to current transfer in the familiar form:

$$\mathbf{J}_p = (B^T(G + sC)B) \cdot \mathbf{V}_p, \tag{4.91}$$

where $B \in \mathbb{R}^{\mathcal{N}+\mathcal{F}+p\times p}$ does not depend on frequency any more and

$G, C \in \mathbb{R}^{\mathcal{N}+\mathcal{F}+p\times\mathcal{N}+\mathcal{F}+p}$. If one is only interested in the port behavior, one can apply general MOR techniques to (4.91). In [41], PRIMA and SVD-Laguerre algorithms showed a good approximation with the original systems in frequency and time domain while preserving passivity. However, if simulations of radiation have to be performed in Fasterix, the super nodes have to be preserved in the reduced circuit. Treating the super nodes as ports, will lead to an original circuit with many ports (up to a few hundreds). Applying projection based techniques will automatically create fill-in in the reduced matrices, and as a result, synthesized reduced circuit may have many electrical components (usually more than the number of elements in the original circuit) [48].

Therefore, there is a necessity to modify the SNA to deliver reduced passive circuits which preferably have the same number of elements as the circuits reduced by the original SNA. In the next section we will consider two approaches for modification.

4.6 Passivity enforcement

In this section we will show two methods to overcome the problem of non-passive reduced circuits delivered by the SNA. The first approach is based on applying modal

approximation directly to the admittance matrix Y_{RL}. This approach has exploratory purposes rather than practical. This is due to the fact that the modal approximation applied to the admittance function requires to keep most of the real poles for the approximation to be accurate enough. As a result, synthesized reduced circuit by the Foster realization may have larger amount of elements than the original one.

The second method is based on applying a passivity enforcement technique [38] to the admittance function after frequency fitting step in the SNA. Since the SNA delivers a very good approximation of the admittance function in frequency domain, we have decided to modify the admittance matrix from the SNA, rather than constructing a new one. We will show that this method delivers passive reduced circuits and keeps the same amount of elements as by the SNA.

4.6.1 Passivity enforcement by the modal approximation

Numerical example in Section 4.4.5 has shown that the simulation involving a fitted $Y_3(s)$ may lead to unstable simulations, even though the elements of $Y_3(s)$ have been fitted in frequency domain using stable poles. To overcome this problem we propose the following passivity enforcement procedure, which we have published in [89]. For the term Y_{RL} we are going to apply modal approximation, and for the term sY_C we enforce positive realness. If both summands in (4.82) are positive real, then $Y_3(s)$ is positive real as well.

First we consider the term sY_C. Following the eigendecomposition

$$Y_C = \mathbf{V} diag(\sigma_1, \sigma_2, \ldots, \sigma_{\mathcal{N}_1}) \mathbf{V}^{-1},$$

all negative eigenvalues σ_i are set to zero. Subsequently, the matrix is reconstructed through the operation

$$\widetilde{Y}_C = \mathbf{V} diag(\widetilde{\sigma}_1, \widetilde{\sigma}_2, \ldots, \widetilde{\sigma}_{\mathcal{N}_1}) \mathbf{V}^{-1},$$

where the modified quantities are denoted with "~". Since in practice Y_C may have a few negative eigenvalues, this procedure allows us to get positive real $s\widetilde{Y}_C$ without loosing too much accuracy.

According to Lemma 4, $Y_{RL}(s)$ is positive real. However, the number of terms in $Y_{RL}(s)$ is related to the number of RL elements in the reduced circuit as $O(n\mathcal{N}_1^2)$, where $\mathcal{N}_1$ is the number of super nodes, and n denotes the number of poles in the pole-residue representation of $Y_{RL}(s)$. Taking it into account, we are interested to obtain an efficient approximation of $Y_{RL}(s)$ which consists of $k < n$ terms and determines the effective admittance function behavior. Positive realness of the new approximation must be preserved. For this goal we use modal approximation. A general framework for modal

approximation has been introduced in Section 3.1.2.

For the admittance matrix

$$Y_{RL}(s) = P_N^T \Psi \left(\Psi^T (R + sL) \Psi \right)^{-1} \Psi^T P_N,$$

we have $\Psi^T R \Psi = (\Psi^T R \Psi)^T \geq 0$ and $\Psi^T L \Psi = (\Psi^T L \Psi)^T \geq 0$. Hence, all generalized eigenvalues are real, and right and left eigenvectors are the same. Hence,

$$H_i = ((\Psi^T P_N)^T \mathbf{x}_i)(\mathbf{x}_i^T (\Psi^T P_N)) = ((\Psi^T P_N)^T \mathbf{x}_i)^2 \geq 0.$$

Therefore, all the summands in the partial fraction expansion

$$Y_{RL}(s) = \sum_{i=1}^{n} \frac{\left(\Psi^T P_N \mathbf{x}_i \right) \left(\mathbf{x}_i^* P_N^T \Psi \right)}{(s - \lambda_i)}, \quad n = \mathcal{F} - \mathcal{N}_2,$$

are passive. Consequently, any modal approximation of the passive admittance matrix $Y_{RL}(s)$ is passive. Applying modal approximation to Y_{RL}, and reconstructing Y_C (if necessary), the full frequency range approximation, $Y_3(s)$, in (4.82) can be approximated as

$$\widetilde{Y}_3(s) \approx \sum_{i=1}^{k} \frac{H_i}{s - \lambda_i} + s\widetilde{Y}_C, \quad k < n, \tag{4.92}$$

which is positive real function. Synthesis of $\widetilde{Y}_3(s)$ can be performed by the Foster realization as we discussed in Section 2.4.6.

Thus, a passive version of the SNA is based on Algorithm 3, where instead of the steps 3-6 one has to make: (1) Y_C positive definite (if necessary); (2) modal approximation of Y_{RL}; (3) synthesis of $\widetilde{Y}_3(s)$ by the Foster realization.

4.6.2 Example: two parallel striplines model

This model was considered in Section 4.4.5 to show instabilities in the simulations due to non-passivity of the reduced circuit.

To obtain a passive model, Y_C in (4.82) was made positive definite. Since initially the system is small, we do not apply modal approximation to $Y_{RL}(s)$ and keep all $n = 4$ terms. To construct a netlist of the compressed circuit, Foster realization was applied to $Y_3(s)$. Figure 4.10 shows simulation in time domain made in PSTAR. (Settings for time domain simulations were taken from Section 4.4.5). It clearly can be seen that the signal does not blow up any more and remains bounded. Moreover, the pole-zero analysis did not detect unstable poles. Comparison of the original and reduced circuits is presented

Table 4.2: *Data for the original and reduced circuits of the two printed striplines model from Fasterix*

model	ports	R	L	C	L_{mutual}	time
original	5	8	8	10	20	0.01
reduced	5	60	60	21	0	0.01

in Table 4.2. Although the number of R,L elements in the reduced is a bit large than in the original one, simulation time in PSTAR stays the same.

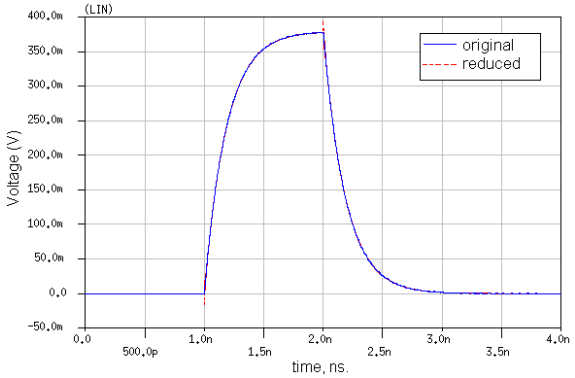

Figure 4.6: *Time domain simulation of the original circuit and reduced passive one.*

4.6.3 Example: lowpass filter model

This interconnect structure has been presented in Section 2.2.3. For the maximum frequency 10 GHz, Fasterix generates a mesh with $\mathcal{N} = 257$ elements and $\mathcal{F} = 452$ common edges between each two neighboring elements. To obtain a reduced circuit, Fasterix chooses 98 super nodes ($\mathcal{N}_1 = 98$). Applying the SNA leads to non-passive reduced circuit, which was detected by the pole-zero analysis in PSTAR.

To guarantee passivity of the reduced circuit, we applied the passive version of the SNA based on modal approximation. The admittance matrix for the full frequency range approximation (4.82) in pole-residue form has $n = 293$ terms, where $H_i, Y_C \in \mathbb{R}^{\mathcal{N}_1 \times \mathcal{N}_1}$. Computed by Fasterix matrix Y_C is positive definite; therefore, we do not apply passivity enforcement for the term sY_C.

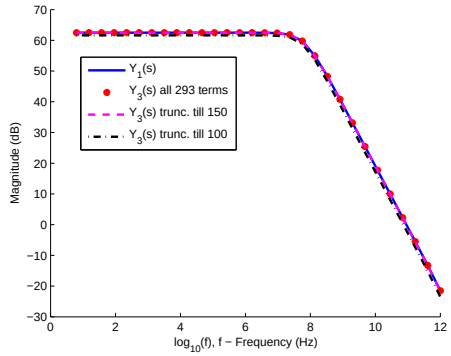

Figure 4.7: *Comparison in frequency domain of $Y_1(s)$ and $Y_3(s)$ for the entry (IN,IN).*

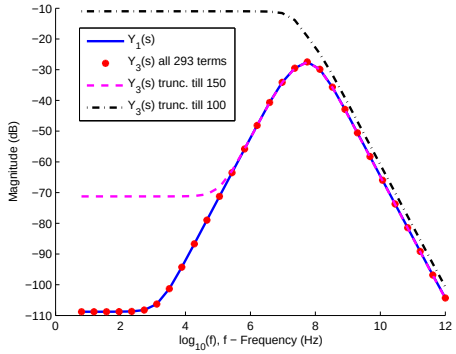

Figure 4.8: *Comparison in frequency domain of $Y_1(s)$ and $Y_3(s)$ for the entry (IN,OUT).*

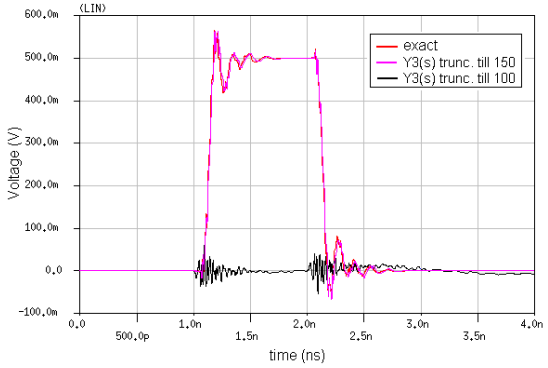

Figure 4.9: *Simulation in time domain of the passive lowpass filter model. Choice of the parameter k influences the accuracy of the reduced circuit.*

Figure 4.7 demonstrates a comparison between $Y_1(s)$, $Y_3(s)$, and approximations of $Y_3(s)$ for the entry (OUT,OUT) in frequency domain. One can observe a good fit between the curves. However, for the entry (IN,OUT), presented in Figure 4.8, it is not the case because DC values are not matched any more, neither for $k = 100$ nor for $k = 150$. This demonstrates a challenge of modal approximation: omitting real poles results in poor approximations. Moreover, truncation till $k = 150$ terms leads to dramatic increase in the amount of circuit elements: obtained reduced circuit will be 26 times larger than the original one. Though the reduced circuit is large, it is passive and transient analysis becomes possible. Figure 4.9 shows simulation in time domain carried out in PSTAR. In both cases ($k = 150$ and $k = 100$) the signal response remains bounded and the pole-zero analysis does not detect any unstable poles.

4.6.4 Summary

Presented above results demonstrate that the passive version of the SNA solves the problem of non-passivity. However, the main drawback is the large complexity of the reduced circuits. Indeed the number of R, L, C elements in the reduced circuit is $O(k\mathcal{N}_1^2)$, where k denotes the number of summands in the admittance matrix Y_{RL} after modal approximation. The number of super nodes, $\mathcal{N}_1$, is chosen by Fasterix and remains fixed for each model. Practical example showed that, to deliver a good approximation, k has to be of the same order as the number of terms in the pole-residue representation of Y_{RL}. As a result reduced circuit may have more R, L, C elements than the original one. (For comparison, reduced circuits by the original SNA has the number of elements of the order $O(\widetilde{k}\mathcal{N}_1^2)$, where $\widetilde{k} \leq 8$.) We conclude that the modified version of the SNA is not recommended to be used for reduction of large models. In the next section we will suggest a different approach which delivers reduced circuits of the same complexity as by the original SNA, while preserving passivity.

4.6.5 Passivity enforcement based on quadratic programming

The idea of the passivity enforcement technique in this section is to enforce passivity directly after applying frequency fitting procedure in the Algorithm 2, [88]. In our case only a small correction needs to be done for the fitted $Y_3(s)$ to become positive real. Indeed, by Theorem 7, the super nodes circuit described by $Y_1(s)$ is passive. The last approximation, $Y_3(s)$, approximates $Y_1(s)$ well till the maximum predefined frequency (Fasterix takes care that the error in (4.79) is small enough). Frequency fitting in (4.86) delivers also fine approximation of $Y_3(s)$ since the last one is described by the real poles (therefore, there are no peaks in frequency domain need to be approximated). Thus the

only disadvantage is that the fitted admittance matrix $Y_3(s)$ with elements of the form

$$Y(s)_{ij} = \sum_{m=1}^{n} \frac{c_m}{s - a_m} + se, \quad i, j = 1, \ldots \mathcal{N}_1, \tag{4.93}$$

might be not positive real. It is a good idea to modify some or all parameters c_m, a_m and e to guarantee positive realness of (4.93) and, therefore, passivity of the circuit. That is exactly can be done by the algorithm in [38]. The idea of the technique is to modify the vector of parameters $\mathbf{x}_0 = \begin{pmatrix} c_m & a_m & e \end{pmatrix}$ with a correction $\Delta\mathbf{x}$ in such a way that the modified rational approximation $Y(s, \widetilde{\mathbf{x}})$ with the vector of parameters $\widetilde{\mathbf{x}} = \mathbf{x}_0 + \Delta\mathbf{x}$ provides a passive system. Ideally we would like to find the correction in an optimal sense, namely minimizing the error between the original solution $Y(s, \mathbf{x}_0) \equiv Y(s)$ and the fitted one $Y(s, \widetilde{\mathbf{x}})$:

$$Y(s) - Y(s, \widetilde{\mathbf{x}}) \to 0. \tag{4.94}$$

Let columns of $Y(s, \widetilde{\mathbf{x}})$ be placed in a vector $\mathbf{y}(s, \widetilde{\mathbf{x}})$ and, similarly, columns of $Y(s)$ are placed in a vector $\mathbf{y}(s)$. In [38] linearization of (4.93) can be written as an incremental relation:

$$\Delta\mathbf{y}(s, \mathbf{x}) = M\Delta\mathbf{x},$$

where M contains the derivatives of (4.93) with respect to c_m. The form of M is shown in Appendix A.2. In other words, changing the vector of parameters, $\mathbf{x}$, changes the vector $\mathbf{y}(s, \mathbf{x})$. Thus, (4.94) can be rewritten as

$$\mathbf{y}(s) - \mathbf{y}(s, \widetilde{\mathbf{x}}) \to 0, \tag{4.95}$$

or, linearizing $\mathbf{y}(s, \widetilde{\mathbf{x}})$ one obtains

$$\mathbf{y}(s) - (\mathbf{y}(s, \mathbf{x}_0) + M\Delta\mathbf{x}) \to 0, \tag{4.96}$$

subject to the passivity constraint that the real part of the eigenvalues of $Y(s, \widetilde{\mathbf{x}})$ has to be positive. The passivity constraint can be also written through the linearization between the eigenvalues, λ, of $Re(Y(s, \widetilde{\mathbf{x}}))$ and the parameter $\Delta\mathbf{x}$ as

$$\Delta\lambda = F\Delta\mathbf{x} \geq -\lambda, \tag{4.97}$$

where F is defined through the normalized eigenvectors of $Re(Y(s, \mathbf{x}))$ which is shown in Appendix A.3. Based on (4.96)–(4.97) a least square solution, $\Delta\mathbf{x}$, can be obtained and added to $\mathbf{x}_0$. The procedure is repeated until all constraints have been satisfied.

We rewrite (4.96) and (4.97) in the standard form:

$$A\Delta\mathbf{x} \to \mathbf{b}, \tag{4.98}$$

$$B\Delta\mathbf{x} \leq \mathbf{c},$$

where

$$A = M, \quad \mathbf{b} = \mathbf{y}(s) - \mathbf{y}(s, \mathbf{x}_0), \quad B = -F, \quad \mathbf{c} = \lambda. \tag{4.99}$$

A least square solution for (4.98)–(4.99) can be computed, for instance, by Quadratic Programming:
minimize

$$\frac{1}{2}\Delta\mathbf{x}^T H \Delta\mathbf{x} - \mathbf{f}^T \Delta\mathbf{x}, \tag{4.100}$$

subject to

$$B\Delta\mathbf{x} \leq \mathbf{c}, \tag{4.101}$$

where

$$H = A^T A = M^T M, \tag{4.102}$$

$$\mathbf{f} = M^T \mathbf{b} = M^T(\mathbf{y}(s) - \mathbf{y}(s, \mathbf{x}_0)),$$

$$B = -\mathbf{F},$$

$$\mathbf{c} = \lambda. \tag{4.103}$$

A code of this passivity enforcement technique [38], called `QPassive`, has been written by B. Gustavsen in Matlab and available in public domain. The code uses the Matlab built-in function *quadprog.m* which comes with the Matlab Optimization Toolbox.

Algorithm 4 shows the main steps of `QPassive`. The parameters of rational approximation are placed in $\mathbf{x}_0$, and a correction $\Delta\mathbf{x}$ is calculated using Quadratic Programming and added to $\mathbf{x}_0$. The procedure is repeated untill all constraints have been satisfied. The frequency samples specified in the input denote frequencies at which passivity violations of $Y(s)$ occur. `Qpassive` calculates the eigenvalues of $Re(Y)$ at these samples and includes violated eigenvalues in the constrained equation. Frequencies which provide passivity violation can be obtained by the frequency sweeping where the eigenvalues of $Re(Y)$ are calculated for a number of frequencies or via Hamiltonian matrix approach [76].

Algorithm 4 Passivity enforcement

INPUT: coefficients of rational function $Y(s)$: a_m, c_m, e and frequency samples
OUTPUT: $\widetilde{a}_m$, $\widetilde{c}_m$, $\widetilde{e}$ which correspond to a passive rational function
1. Enforce e to be positive definite;
2. Calculate H, B, $\mathbf{f}$ and $\mathbf{c}$ in (4.102)–(4.103);
3. Solve (4.100)–(4.101) with respect to $\Delta\mathbf{x}$; using Quadratic Programming;
4. $\mathbf{x}_0 = \mathbf{x}_0 + \Delta\mathbf{x}$;
5. Go to the step 2

To make the SNA passive, we include `QPassive` in Algorithm 3 between the steps 5 and 6. This allows us to keep the structure of the reduced circuit unchanged and to guarantee passivity.

In the next section the SNA with passivity enforcement correction will be analyzed on some Fasterix models. The algorithm was implemented in Matlab and tested on Core 2 Duo 1.6 GHz PC.

4.6.6 Example: two parallel striplines

This model was considered in Section 4.4.5 to show instabilities in the simulations due to non-passivity of the reduced circuit. Figure 4.10 shows simulations in time domain of the original and the reduced passive circuits made in PSTAR. CPU run time for the SNA is 0.06 sec., while passivity enforcement procedure on 100 samples takes 1.06 sec. Figure 4.10 shows that the time response of the synthesized passive reduced circuit matches well the time response of the original one. Note that the number of elements in the reduced circuit is the same as after applying the original SNA, see Table 4.1.

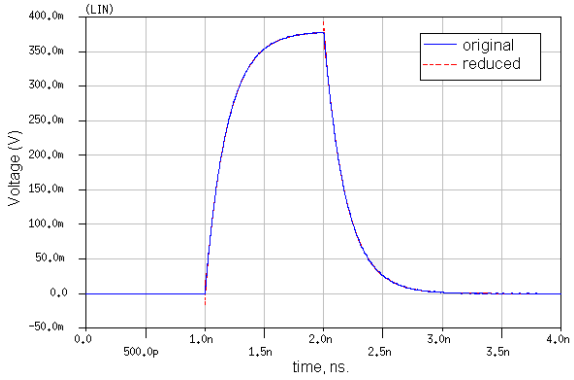

Figure 4.10: *Time domain simulation of the original circuit and reduced passive one.*

4.6.7 Example: lowpass filter

This model has been discussed in Sections 2.2.3 and 4.6.3. Table 4.3 shows comparison of the original and reduced circuits after applying the SNA with the passivity enforcement. It can be seen that the reduced circuit has less elements than the original circuit and the simulation time for the transient analysis in PSTAR was decreased significantly. For the transient analysis, a trapezoidal pulse having rise/fall times of 1 ps and pulse width of 1 ns is applied between the two ports. A 50 Ω resistor R_{in} is connected between the first

Table 4.3: *Data of the original, reduced passive circuits of the lowpass filter model*

Model	ports	R	L	C	L_{mutual}	time
Original	2	452	452	2763	50950	1011.7 s.
Reduced	2	19012	19012	4851	0	47.9 s.

port and the voltage source. A 50 Ω resistor R_{out} is connected between the second port and the ground. The voltage is measured over R_{out}. Figure 4.11 shows that the signal response remains bounded in time domain.

Figure 4.12 demonstrates comparison of the original admittance matrix and the admittance matrix obtained after the SNA with the passivity enforcement for the entry (IN, OUT). One can see a wide range of frequencies with a good agreement, while a visible difference between the curves is the result of the frequency fitting step in the SNA. CPU run time for the SNA is 3.72 sec. and for passivity enforcement on 150 samples is 5.68 sec.

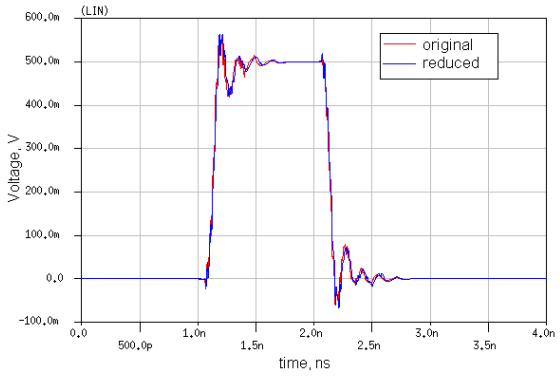

Figure 4.11: *Time response of the original circuit and the reduced passive one.*

4.6.8 Summary

Results above demonstrates that the SNA with incorporated passivity enforcement technique solves the problem of non-passivity. The passivity enforcement technique calculates a correction to the rational approximation of admittance matrix based on linearization and constrained minimization by Quadratic Programming. Due to this correction the modified version of SNA always delivers passive circuits which have exactly the same size as after applying the original SNA. The drawback of the suggested approach is a choice of frequency samples which specify the intervals of passivity violations. They

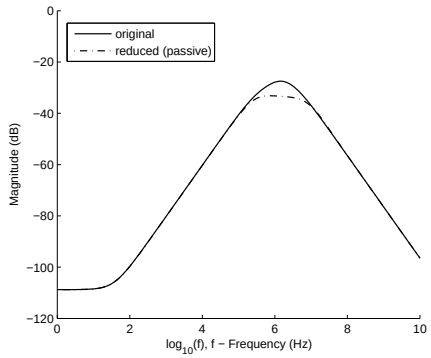

Figure 4.12: *Comparison in frequency domain of admittances of the original and reduced passive circuits for the entry* (IN, OUT).

can be found by the frequency sweeping where the eigenvalues of $Re(Y_3(s))$ are calculated for a number of frequencies or via the Hamiltonian matrix approach [76].

4.7 Concluding remarks

In this chapter an overview of a reduction technique used in the EM tool Fasterix, the SNA, has been presented. For this goal we have shown a derivation of Kirchhoff equations from Maxwell's equations, which constitute a starting point in applying any reduction technique. We also have proved an important property of the resistance matrix used in Kirchhoff equations. The advantage of the SNA is that it delivers reduced models which are applicable for simulation of radiation in Fasterix. In other words, besides the ports, the SNA preserves extra nodes (super nodes) in which the measurements of current will be performed. The SNA delivers stable models, nevertheless we have shown that passivity is not always guaranteed.

To overcome this problem, two approaches for passivity enforcement have been suggested. The first approach, which is based on modal approximation, appears to deliver large passive reduced models especially if original models are large. Therefore, we do not recommend to apply it for reduction of originally large models. The second approach for passivity enforcement is based on calculating a correction to the rational approximation of admittance matrix based on linearization and constrained minimization by Quadratic Programming. The SNA with such correction always delivers passive circuits which have exactly the same size as after applying the algorithm without passivity enforcement. Consequently, the modified SNA delivers reduced models compatible for computation of radiation in Fasterix. Numerical examples have validated the proposed approach.

Chapter 5

Reduction and simplification of resistor networks

The interconnect layouts of chips can be modeled by large resistor networks. To be able to speed up simulations of such large networks, reduction techniques are applied to reduce the size of the networks. For some classes of networks, an existing reduction strategy does not provide sufficient reduction in terms of the number of resistors appearing in the final network. In this chapter we propose an approach for obtaining a further reduction in the amount of resistors. The suggested approach improves sparsity of the conductance matrix by neglecting resistors which do not contribute significantly to the behaviour of the circuit. Explicit error bounds, which give an opportunity to control the errors due to approximation, will be derived. Numerical examples show that the suggested approach appears promising for multi-terminal resistor networks and, in combination with the existing reduction strategy, leads to better reduction.

5.1 Introduction

The interconnect and substrate layouts of chips can be modeled by large resistor networks. Such networks may contain up to millions of resistors, hundreds of thousands of internal nodes and thousands of external nodes and, as a result, simulations of such networks may be very time consuming or not possible. To be able to carry out simulations, model order reduction techniques are used. In [74], an exact reduction technique, *ReduceR*, for resistor networks has been suggested. The approach is based on finding a special order in which internal nodes are eliminated. This allows to maximize spar-

sity of conductance matrix and, therefore, the number of resistors in the reduced model. However, the above approach does not always deliver a good reduction in terms of the number of resistors in the final circuit. For instance, resistor networks with many terminals, extracted by the use of finite element method [29], cannot be reduced efficiently since any elimination of any internal nodes will lead to hardly any reduction in the amount of resistors.

In this chapter we propose an approach for obtaining a reduction in the amount of resistors. The suggested approach improves sparsity of the conductance matrix by neglecting resistors, which do not contribute significantly to the behaviour of the circuit. Further we refer to it as *simplification* of resistor networks. To control the quality of approximation, we derive explicit error bounds and analyze them from different perspectives (sharpness, implementation issues). The suggested approach appears promising for multi-terminal resistor networks and, in combination with *ReduceR*, can improve reduction.

This chapter is organized as follows. In Section 5.2, we summarize the modeling of resistor networks. In Section 5.3 we summarize *ReduceR* and prove a theorem which provides us with a valuable result for resistor networks obtained by the elimination of internal nodes. In Section 5.4, we suggest an idea of simplification and present four criteria, which are used to measure the quality of approximation of the resistor networks. We derive error estimations for these criteria in Sections 5.6 and 5.8, while the algorithms for simplification which allow to delete resistors by groups are suggested in Section 5.5. In Section 5.7, we provide numerical examples and discuss the performance of the suggested algorithms. Relation of simplification with incomplete factorizations is presented in Section 5.9. Section 5.11 concludes.

5.2 Circuit equations and matrices

For an n-port resistor network Ohm's law can be written as [91]:

$$G\mathbf{v} = \mathbf{i}, \tag{5.1}$$

where $G \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite conductance matrix ($G \geq 0$), $\mathbf{v} \in R^n$ are the node voltages, and $\mathbf{i} \in R^n$ are the currents injected in external nodes[1]. Subdividing a set of nodes into external and internal, one can rewrite (5.1) in a block form:

$$\begin{pmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{pmatrix} \begin{pmatrix} \mathbf{v}_e \\ \mathbf{v}_i \end{pmatrix} = \begin{pmatrix} B \\ 0 \end{pmatrix} \mathbf{i}_e, \tag{5.2}$$

[1] ports which define the access to the network

where $\mathbf{v}_e \in \mathbb{R}^{n_e}$ and $\mathbf{v}_i \in \mathbb{R}^{n_i}$ are the voltages at external and internal nodes, respectively $(n = n_e + n_i)$, $\mathbf{i}_e \in \mathbb{R}^{n_e}$ are the currents injected in external nodes, $B \in \{-1, 0, 1\}^{n_e \times n_e}$ is the incidence matrix for the current injections, $G_{11} = G_{11}^T \in \mathbb{R}^{n_e \times n_e}$, $G_{12} \in \mathbb{R}^{n_e \times n_i}$, and $G_{22} = G_{22}^T \in \mathbb{R}^{n_i \times n_i}$.

A k-th current source between terminals a and b with current j leads to contributions $B_{a,k} = 1$, $B_{b,k} = -1$, and $\mathbf{i}_e(k) = j$. If current is only injected in a terminal a, then $B_{a,k} = 1$ and $\mathbf{i}_e(k) = j$. Systems (5.2) must be grounded, i.e., the equation corresponding to the ground node must be removed from the system. This makes the conductance matrix positive definite ($G > 0$).

Deleting a single conductance, g, between two nodes a and b from the conductance matrix G leads to a network with a new conductance matrix $\widetilde{G}$, obtained from G as

$$\widetilde{G} = G - (\mathbf{e}_a - \mathbf{e}_b)g(\mathbf{e}_a - \mathbf{e}_b)^T, \tag{5.3}$$

where $\mathbf{e}_a \in \mathbb{R}^n$ and $\mathbf{e}_b \in \mathbb{R}^n$ are the a-th and b-th unit vectors, respectively. In this case we say that $\widetilde{G}$ is obtained from G by a rank-one correction: rank-one matrix $(\mathbf{e}_a - \mathbf{e}_b)g(\mathbf{e}_a - \mathbf{e}_b)^T$ is a stamp corresponding to the conductance g. Introducing the notation

$$(\mathbf{e}_a - \mathbf{e}_b) = \mathbf{m}, \tag{5.4}$$

the product $\mathbf{m}g\mathbf{m}^T$ is called a *branch-oriented modification* [4]. Thus G can be written as a sum of rank-one corrections:

$$G = \sum_{i=1}^{N} \mathbf{m}_i g_i \mathbf{m}_i^T. \tag{5.5}$$

When n_1 resistors are deleted simultaneously, a new conductance matrix, $\widetilde{G}$, is obtained from G as

$$\widetilde{G} = G - M\widehat{G}M^T, \tag{5.6}$$

where $\widehat{G}$ is $n_1 \times n_1$ diagonal matrix with conductances g_i on the diagonal, and M is a $n \times n_1$ matrix, which consists of columns $\mathbf{m}_i$, $i = 1, \ldots, n_1$.

5.3 Reduction of resistor networks

In general, networks arising during the extraction of chips contain large resistor subnetworks and nonlinear elements like diodes and transistors. Simulation of such complex networks may be very time consuming or unfeasible. A way to overcome this problem is to use model order reduction techniques for resistor subnetworks which will lead to decreased simulation times of the complex networks.

The existing algorithm, *ReduceR* [74], which we discussed in Section 3.2.2, finds a subset of nodes which after being eliminated leads to a sparse conductance matrix. This is done by the use of three strategies: graph algorithms, a reordering strategy (Approximate Minimum Degree algorithm [7]), and a node elimination strategy. These strategies together guarantee that the reduced network is exact up to round-off errors, i.e., no approximation error is made.

Elimination of nodes in *ReduceR* can be viewed as a Schur complement procedure. For example, eliminating all internal nodes from the network described by (5.2) leads to an equivalent network described as:

$$\underbrace{(G_{11} - G_{12}G_{22}^{-1}G_{12}^{T})}_{G_s}\mathbf{v}_e = B\mathbf{i}_e. \tag{5.7}$$

Suppose, a network with conductance matrix G consists of positive resistors and it is grounded. In the next subsection we will prove that the network with conductance matrix G_s in (5.7), obtained after elimination of internal nodes, consists of positive resistors as well. Positiveness of the resistances in the network is important since it implies that the resistance network is passive.

5.3.1 The Schur complement

If a network consists of positive resistors, then the structure of the conductance matrix G in (5.1) is the following: all diagonal elements are positive and all non-diagonal elements are non-negative. Suppose that one of the nodes in the system is grounded. First we will show in Theorem 8, that after elimination of any set of internal nodes, the property of the new conductance matrix G_s will be preserved. This theorem can be found in [46]. Since the proof of this theorem is not available, we will present it here. Before doing this, we introduce some definitions which can be found, for instance, in [47], [14].

Definition: The set $Z_n \subset \mathbb{R}^{n\times n}$ is defined by

$$Z_n = \left\{ A = [a_{ij}] \in \mathbb{R}^{n\times n} : a_{ij} \leq 0 \text{ if } i \neq j,\ i, j = 1, \dots n \right\}.$$

Definition: A matrix $A \in \mathbb{C}^{n\times n}$ is positive stable if the real part of each eigenvalue of A is positive.

Definition: A matrix $A \in \mathbb{R}^{n\times n}$ is called an M-matrix if $A \in Z_n$ and A is positive stable.

Since the network is grounded, G is positive definite ($G > 0$) and, therefore, G is an

M-matrix. The following Theorem 8 shows that the Schur complement of G is an M-matrix. In the proof we refer to the Theorems 9–12 which can be found in Appendix B.1. For convenience, the notation $A \succ 0$ means that the elements of the matrix A are positive.

Theorem 8. *Let*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \in Z_n,$$

where A_{11} and A_{22} are square blocks. A is an M-matrix if and only if A_{11} and its Schur complement $A_{22} - A_{21}A_{11}^{-1}A_{12}$ are M-matrices.

Proof. Let A be an M-matrix. We need to show that A_{11} and $A_{22} - A_{21}A_{11}^{-1}A_{12}$ are M-matrices. Since A_{11} is a principal submatrix of A, A_{11} is an M-matrix according to Theorem 11. Applying the block inversion formulas one obtains, (see [46], p.18):

$$A^{-1} = \begin{pmatrix} \left(A_{11} - A_{12}A_{22}^{-1}A_{21}\right)^{-1} & A_{11}^{-1}A_{12}\left(A_{21}A_{11}^{-1}A_{12} - A_{22}\right)^{-1} \\ \left(A_{21}A_{11}^{-1}A_{12} - A_{22}\right)^{-1}A_{21}A_{11}^{-1} & \left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1} \end{pmatrix}. \tag{5.8}$$

To show that $A_{22} - A_{21}A_{11}^{-1}A_{12}$ is an M-matrix, first we note that $A^{-1} \succ 0$, by Theorem 12. Therefore

$$\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1} \succ 0,$$

as a principal submatrix of A^{-1}. Thus, to show that $A_{22} - A_{21}A_{11}^{-1}A_{12}$ is an M-matrix, the only thing left is to show that $A_{22} - A_{21}A_{11}^{-1}A_{12} \in Z_n$. To this extend, we first notice that

$$A_{21}A_{11}^{-1}A_{12} \succ 0,$$

because $A_{12} \prec 0$, $A_{21} \prec 0$ and $A_{11} \succ 0$. Further $A_{22} \in Z_n$ can be presented as $A_{22} = \alpha I - P$ with $\alpha \in \mathbb{R}$ and $P \succ 0$. Therefore

$$A_{22} - A_{21}A_{11}^{-1}A_{12} = \alpha I - P - A_{21}A_{11}^{-1}A_{12} = \alpha I - \left(P + A_{21}A_{11}^{-1}A_{12}\right) = \alpha I - \widetilde{P}, \tag{5.9}$$

where $\widetilde{P} \succ 0$. Thus $\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right) \in Z_n$.

Conversely, let A_{11} and $A_{22} - A_{21}A_{11}^{-1}A_{12}$ be M-matrices. We need to show that A is an M-matrix. In particular since $A \in Z_n$, it is enough to show that $A^{-1} \succ 0$. An alternative representation for the block $\left(A^{-1}\right)_{11}$ is

$$\left(A^{-1}\right)_{11} = A_{11}^{-1} + A_{11}^{-1}A_{12}\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1}A_{21}A_{11}^{-1} \succ 0. \tag{5.10}$$

We also note that

$$\left(A^{-1}\right)_{22} = \left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)^{-1} \succ 0, \tag{5.11}$$

because $\left(A_{22} - A_{21}A_{11}^{-1}A_{12}\right)$ is an M-matrix. Taking into account (5.10), (5.11) and the fact that $A_{21} \prec 0$ and $A_{12} \prec 0$, it follows that $\left(A^{-1}\right)_{12} \succ 0$ and $\left(A^{-1}\right)_{21} \succ 0$. Therefore, $A^{-1} \succ 0$. □

The above Theorem provides us with valuable information. As we mentioned above if the network with positive resistors is grounded, then G is an M-matrix. Consequently, due to Theorem 8, any Schur complement, G_s, of G is also M-matrix and, therefore, positive stable (in our case, positive definite since G_s is Hermitian). One of the necessary conditions of Hermitian positive definite matrix is that the elements with largest modulus lay on the main diagonals, i.e., $|G_{s,ii}| \geq \sum_{j=1}^{n}(|G_{s,ij}|)$, $i \neq j$. This is enough to conclude that G_s can be synthesized with positive resistors.

5.3.2 Challenge in exact reduction of resistor networks

Unfortunately, the algorithm for reduction of resistor networks, *ReduceR* [74], does not always achieve a great reduction in terms of resistors and internal nodes. For example, networks which come from substrate extraction based on the finite element method usually have a specific quadrilateral structure with large and sparse conductance matrices [29], [80]. The exact reduction of such networks with many terminals is challenging: full or partial elimination of internal nodes may not lead to efficient reduction. The following small example demonstrates the problem.

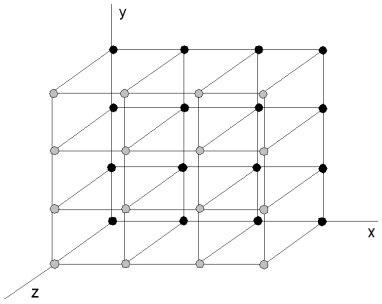

Figure 5.1: *Network with 16 external nodes (black dots), 16 internal nodes and 64 resistors (edges of the cubes).*

Example

The network presented in Figure 5.1 consists of 32 nodes (16 internal nodes, 16 external nodes) and 64 resistors, which correspond to edges of the cubes. Elimination of all internal nodes is not an option here because the reduced network will give a dense matrix with 120 resistors. Reduction by *ReduceR* does not help much: the reduced network has 15 internal nodes, 16 external nodes and the same 64 resistors. In fact the best exact reduction for this network is quite poor: 12 internal nodes, 16 external nodes and 64 resistors. Note that the original network has 8 nodes (in the corners) with degree 3, 16 nodes with degree 4, and 8 nodes with degree 5. At this point, elimination of any internal node in the corners does not decrease the number of resistors (edges) and elimination of any non-corner internal node will only increase the number of resistors. Consequently, one cannot reduce the network further.

This example demonstrates that *ReduceR* does not always deliver sufficient *exact* reduction in the amount of resistors. In the next section we will suggest an approach which improves reduction in the amount of resistors together with *ReduceR*. The new approach is aimed to deliver *approximate* reduced resistor networks, but approximation error is supposed to be under control.

5.4 Simplification of resistor networks

The reduction technique (*ReduceR*) for resistor networks described above is exact up to roundoff errors. As we have seen, it does not always deliver a good reduction for the networks, which come from the substrate extraction based on finite element method and have large amount of terminals.

To deal with such cases, we suggest a new approach, which we will further call simplification. The idea of simplification is to neglect some resistances which do not affect significantly the behaviour of the network. This approach does not deliver exact reduction as in case of *ReduceR*; however, the error of reduction is supposed to be controlled explicitly. The goal of simplification is to improve sparsity of the conductance matrix before or after reduction. Thus, if the number of terminals is large and exact reduction is poor, simplification can be a good alternative to exact reduction. The drawback of simplification is that success will depend on the value of resistances in the network and a given tolerance for approximation.

5.4.1 Error control

Before going further we need to choose an error which we want to control under a certain tolerance. Below we present a few alternatives. The first possibility is to consider a relative error of voltages at *all* nodes:

$$Err_v := \frac{||\mathbf{v} - \widetilde{\mathbf{v}}||}{||\mathbf{v}||} = \frac{||G^{-1}\mathbf{i} - \widetilde{G}^{-1}\mathbf{i}||}{||G_s^{-1}\mathbf{i}||} < \epsilon, \tag{5.12}$$

i.e., for a given ϵ and G one has to find a simplified $\widetilde{G}$ such that (5.12) holds true. It is supposed that $\widetilde{G}$ is obtained from G by neglecting certain entries. Since in practice the current, $\mathbf{i}$ is unknown in advance, computing (5.12) requires the knowledge of an error bound (estimation) which must be independent of $\mathbf{i}$. In Section 5.6.1 we will derive an error estimation based on the condition number of G. It will be shown that such estimation is not sharp and does not deliver significant improvements in sparsity neither for original nor for reduced resistor networks. This topic we also discussed in the publications [87] [86]. Then, in Section 5.6.2, we will derive a different estimation for (5.12) which is sharper that the estimation based on the condition number and relatively cheap to compute. The sharper the estimation, the more resistors it allows to delete.

Another possibility is to consider an error similar to (5.12), which includes voltages only at the external nodes:

$$Err_{ve} := \frac{||\mathbf{v}_e - \widetilde{\mathbf{v}}_e||}{||\mathbf{v}_e||} = \frac{||G_s^{-1}\mathbf{i}_n - \widetilde{G}_s^{-1}\mathbf{i}_n||}{||G_s^{-1}\mathbf{i}_n||} < \epsilon, \tag{5.13}$$

where $\mathbf{v}_e$ and $\widetilde{\mathbf{v}}_e$ are the vectors of voltages at the external nodes before and after simplification, $\mathbf{i}_n$ denotes the vector of currents injected into external nodes, ϵ is a user-defined tolerance, G_s and $\widetilde{G}_s$ are the conductance matrices before and after simplification obtained after elimination of all internal nodes. The error (5.13) is more attractive than the error (5.12) since in reality only external nodes are accessible for measurements. Therefore, there is no need to consider internal nodes as it is done in (5.12). Then, in Section 5.8 we will derive a possible estimation for such error; however, it has some inherent difficulties in application to large networks. Nevertheless, we provide some analytical investigations in this direction.

The second possibility is to consider an error which is based on neglecting resistors that do not affect much all path resistances, i.e.,

$$Err_p := \left| \frac{R_{ij} - \widetilde{R}_{ij}}{R_{ij}} \right| < \delta, \quad i, j = 1, \ldots, n_e, \tag{5.14}$$

where R_{ij} is the path resistance between the external nodes i and j in the original net-

work, $\widetilde{R}_{ij}$ is the path resistance of the simplified network, and δ is a given tolerance. From a practical point of view, such choice of error is useful since it helps to control the condition (b) (Section 3.2.2) used for exact reduction of resistor networks. If n is large, then the direct computation of (5.14) is expensive ($O(n^3)$) for each deleted resistor (or group of resistors). Therefore, there is a need for the error bound which accurately enough estimates (5.14) and has less computational cost than the direct computation.

The third possibility is to consider the relative error between total path resistances. Given a tolerance δ, the goal is to delete resistors in the network such that

$$Err_{tp} := \frac{|R_{tot} - \widetilde{R}_{tot}|}{|R_{tot}|} < \delta, \tag{5.15}$$

where R_{tot} is the total path resistance of the original network (G) and $\widetilde{R}_{tot}$ is the total path resistance of the simplified network ($\widetilde{G}$). The total path resistance is defined as [5]:

$$R_{tot} = \sum_{i<j} R_{ij} = n \sum_{i=1}^{n-1} \frac{1}{\lambda_i}, \tag{5.16}$$

where R_{ij} is the path resistance between nodes i and j, and $\lambda_1 \geq \cdots \geq \lambda_n = 0$ are the eigenvalues of G. Note, that close values of total path resistances do not imply that the networks have similar behaviour. However, if the networks have similar behaviour, then the corresponding total path resistances are similar. Since in our case the simplified network is obtained from the original one by deleting some resistors, we can expect (5.15) to be a measure that indicates, how well the reduced network approximates the original one. In Section 5.6.4 we show this fact via a numerical experiment, and indeed, the smaller δ in (5.15), the better the approximation to the original network. Simplification based on (5.15), of course, requires an efficient estimation with less computational cost than the direct computation of Err_{tp}. Derivation of such estimation will be done in Section 5.6.4.

5.4.2 Problem formulation

Similarly to the problem of reduction of large resistor networks presented in Section 3.2.2, we define the problem of simplification of large resistor networks as follows: given a resistor network described by (5.1), find a reduced network that:

(a) has the same terminals,

(b) has $\hat{N} \ll N$ resistors,

(c) is realizable as a netlist,

(d) has the same internal nodes,

(e) for a given tolerance δ, one of the conditions (5.12)–(5.15) holds true.

Thus the difference between simplification and exact reduction of resistor networks is that after simplification the number of internal nodes always stays the same, while the number of resistors may decrease, and the path resistances may differ from the original path resistances. Another possibility, that could be a subject for further research, is that simplification can be obtained by decreasing the number of nodes in the network. For example, if a resistor between two nodes is relatively small, then current goes through such resistor without obstruction. As a result, two nodes can be considered as one node.

The idea of simplification (neglecting of resistors) is not new, see, for instance [56,74,95]; however, it has not been deeply developed. In [56], a simple criterion to simplify resistor networks has been suggested. The criterion is based on a physical intuition that larger resistors do not affect the behaviour of the network and, therefore, can be removed from the network. According to the criterion, a resistor between nodes i and j is removed if

$$\frac{|G_{ij}|}{|G_{ii}|} < tol, \quad \text{and} \quad \frac{|G_{ij}|}{|G_{jj}|} < tol, \tag{5.17}$$

where tol is a user defined tolerance. The main disadvantage of the criterion is that none of the conditions (5.12)–(5.15) are controlled explicitly and, therefore, the accuracy of the simplified network is not guaranteed.

We will consider simplification and reduction by *ReduceR* as totally independent and complementary procedures. We will distinguish different strategies: 1) simplification, 2) reduction, 3) simplification and then reduction, 4) reduction and then simplification. One may expect that simplification before reduction may improve further reduction because neglecting some resistors in the original network changes network's topology. Note that the simplification procedure may disconnect the network or even lead to a singular conductance matrix; therefore, an extra care should be taken to prevent such cases. In this chapter we create a framework: when to simplify (before or after reduction) and how to simplify networks according to the criteria (5.12)–(5.15) in an efficient way.

Example

One can think that neglecting small entries of conductance matrix by the heuristic approach (5.17) can be sufficient to obtain a network of desired accuracy. However, our further experiment will show that neglecting small entries of a large resistor network does not lead to accurate results. To demonstrate it, we are going to drop off-diagonal

Table 5.1: *Results of simplification by dropping entries G_{ij} such that $G_{ij} < G_{ii} * tol$ for $i \neq j$ in comparison with the original network*

	original	$tol = 10^{-5}$	$tol = 10^{-4}$
#resistors	110413	109143	70872
CPU time	-	60 s.	121 s.
max $\frac{\lvert R-\widetilde{R}\rvert}{\lvert R\rvert}$	-	0.08	0.41

entries of the conductance matrix that are smaller than 0.0001% and 0.00001% of the corresponding diagonal terms. Table 5.1 demonstrates that in the first case, reduction of resistors was not sufficient, while in the second case, the maximum relative error between path resistances was very large (41%).

5.4.3 Deleting a single resistor

In this Section we will show how to compute (5.12) and (5.13) efficiently when a single resistor has been deleted from the network with conductance matrix G resulting to the conductance matrix $\widetilde{G}$.

First we consider the system $G\mathbf{v} = \mathbf{i}$ and evaluate the costs to solve it for $\mathbf{v}$. Matrix G is symmetric positive definite. Computing the Cholesky factorization $G = LL^T$ is feasible since G is sparse and allows for effective column reordering (with e.g., AMD [7]), thus the factor L is also sparse. It is well-known that for matrices which arise in circuit simulations, the complexity for computing the Cholesky factorization is typically $O(n^\alpha)$, where $1 < \alpha \leq 2$ [70]. Solving the resulting triangular system for a single right-hand-side vector by forward- and back substitutions requires about n^2 multiplications and similar number of additions (in case of dense factor L). Thus the total cost to solve a linear system is the sum of factorization cost and cost for solving the triangular system.

After deleting a single resistor, a new conductance matrix, $\widetilde{G}$, can be obtained from G by the rank-one modification (because the outer product matrix $\mathbf{m}g\mathbf{m}^T = \widetilde{\mathbf{m}}\widetilde{\mathbf{m}}^T$, where $\widetilde{\mathbf{m}} = \sqrt{g}\mathbf{m}$, has rank one, i.e., only one linearly independent row and column, and any rank-one matrix can be expressed as an outer product of two vectors). This is an important fact since to solve $\widetilde{G}\widetilde{\mathbf{v}} = \mathbf{i}$ for $\widetilde{\mathbf{v}}$, a new factorization can be avoided.

Let $A \in \mathbb{R}^{n\times n}$ and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. The Sherman-Morrison formula [40]

$$\left(A - \mathbf{u}\mathbf{v}^T\right)^{-1} = A^{-1} + cA^{-1}\mathbf{u}\mathbf{v}^T A^{-1}, \tag{5.18}$$

where

$$c = 1/(1 - \mathbf{v}^T A^{-1}\mathbf{u}),$$

gives the inverse of a matrix resulting from a rank-one modification of a matrix whose inverse is already known. If A is dense, evaluation of this formula requires $O(n^2)$ work rather than the $O(n^3)$ work that would be required to invert the modified matrix from scratch.

If we have the Cholesky factorization of the original matrix G, then the solution $\widetilde{\mathbf{v}}$ of the modified system $\widetilde{G}\widetilde{\mathbf{v}} = \mathbf{i}$, where $\widetilde{G} = G - \widetilde{\mathbf{m}}\widetilde{\mathbf{m}}^T$, can be obtained using the Algorithm 5, which is based on the algorithm presented in [40]. This algorithm involves solving four triangular systems and computing the inner product of vectors. Thus, the total cost to solve $\widetilde{G}\widetilde{\mathbf{v}} = \mathbf{i}$ for $\widetilde{\mathbf{v}}$ is approximately of the order $5O(n^2)$ instead of $O(n^3)$ that is required to solve the system $\widetilde{G}\widetilde{\mathbf{v}} = \mathbf{i}$ from scratch.

Algorithm 5 Solving a linear system $\left(G - \widetilde{\mathbf{m}}\widetilde{\mathbf{m}}^T\right)\widetilde{\mathbf{v}} = \mathbf{i}$ for $\widetilde{\mathbf{v}}$

INPUT: Cholesky factor L of matrix G, $\widetilde{\mathbf{m}}$, $\mathbf{i}$
OUTPUT: $\widetilde{\mathbf{v}}$
1. Solve $L\mathbf{z} = \mathbf{i}$ for $\mathbf{z}$, so that $\mathbf{z} = L^{-1}\mathbf{i}$;
2. Solve $L^T\widetilde{\mathbf{z}} = \mathbf{z}$ for $\widetilde{\mathbf{z}}$, so that $\widetilde{\mathbf{z}} = L^{-T}\mathbf{z}$;
3. Solve $L\mathbf{y} = \widetilde{\mathbf{m}}$ for $\mathbf{y}$, so that $\mathbf{y} = L^{-1}\widetilde{\mathbf{m}}$;
4. Solve $L^T\widetilde{\mathbf{y}} = \mathbf{y}$ for $\widetilde{\mathbf{y}}$, so that $\widetilde{\mathbf{y}} = L^{-T}\mathbf{y}$;
5. Compute $\widetilde{\mathbf{v}} = \widetilde{\mathbf{z}} + (\widetilde{\mathbf{y}}\widetilde{\mathbf{y}}^T\mathbf{i}_n)/(1 - \widetilde{\mathbf{m}}^T\widetilde{\mathbf{y}})$.

We will rewrite the Sherman-Morrison formula in a way, suitable for modification of resistor networks, with a modification term in the form $M\widehat{G}M^T$ as in (5.6):

$$\left(G - M\widehat{G}M^T\right)^{-1} = G^{-1} - G^{-1}MCM^TG^{-1}, \tag{5.19}$$

The formula for matrix C can be rewritten in many different ways, all mathematically but not computationally equivalent to each other. Three of the most useful arrangements are [6]:

$$C = (Z - \widehat{G}^{-1})^{-1} = \tag{5.20}$$

$$-(I + \widehat{G}Z)^{-1}\widehat{G} = \tag{5.21}$$

$$-Z^{-1}(Z^{-1} - \widehat{G})^{-1}\widehat{G}, \tag{5.22}$$

where $Z = M^TG^{-1}M$ and I is the identity matrix. For a single branch-oriented modification between nodes i and k, C is a scalar [6]:

$$C = (-\frac{1}{\widehat{G}} + K_{ii} + K_{kk} - K_{ik} - K_{ki})^{-1},$$

where $K = G^{-1}$.

To compute (5.13), one first has to obtain G_s and $\widetilde{G}_s$ and then to solve the systems $G_s\mathbf{v}_e = \mathbf{i}_n$ and $\widetilde{G}_s\widetilde{\mathbf{v}}_e = \mathbf{i}_n$. Note that G_s and $\widetilde{G}_s$ will be most lightly dense.

The Schur complement $G_s = G_{11} - G_{12}G_{22}^{-1}G_{12}^T$ can be computed efficiently using the Cholesky factorization:

$$G_s = G_{11} - QQ^T, \tag{5.23}$$

where $Q = L^{-1}G_{12}^T$ and $G_{22} = LL^T$. Similarly one can compute the Schur complement $\widetilde{G}_s = \widetilde{G}_{11} - \widetilde{G}_{12}\widetilde{G}_{22}^{-1}\widetilde{G}_{12}^T$ and then to solve the above linear systems for $\mathbf{v}_e$ and $\widetilde{\mathbf{v}}_e$.

Let a single resistor be deleted from the block G_{22} of the conductance matrix G. To compute $\widetilde{G}_s$ efficiently, one can apply a procedure know as *downdating* [23] in order to downdate the Cholesky factorization $G_{22} = LL^T$:

$$\widetilde{L}\widetilde{L}^T = LL^T - \widetilde{\mathbf{m}}\widetilde{\mathbf{m}}^T,$$

where $\widetilde{L}\widetilde{L}^T$ is the downdated Cholesky factorization of $\widetilde{G}_{22}$, and vector $\widetilde{\mathbf{m}}$ has rank one. Downdating the Cholesky factorization is cheaper than recomputing the Cholesky factorization from scratch ($O(n^2)$ instead of $O(n^3)$). In Section 5.8 we will use this approach to delete resistors from a resistor network with an arbitrary input current $\mathbf{i}_n$.

Deleting resistors one by one and then computing (5.12) or (5.13) may be an expensive procedure if the number of deleted resistors is large. Therefore, we need error estimations which bound (5.12) and (5.13). We are particularly concerned with estimations which are computationally inexpensive and allow us to estimate the errors after deleting a group of resistors. In the next section we discuss a few strategies to delete resistors by groups, while in Sections 5.6 and 5.8 we are concerned with deriving the error estimations for (5.12)–(5.15).

5.5 Deleting resistors by groups

For convenience we will use Err to denote a generic error, i.e., Err_v, Err_{ve}, or Err_p. Now the question is as follows. Which resistors should we delete from G and in which order should they be deleted to obtain $\widetilde{G}$, such that $Err < \delta$, where δ is a given tolerance?

Before answering this question we give some physical intuition. The larger resistor, the less current flows through it. Thus, if a resistor has a very large value, then almost no current goes through it, therefore, such resistor can be neglected in some sense. This principle can be used in our case. Figure 5.2 shows computed errors Err_v and Err_p after deleting each resistor individually. In general, the larger deleted resistor, the smaller

errors Err_v and Err_p become.

Further, before deleting resistors we suggest to sort them in decreasing order. After that a procedure for deleting resistors can be performed. The simplest (but not the optimal)

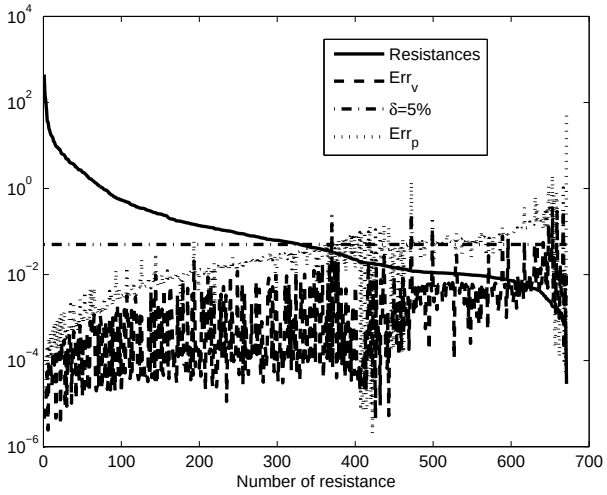

Figure 5.2: *Computed errors Err_p and Err_v after deleting each resistor individually. A visual tendency: the larger resistor value, the smaller computed errors.*

case is to delete resistors one-by-one and then to check whether $Err \leq \delta$. Since deleting resistors one by one is not a fast option, we would like to delete resistors by groups. For this we suggest the following alternatives:

(1) *Golden search 1*. Choose k, which is less than the number of all resistors. (For instance, k can be chosen as 10% of the whole amount of resistors.) Try to delete at once k resistors and check whether the network becomes disconnected. If the network is still connected, then compute Err. If $Err < \delta$, then try to delete the next $2k$ resistors, otherwise try to delete $k/2$ resistors. If the network is disconnected, then try to delete $k/2$ resistors. If $k = 1$ and the network is disconnected, then skip the first resistor and continue the procedure from the beginning. As soon as $Err > \delta$ and $k = 1$ appears T_{max} times (e.g., in the further examples $T_{max} = 5$), the procedure is stopped.

(2) *Golden search 2*. It is the same procedure as "Golden search 1" but with a different stop criterion. Simplification has to be stopped immediately after $Err > \delta$ *and* $k = 1$ (parameter $T_{max} = 1$).

The *Golden search 2* is faster than *Golden search 1*, though *Golden search 2* usually allows

to delete fewer resistors. To make a choice between these approaches, one needs to find a balance between the time and the amount of deleted resistors.

The above algorithms are some options for selecting resistors that are candidates to be eliminated. The set of resistors to be eliminated can be adapted for deleting as many resistors as possible, while keeping the error under control.

5.6 Error estimations

In this section we will derive estimations for the errors $Err_v = \frac{||\mathbf{v}-\widetilde{\mathbf{v}}||}{||\widetilde{\mathbf{v}}||}$, $Err_p = \frac{|R_{ij}-\widetilde{R}_{ij}|}{|R_{ij}|}$, and $Err_{tp} = \frac{|R_{tot}-\widetilde{R}_{tot}|}{|R_{tot}|}$.

5.6.1 Error estimation for $\frac{||\mathbf{v}-\widetilde{\mathbf{v}}||}{||\widetilde{\mathbf{v}}||}$ (first version)

In this section we suggest a simple error estimation for (5.12) which is based on the approach in [95]. Results in this section will demonstrate that obtained error estimation is not very sharp since it is based on computing the condition number of G. Let some resistors be deleted from the resistor network with the conductance matrix G and let $\widetilde{G}$ be the conductance matrix after deleting resistors. The criteria for simplification can be described as

$$||\Delta G|| \leq tol \cdot ||G||, \tag{5.24}$$

where $\Delta G = \widetilde{G} - G$ and tol is a tolerance. Thus, the solution of the network after simplification, $\widetilde{\mathbf{v}}$, satisfies

$$\begin{aligned}\widetilde{\mathbf{v}} = \widetilde{G}^{-1}\mathbf{i} = (G+\Delta G)^{-1}\mathbf{i} = (G+\Delta G)^{-1}(G+\Delta G-\Delta G)G^{-1}\mathbf{i}\\ = (I-(G+\Delta G)^{-1}\Delta G)G^{-1}\mathbf{i} = (I-(G+\Delta G)^{-1}\Delta G)\mathbf{v}.\end{aligned} \tag{5.25}$$

From (5.25) it follows that the error, induced by simplification, can be expressed as

$$\frac{||\mathbf{v}-\widetilde{\mathbf{v}}||}{||\mathbf{v}||} \leq ||(G+\Delta G)^{-1}\Delta G|| \leq ||(G+\Delta G)^{-1}|| \cdot ||\Delta G||. \tag{5.26}$$

Taking into account (5.24), we obtain

$$Err_v = \frac{||\mathbf{v}-\widetilde{\mathbf{v}}||}{||\mathbf{v}||} \leq ||(G+\Delta G)^{-1}|| \cdot ||\Delta G|| \cdot \frac{||G||}{||G||}$$

Table 5.2: *Number of resistors in the original networks, after reduction by* ReduceR *(R), and after simplification (S) by* Err_{vc}. *Parameters are* $\delta = 5\%$, $Tmax = 5$

	I	II	III	IV
#resistors originally	23222	2476	70006	1936
#resistors after (S)	0	0	10 (2.3 s.)	0
#resistors after (R)	3315	702	1485	1397
#resistors after (R+S)	922 (0.2 s.)	61 (0.1 s.)	0	12 (1.2 s.)

$$\approx \kappa(G)\frac{||\Delta G||}{||G||} \leq \kappa(G)\cdot tol \equiv Err_{vc}, \tag{5.27}$$

where $\kappa(G)$ is the condition-number of G. Thus Err_{vc} is a bound of the relative error Err_v. Inequality (5.27) demonstrates the controllability of the error induced by the simplification of G.

Based on (5.24) and (5.27), a cheap and fast simplification procedure can be defined. For a given G and tolerance δ, one has to compute $\kappa(G)$ and to choose the parameter tol such that it is less than $\delta/\kappa(G)$. After deleting a resistor (a group of resistors), the condition (5.24) is checked. If it holds true, then the deleted resistor is confirmed, and the next resistor is tried to be deleted. Otherwise, the deleted resistor is not confirmed, and the next resistor is considered. This simplification procedure can be incorporated with a strategy for deleting groups of resistors which is presented in Section 5.5.

In practice, however, the condition number, $k(G)$, (after the network is grounded) is usually in the range from 10^5 till 10^8; therefore, for the required accuracy, e.g., $\epsilon = 5\%$, parameter tol must be small. As a result, condition (5.24) may become too strict for deleting a big amount of resistors, which makes the estimation not sharp enough. We remind that grounding of the network is required here in order to prevent G from being singular. If the network is not initially grounded, one can temporally ground an arbitrary external node and then perform simplification according to (5.24) and (5.27). After that, the deleted external node is added to the network.

Examples

Table 5.2 shows results of simplification by using the error estimation (5.27). Simplification applied to the original networks I, II and IV does not lead to reduction of resistors, and for the network III a modest reduction has been obtained. Simplification applied after *ReduceR* leads to better results then simplification applied before reduction. Following the reasoning in Section 5.6.1, the results demonstrate that the error estimation, Err_{vc},

does not generally lead to extensive reduction.

5.6.2 Error estimation for $\frac{||\mathbf{v}-\widetilde{\mathbf{v}}||}{||\mathbf{v}||}$ (second version)

In this section we suggest a new error estimation for Err_v, which shows to be sharper than the error estimation based on the condition number (5.27). To derive it we consider:

$$\max_{i\in\mathcal{A}} \frac{||\mathbf{v}-\widetilde{\mathbf{v}}||_2}{||\mathbf{v}||_2} = \max_{i\in\mathcal{A}} \frac{||G^{-1}\mathbf{i}-\widetilde{G}^{-1}\mathbf{i}||_2}{||G^{-1}\mathbf{i}||_2} = \max_{\mathbf{f}} \frac{||(I-\widetilde{G}^{-1}G)\mathbf{f}||_2}{||\mathbf{f}||_2}$$

$$\leq \max_{f\in\mathbb{R}^n,\mathbf{f}\neq 0} \frac{||(I-\widetilde{G}^{-1}G)\mathbf{f}||_2}{||\mathbf{f}||_2} = \sigma_1,$$

where $\mathbf{f} = G^{-1}\mathbf{i}_n$, σ_1 is the maximum singular value of $(I-\widetilde{G}^{-1}G)$, and

$$\mathcal{A} = \left\{\mathbf{i}\in\mathbb{R}^n | (\mathbf{i})_k = 1, (\mathbf{i})_l = -1, k,l \in \{1,\ldots,n_e\}, k\neq l\right\}.$$

Thus we obtain

$$Err_v = \frac{||\mathbf{v}-\widetilde{\mathbf{v}}||_2}{||\mathbf{v}||_2} \leq \max_{\mathbf{f}\in\mathbb{R}^n,\mathbf{f}\neq 0} \frac{||(I-\widetilde{G}^{-1}G)\mathbf{f}||_2}{||\mathbf{f}||_2} \tag{5.28}$$

$$= \left(\lambda_{max}\left((I-\widetilde{G}^{-1}G)^T(I-\widetilde{G}^{-1}G)\right)\right)^{\frac{1}{2}} = \sigma_1 = Err_{vs}, \tag{5.29}$$

where λ_{max} denotes the largest eigenvalue, and Err_{vs} demonstrates estimation for the relative error Err_v. To show that Err_{vs} in (5.29) is sharper than Err_{vc} in (5.27), we note that

$$\frac{||\mathbf{v}-\widetilde{\mathbf{v}}||_2}{||\mathbf{v}||_2} = \frac{||G^{-1}\mathbf{i}-\widetilde{G}^{-1}\mathbf{i}||_2}{||G^{-1}\mathbf{i}||_2} \leq \kappa(G)\frac{||\Delta G||_2}{||G||_2}. \tag{5.30}$$

By taking the maximum over $\mathbf{i}\in\mathbb{R}^n$, $\mathbf{i}\neq 0$ and noting that the right part of (5.30) does not depend on $\mathbf{i}$, we conclude that $Err_{vs} \leq Err_{vc}$.

Computation of the maximum singular value can be performed, for instance, by the implicitly restarted Lanczos bidiagonalization methods [11], Jacobi-Davidson type SVD method [44] [45], or Krylov-Schur method [84], which is used for numerical examples in Section 5.7. We note that within the Krylov-Schur method there is no need to compute $(I-\widetilde{G}^{-1}G)$ directly, one only requires to compute the matrix-vector product of the form $(I-\widetilde{G}^{-1}G)\mathbf{x}$, which is performed by solving one linear system and one matrix-vector product. This requires $\widetilde{G}$ to be nonsingular and, therefore, the network has to be grounded. If the network is not grounded, one can temporarily ground an arbitrary external node and after simplification unground it, i.e., to insert back the corresponding row and

column in G.

We note that the estimation Err_{vs} can be incorporated into the algorithms, presented in Section 5.5, to delete resistors by groups.

5.6.3 Error estimation for $\left|\frac{R_{ij}-\widetilde{R}_{ij}}{R_{ij}}\right|$

The path resistance between two nodes i and j is defined as the ratio of voltage across i and j to the current injected into them. In practice, path resistances from the input of one device to the output of one or more devices are used. Path resistances can be used for example for the analysis of the power dissipation, and in electro static discharge analysis to check whether unintended peak currents are conducted well enough through the resistive protection network to prevent damage to the chip [74].

The path resistance between ports i and j is defined as [30]

$$R_{ij} = (\mathbf{e}_i - \mathbf{e}_j)^T G^{-1} (\mathbf{e}_i - \mathbf{e}_j), \tag{5.31}$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the ith and jth unit vectors, respectively. It is easy to show that the path resistances of the original network, with conductance matrix G, are equal to the path resistances of the network obtained after elimination of (all) internal nodes. Indeed, if G has been split into the blocks as in (5.2) and $\mathbf{e}_t^T = \begin{pmatrix} \widetilde{\mathbf{e}}_t & 0 \end{pmatrix}^T$, then

$$\begin{aligned} R_{ij} = (\mathbf{e}_i - \mathbf{e}_j)^T G^{-1} (\mathbf{e}_i - \mathbf{e}_j) &= (\widetilde{\mathbf{e}}_i - \widetilde{\mathbf{e}}_j)^T (G_{11} - G_{12} G_{22}^{-1} G_{12}^T)^{-1} (\widetilde{\mathbf{e}}_i - \widetilde{\mathbf{e}}_j) \\ &= (\widetilde{\mathbf{e}}_i - \widetilde{\mathbf{e}}_j)^T G_s^{-1} (\widetilde{\mathbf{e}}_i - \widetilde{\mathbf{e}}_j), \end{aligned}$$

which follows based on the inverse of 2×2 block matrix [46]. Note that if a network has only positive resistors, then all path resistances in the network are positive.

We will treat simplification of the network from the point of view that the path resistances of the simplified network should not differ much from the path resistances of the original network. Substituting (5.31) into (5.14) leads to

$$Err_p = \left| \frac{\mathbf{e}_{ij}^T G^{-1} \mathbf{e}_{ij} - \mathbf{e}_{ij}^T \widetilde{G}^{-1} \mathbf{e}_{ij}}{\mathbf{e}_{ij}^T G^{-1} \mathbf{e}_{ij}} \right| < \delta, \tag{5.32}$$

where $\mathbf{e}_{ij} = \mathbf{e}_i - \mathbf{e}_j$, $i, j = 1, \ldots, n_e$, and $i \neq j$.

We note that computing (5.32) directly is not an option, especially, if the number of external nodes, n_e, is large. Therefore, it is a good idea to have an estimation for (5.32),

which is sharp enough and can be easily computed.

A grounded network guarantees that G is positive definite ($G > 0$). If a network is not grounded, one can *temporarily* ground an arbitrary external nodes and after simplification to ungrounded it, i.e., to insert back the corresponding row and column in G. Let L be the Cholesky factor of G, i.e., $G = LL^T$, then Err_p for arbitrary i, j, $(i, j = 1, \ldots, n_e)$ is less or equal than the maximum relative error between all path resistances, i.e.,

$$Err_p \leq \max_{e_{ij} \in \mathcal{A}} \left| \frac{\mathbf{e}_{ij}^T G^{-1} \mathbf{e}_{ij} - \mathbf{e}_{ij}^T \widetilde{G}^{-1} \mathbf{e}_{ij}}{\mathbf{e}_{ij}^T G^{-1} \mathbf{e}_{ij}} \right| = \max_{\mathbf{e}_{ij} \in \mathcal{A}} \left| \frac{\mathbf{e}_{ij}^T (G^{-1} - \widetilde{G}^{-1}) \mathbf{e}_{ij}}{\mathbf{e}_{ij}^T L^{-T} L^{-1} \mathbf{e}_{ij}} \right|, \tag{5.33}$$

where

$$\mathcal{A} = \left\{ \mathbf{e}_{ij} \in \mathbb{R}^n | \mathbf{e}_{ij} = \mathbf{e}_i - \mathbf{e}_j, \quad \mathbf{e}_i = 1, \quad \mathbf{e}_j = -1, \quad i, j \in \{1, \ldots, n_e\}, i \neq j \right\}.$$

Setting up $\mathbf{y} = L^{-1} \mathbf{e}_{ij}$, one can rewrite (5.33) as follows

$$Err_p \leq \max_{\mathbf{e}_{ij} \in \mathcal{A}} \left| \frac{\mathbf{e}_{ij}^T (G^{-1} - \widetilde{G}^{-1}) \mathbf{e}_{ij}}{\mathbf{e}_{ij}^T L^{-T} L^{-1} \mathbf{e}_{ij}} \right| \leq \max_{\mathbf{y} \in \mathbb{R}^n} \left| \frac{\mathbf{y}^T L^T (G^{-1} - \widetilde{G}^{-1}) L \mathbf{y}}{\mathbf{y}^T \mathbf{y}} \right| = \max(|\lambda_1|, |\lambda_n|), \tag{5.34}$$

where λ_1 and λ_n are the largest and the smallest eigenvalues of

$$L^T (G^{-1} - \widetilde{G}^{-1}) L.$$

The advantage of (5.34) is that it is independent of $\mathbf{e}_{ij}$; however, it contains inverse matrices. To make computations efficient, we would like to get rid of the inverse matrices. First, we consider the following eigenvalue problem:

$$L^T (G^{-1} - \widetilde{G}^{-1}) L \mathbf{x} = \lambda \mathbf{x}, \tag{5.35}$$

Multiplying (5.35) on the left side by L^{-T} and setting up $\mathbf{w} = L\mathbf{x}$, one obtains:

$$(G^{-1} - \widetilde{G}^{-1}) \mathbf{w} = \lambda L^{-T} L^{-1} \mathbf{w},$$

or, equivalently,

$$(G^{-1} - \widetilde{G}^{-1}) \mathbf{w} = \lambda G^{-1} \mathbf{w}. \tag{5.36}$$

Multiplying (5.36) from the left by G and setting up $\mathbf{z} = \widetilde{G}^{-1} \mathbf{w}$, (5.36) becomes

$$(\widetilde{G} - G) \mathbf{z} = \lambda \widetilde{G} \mathbf{z}. \tag{5.37}$$

Thus Err_p is approximated as follows:

$$Err_p \leq \max(|\lambda_1|, |\lambda_n|) \equiv Err_{pa}, \tag{5.38}$$

where λ_1 and λ_n are the largest and the smallest eigenvalues of the generalized eigenvalue problem (5.37). Err_{pa} is an error bound of Err_p and requires to compute one largest magnitude eigenvalue. Since not the full spectrum of eigenvalues is required, one can use iterative methods such as the Lanczos method [67], the Arnoldi method [55], or the Jacobi-Davidson method [26] that exploit the sparsity of the system to limit memory and CPU requirements. For numerical examples in Section 5.7 we will use Matlab build-in function *eigs* which uses the Arnoldi method.

Computing the estimations Err_{pa} in (5.37) and Err_{vs} in (5.29) involves calculation of eigenvalues which are related to some extend. Multiplying (5.37) from the left by $\widetilde{G}^{-1}$, the estimation Err_{pa} requires computing the largest magnitude eigenvalue of the matrix $A := (I - \widetilde{G}^{-1}G)$, while Err_{vs} requires computing the maximum eigenvalue of A^TA. In the particular case of a symmetric positive definite matrix A, one obtains $\sqrt{\lambda_{max}(A^TA)} = \lambda_{max}(A)$, or, equivalently,

$$\sigma_{max}(A) = \lambda_{max}(A).$$

Since in our case A is nonsymmetric, the relation between $\sigma_{max}(A)$ and $\lambda_{max}(A)$ is nontrivial. Note that only those resistors can be considered for deleting which do not make $\widetilde{G}$ singular. For example, $\widetilde{G}$ becomes singular when network is disconnected. Therefore, care should be taken that deleted resistors will not disconnect the network. This can be achieved by using a graph algorithm, which computes strongly connected components of an undirected graph, which corresponds to the resistor network [31]. If the number of strongly connected components is larger than 1, then the network is disconnected.

Restriction on G to be positive definite is important because it allows us to use the Cholesky factorization and to keep the matrix pencil $(\widetilde{G} - G, \widetilde{G})$ regular. To demonstrate the last proposition, let resistor network not be grounded, then the matrix pencil $(\widetilde{G} - G, \widetilde{G})$ is not regular, i.e., $\widetilde{G} - G - \gamma\widetilde{G}$ is singular for any $\gamma \in \mathbb{C}$:

$$(\widetilde{G} - G - \lambda\widetilde{G})\mathbf{x} = 0. \tag{5.39}$$

Suppose a resistor has been deleted, i.e., $\widetilde{G}$ has been obtained from G by a rank-one update:

$$\widetilde{G} = G - \mathbf{e}\mathbf{e}^T. \tag{5.40}$$

Substituting (5.40) into (5.39), one obtains

$$(-\lambda G + (\lambda - 1)\mathbf{e}\mathbf{e}^T)\mathbf{x} = 0. \tag{5.41}$$

If the network is not grounded (at least temporarily), then G is singular. Adding a rank-one matrix (which is also singular) will lead to a singular matrix for any λ. Therefore, (5.41) will not have a unique solution. If the network is grounded, G becomes positive definite, thus deleting any conductances which do not disconnect the network, will help

to keep the pencil $(\widetilde{G} - G, \widetilde{G})$ regular.

We note that the estimation (5.38) can be incorporated into the algorithms, presented in Section 5.5, to delete resistors by groups.

5.6.4 Error estimation for $\frac{|\widetilde{R}_{tot} - R_{tot}|}{|R_{tot}|}$

Another way to simplify resistor networks can be based on the relative error of total path resistances. We suppose that a tolerance δ is given, and one has to delete resistors to guarantee

$$Err_{tp} = \frac{|\widetilde{R}_{tot} - R_{tot}|}{|R_{tot}|} \leq \delta, \tag{5.42}$$

where R_{tot} is the total path resistance. The total path resistance is defined as

$$R_{tot} = \sum_{i<j} R_{ij} = 0.5 \cdot \mathbf{1}^T \hat{R} \mathbf{1},$$

where $\hat{R}$ denotes a matrix constructed from the path resistances with zero diagonal. Another representation of the total path resistance is [5](section 3.4)

$$R_{tot} = n \sum_{i=1}^{n-1} \frac{1}{\lambda_i},$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0$ are the eigenvalues of the original conductance matrix G, which is positive semidefinite. If the total path resistances of two resistor networks are equal, it does not imply that the resistor networks are equal. Nevertheless, (5.42) may be a measure of approximation between resistor networks, when one network has been obtained from another one by deleting some resistors.

Let $G \in \mathbb{R}^{n \times n}$ be an original, positive semidefinite conductance matrix with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n = 0,$$

and let $\widetilde{G}$ denote a conductance matrix obtained after simplification with eigenvalues

$$\widetilde{\lambda}_1 \geq \widetilde{\lambda}_2 \geq \cdots \geq \widetilde{\lambda}_n = 0.$$

Thus

$$\widetilde{G} = G + (-\Delta G),$$

where $-\Delta G$ is a negative semidefinite matrix which contains stamps of deleted resistors

and has eigenvalues

$$0 = \epsilon_1 \geq \epsilon_2 \geq \cdots \geq \epsilon_n.$$

Applying Corollary 2 from Appendix B.2, one obtains

$$|\widetilde{\lambda}_i - \lambda_i| \leq \max |\widetilde{\lambda}_i - \lambda_i| \leq \max \{|\epsilon_1|, |\epsilon_n|\}, \quad \text{for} \quad i = 1, \ldots, n. \tag{5.43}$$

Multiplying (5.43) by $\frac{1}{\widetilde{\lambda}_i \lambda_i}$ ($\lambda_i \neq 0$, $\widetilde{\lambda}_i \neq 0$) and summing up from 1 till $n-1$, we have

$$\sum_{i=1}^{n-1} \frac{|\widetilde{\lambda}_i - \lambda_i|}{\widetilde{\lambda}_i \lambda_i} \leq \max \{|\epsilon_1|, |\epsilon_n|\} \sum_{i=1}^{n-1} \frac{1}{\widetilde{\lambda}_i \lambda_i}.$$

The absolute error of total path resistances can be bounded as

$$|\widetilde{R}_{tot} - R_{tot}| = n \left| \sum_{i=1}^{n-1} \frac{1}{\widetilde{\lambda}_i} - \sum_{i=1}^{n-1} \frac{1}{\lambda_i} \right| = n \left| \sum_{i=1}^{n-1} \frac{\lambda_i - \widetilde{\lambda}_i}{\widetilde{\lambda}_i \lambda_i} \right| \leq n \sum_{i=1}^{n-1} \frac{|\lambda_i - \widetilde{\lambda}_i|}{\widetilde{\lambda}_i \lambda_i}$$

$$\leq \max \{|\epsilon_1|, |\epsilon_n|\}\, n \sum_{i=1}^{n-1} \frac{1}{\widetilde{\lambda}_i \lambda_i}.$$

Since

$$\frac{1}{\widetilde{\lambda}_i \lambda_i} \leq \frac{1}{\widetilde{\lambda}_{n-1}} \cdot \frac{1}{\lambda_i}, \quad \text{for } i = 1, \ldots, n-1,$$

we have

$$|\widetilde{R}_{tot} - R_{tot}| \leq \frac{\max \{|\epsilon_1|, |\epsilon_n|\}}{\widetilde{\lambda}_{n-1}} \cdot n \sum_{i=1}^{n-1} \frac{1}{\lambda_i}.$$

Therefore,

$$\frac{|\widetilde{R}_{tot} - R_{tot}|}{|R_{tot}|} \leq \frac{\max \{|\epsilon_1|, |\epsilon_n|\}}{\widetilde{\lambda}_{n-1}}. \tag{5.44}$$

This inequality provides an error bound for the relative error of the total path resistance. Note, that $\epsilon_1 = 0$, thus the error bound only depends on the second smallest eigenvalue of $\widetilde{G}$ and the largest absolute eigenvalue of $-\Delta G$. After deleting each new resistor, the error bound (5.44) has to be recomputed. Using Corollary 1 from Appendix B.2, it follows that

$$\lambda_{n-1} + \epsilon_n \leq \widetilde{\lambda}_{n-1} \leq \lambda_{n-1} + \epsilon_1,$$

or

$$\frac{1}{\lambda_{n-1} + \epsilon_n} \geq \frac{1}{\widetilde{\lambda}_{n-1}} \geq \frac{1}{\lambda_{n-1} + \epsilon_1}. \tag{5.45}$$

Taking into account (5.45) and the fact that $\epsilon_1 = 0$, the relative error Err_{tp} can be boun-

ded as:

$$Err_{tp} = \frac{|\widetilde{R}_{tot} - R_{tot}|}{|R_{tot}|} \leq \frac{\max\{|\epsilon_1|, |\epsilon_n|\}}{\widetilde{\lambda}_{n-1}} \leq \frac{\max\{|\epsilon_1|, |\epsilon_n|\}}{\lambda_{n-1}} \equiv Err_{tpa}. \tag{5.46}$$

The attractive feature is that λ_{n-1} is the second smallest eigenvalue of the original matrix G and does not need to be recomputed after deleting each new resistor. Consequently, only the largest magnitude eigenvalue of $-\Delta G$ has to be recomputed. This makes Err_{tpa} more attractive from computational point of view than the direct computing of Err_{tp}. At this point one can incorporate the error bound Err_{tpa} into the procedures *Golden search 1* or *Golden search 2*, presented in Section 5.5, to delete resistors by groups.

Special case

We will simplify Err_{tpa} in (5.46) in order to delete conductances without recomputing the largest magnitude eigenvalue, $\max\{|\epsilon_1|, |\epsilon_n|\}$. Suppose ΔG is obtained by deleting resistors which do not have common nodes (here we also suppose that deleted resistors do not disconnect the network), i.e., upon permutation, ΔG, has a special structure, e.g.,

$$\Delta G = \begin{pmatrix} g_1 & -g_1 & & & \cdots \\ -g_1 & g_1 & & & \cdots \\ & & g_2 & -g_2 & \\ & & -g_2 & g_2 & \\ \vdots & \vdots & & & \end{pmatrix}.$$

If k resistors have been deleted, the eigenvalues of $-\Delta G$ are $\{0, -2g_1, \ldots, -2g_k\}$, i.e., $\epsilon_1 = 0$ and $\epsilon_n = -2g_k$. Thus

$$\max\{|\epsilon_1|, |\epsilon_n|\} = 2 \max_k g_k.$$

This assumption helps us to derive the desired condition, which allows to delete several resistors (conductances) at once without recomputing $\max\{|\epsilon_1|, |\epsilon_n|\}$ in (5.46). Suppose that a required tolerance δ is given and one has to simplify the network and guarantee (5.42). Based on (5.46), conductances g_i, which satisfy

$$g_i \leq \delta \frac{\lambda_{n-1}}{2}, \tag{5.47}$$

can be deleted from the network at once. We note that under the above assumption on ΔG, the maximum number of resistors which can be deleted is $n/2$, where n is the number of nodes in the network. Success of the simplification will depend not only on the values δ and λ_{n-1} but also on the values of conductances g_i. In practice it sometimes happens that λ_{n-1} is small enough (usually $\lambda_{n-1} \approx 10^{-5}$); however, the smallest

Table 5.3: *The smaller δ, the close simplified network to the original one*

δ	5%	15%	25%
Err_p	$4.38e^{-5}$	$1.64e^{-4}$	$5.79e^{-4}$

conductance is large enough. As a result, no simplification is possible.

Since not full spectrum of eigenvalues is required, one can use iterative methods like Arnoldi [55] or Lanczos [67] to compute the second smallest eigenvalue, λ_{n-1}, in (5.47) or the largest magnitude eigenvalue $|\epsilon|$ in (5.46).

Example

To show that Err_{tpa} in fact indicates how close the original and the simplified networks are, we performed the following experiment. We applied simplification by Err_{tpa} to the reduced network obtained by *ReduceR* from an original resistor network [74], which has 12661 internal nodes, 160 ports, and 23333 resistors. Then, we computed the maximum relative error, Err_p, between the path resistances of the original and simplified network. From Table 5.3 it can be seen that the smaller the value of δ (with $Err_{tpa} \leq \delta$), the smaller the value of Err_p, which implies that a network with smaller Err_{tpa} is closer to the original network than a network with larger Err_{tpa}.

5.6.5 Implementation issues

Algorithms for simplification of resistor networks can be based on the algorithms for deleting resistors by groups presented in Section 5.5. Within these algorithms, instead of computing the errors Err_v, Err_p or Err_{tp} directly, we suggest to use their estimations Err_{vs} in (5.29), Err_{pa} in (5.38), or Err_{tpa} in (5.46).

5.7 Numerical results

We will show how the suggested above approach for simplification of resistor networks and reduction by *ReduceR* work for the networks from industry. The networks I, II and IV come from realistic designs of very-large-scale integration chips [74] and the network III comes from handlewafer model which has a specific structure similar to the one in

Figure 5.1. The simplification algorithms based on the selective strategies *Golden search 1* and *Golden search 2* have been implemented in Matlab 7.5 and have been tested on Core 2 Duo 1.6 GHz PC. Since the given resistor networks are not grounded, to compute Err_{pa} and Err_p, we initially ground the first terminal and after simplification plug it back to the networks.

5.7.1 Simplification by Err_{pa} and Err_{vs} applied to the original networks

To investigate how sharp Err_{pa} is in comparison to Err_p, we performed simplification to the original networks by Err_{pa} and Err_p individually. From Table 5.4 it can be seen that simplification with Err_{pa} works faster than simplification with Err_p. However, for the networks I and II, Err_{pa} allows to delete fewer resistors than Err_p.

Table 5.5 shows a similar experiment but with the use of the option *Golden search 2*. As expected, *Golden Search 2* works faster than *Golden search 1*, and it allows, in general, to delete fewer resistors.

Table 5.6 shows the results obtained after applying simplification by Err_{vs} to the *original* networks. For $\delta = 5\%$, simplification by Err_{vs} works slower than simplification by Err_{pa}. This happens due to the costs of computing the matrix vector product of the form $(I - \widetilde{G}^{-1}G)\mathbf{x}$ within the Krylov-Schur method [84], which involves solving the system of the form $\widetilde{G}\mathbf{y} = G\mathbf{x}$ for $\mathbf{y}$. For the networks with large scale G (e.g., network III), this step becomes time-consuming. Nevertheless, the estimation Err_{vs} shows to be sharper than Err_{pa}.

Figure 5.3 shows the values of $Err_{vs} \equiv \sigma_{max} < 5\%$ and corresponding $Err_{pa} \equiv \lambda_{max}$ computed on the sequence of conductance matrices obtained during *Golden search 2*. For the network III, 10222 resistors from 70006 have been deleted and 34 computations of σ_{max} from 53 correspond to $\sigma_{max} < 5\%$. The plot demonstrates noticeable difference in computed values of σ_{max} and λ_{max} which is result of nontrivial relation between the error estimations.

From the above we conclude that simplification applied to the *original* networks is not recommended to be considered as independent reduction since the number of resistors is generally not decreased significantly. However, for the network III simplification by Err_{pa} noticeably improves sparsity in reasonable time. Another observation is that with $\delta = 5\%$ simplification by Err_{vs} usually delivers better reduction in the amount of resistors than simplification by Err_{pa}.

Table 5.4: *Results of simplification by Err_p and Err_{pa} (*Golden search 1*) applied to three networks (I,II,III). Table includes the original number of resistors, CPU time of simplification, the number of deleted resistors after simplification and the number of computations of Err_p and Err_{pa}. $\delta = 5\%$, $Tmax = 5$, $h = 50$*

	I	II	III
# resistors originally	23222	2476	70006
CPU time for Err_p	272 s.	2.7 s.	1185 s.
CPU time for Err_{pa}	3.2 s.	0.7 s.	208 s.
# deleted resistors by Err_p	576	29	9878
# deleted resistors by Err_{pa}	35	2	9878
# computations by Err_p	74	23	78
# computations by Err_{pa}	14	10	23

Table 5.5: *Results of simplification by Err_p and Err_{pa} (*Golden search 2*) applied to three networks (I,II,III). Table includes the original number of resistors, CPU time of simplification, the number of deleted resistors after simplification and the number of computations of Err_p and Err_{pa}. $\delta = 5\%$, $Tmax = 5$, $h = 50$*

	I	II	III
# resistors originally	23222	2476	70006
CPU time for Err_p	245 s.	0.8 s.	377 s.
CPU time for Err_{pa}	2.1 s.	0.5 s.	57.6 s.
# deleted resistors by Err_p	570	18	9695
# deleted resistors by Err_{pa}	35	2	9695
# computations by Err_p	64	6	78
# computations by Err_{pa}	9	5	23

Table 5.6: *Results of simplification by Err_{vs} (*Golden search 2*) applied to three networks (I,II,III). Table includes the original number of resistors, CPU time of simplification, the number of deleted resistors after simplification, the number of computations of Err_{vs}. $\delta = 5\%$, $Tmax = 5$, $h = 50$*

	I	II	III
# resistors originally	23222	2476	70006
CPU time for Err_{vs}	23.4 s.	1.7 s.	1385 s.
# deleted resistors Err_{vs}	35	2	10222
# computations Err_{vs}	9	5	53

Table 5.7: *Results of simplification by Err_{pa} (S_1), Err_p (Sd), Err_{vs} (S_2), Err_{vc} (S_3), Err_{tpa} (S_4) and reduction by* ReduceR *(R) of four networks. $\delta = 5\%$, $T_{max} = 5$*

Network I	Original	R	$S_1 + R$	$R + S_1$	$R + Sd$	$R + S_2$	$R + S_3$	$R + S_4$
# terminals	160	160	160	160	160	160	160	160
# int. nodes	12661	138	157	138	138	138	138	138
# resistors	23222	3315	3315	1359	1244	1187	2393	1909
CPU time	-	23.8 s.	31.2 s.	25.7 s.	229 s.	28 s.	23.9 s.	24.9 s.

Network I	Original	R	$S_1 + R$	$R + S_1$	$R + Sd$	$R + S_2$	$R + S_3$	$R + S_4$
# terminals	39	39	39	39	39	39	39	39
# int. nodes	1503	8	8	8	8	8	8	8
# resistors	2476	702	702	589	541	510	641	622
CPU time	-	1 s.	1.7 s.	1.7 s.	6.2 s.	1.9 s.	1.01 s.	1.7 s.

Network I	Original	R	$S_1 + R$	$R + S_1$	$R + Sd$	$R + S_2$	$R + S_3$	$R + S_4$
# terminals	55	55	55	55	55	55	55	55
# int. nodes	31356	0	0	0	0	0	0	0
# resistors	70006	1485	1485	1480	1479	455	1485	1480
CPU time	-	62.4 s.	107 s.	68.8 s.	65.7 s.	79 s.	62.4 s.	62.9 s.

Network I	Original	R	$S_1 + R$	$R + S_1$	$R + Sd$	$R + S_2$	$R + S_3$	$R + S_4$
# terminals	76	76	76	76	76	76	76	76
# int. nodes	1134	303	308	308	308	308	308	308
# resistors	1936	1397	1397	1269	1269	1264	1385	1312
CPU time	-	1.12 s.	1.7 s.	2.1 s.	14 s.	2.8 s.	1.2 s.	1.8 s.

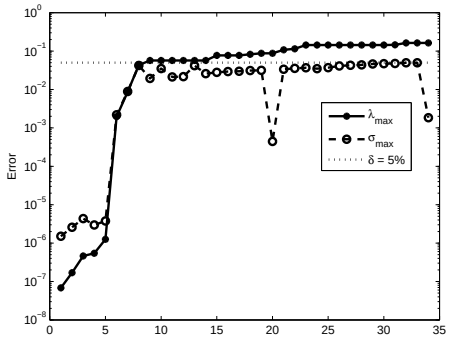

Figure 5.3: *For the network III: values of* $\sigma_{max} < 5\%$ *and corresponding values of* λ_{max} *which were computed during performance of* Golden search 2.

5.7.2 Simplification and reduction by Err_{pa}

We will show how simplification by Err_{pa} and by Err_p with *Golden search 1* approach works together with reduction by *ReduceR*. Table 5.7 demonstrates results of simplification and reduction by *ReduceR* applied to four networks in different combinations: 1) reduction, 2) simplification by Err_{pa} and then reduction, 3) reduction and then simplification by Err_{pa}, and 4) reduction and then simplification by Err_p.

Simplification applied after reduction works better than before reduction. This happens because simplification does not make crucial changes in the topology of the networks which can be recognized by *ReduceR*. Thus combinations 3 and 4 are, in general, better in the amount of resistors than the combination 2. When a reduced network has many internal nodes and terminals (see Table 5.7, e.g., networks I and IV), combination 3 shows better compromise between the amount of deleted resistors and time than combination 4. In this case direct computation of Err_p becomes very time-consuming. If the number of terminals and internal nodes is not large (e.g., networks II and III), then combination 4 is preferable since it allows to delete more resistors in small extra time.

Simplification of a reduced network with many internal nodes is, in general, more efficient than simplification of the reduced network with only a few internal nodes: the more internal nodes, the more options for neglecting resistors which do not affect path resistances. This explains why the simplification after reduction (in the amount of resistors) of the networks I and IV is better than in the case of the networks II and III.

The larger parameter T_{max}, introduced in Section 5.5, the more resistors can be deleted and more time is required. For the network IV, we used $T_{max} = 5$. This allowed us to delete after reduction 128 resistors in 0.9 sec., while with $T_{max} = 80$ one could delete 153 resistors in 4 sec.

Figure 5.4 confirms that after simplification by Err_{pa}, the relative error of path resistances, Err_p, in (5.32) stays smaller than $\delta = 5\%$.

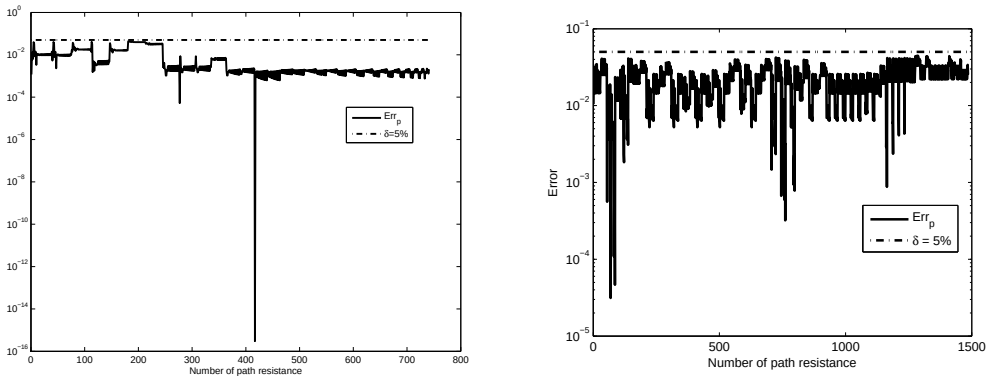

Figure 5.4: *Comparison of computed error Err_p with 5% error for each path resistance for the networks I (left) and III (right).*

5.7.3 Simplification and reduction by Err_{vs}, Err_{vc}, Err_{pa} and Err_{tpa}

In the previous section we have shown that simplification applied after reduction leads to fewer resistors than simplification applied before reduction. Therefore, in this section we consider the case of applying simplification by Err_{vs}, Err_{vc}, Err_{pa} and Err_{tpa} after reduction by *ReduceR*.

From Table 5.7 it can be seen that simplification by Err_{vc} is faster than all other estimations applied after *ReduceR*. As was mentioned in Section 5.6.1, Err_{vc} depends on the condition number of G which makes it less sharp than Err_{vs}.

Noticeable reduction by Err_{vs} has been achieved for the networks I and III, where the number of resistors has been decreased up to 65% and 70%, respectively. Thus estimation, Err_{vc}, can be considered for fast improvements in the amount of resistors, while Err_{vs} can be used for obtaining more advanced reduction.

To compare the overall CPU simulation time, we measured the time required to compute the path resistances. We first applied *ReduceR* and then simplification by Err_{vs} to obtain the reduced network. The usage of simplification after *ReduceR* provides a further reduction in simulation time of 30% for the network I, and similarly, for the networks II, III, and IV the CPU time was reduced at 4%, 2%, and 4%, respectively.

Figure 5.5 demonstrates that the higher tolerance δ, the more resistors can be deleted. With $\delta = 20\%$, computational time is increased by 10% maximum.

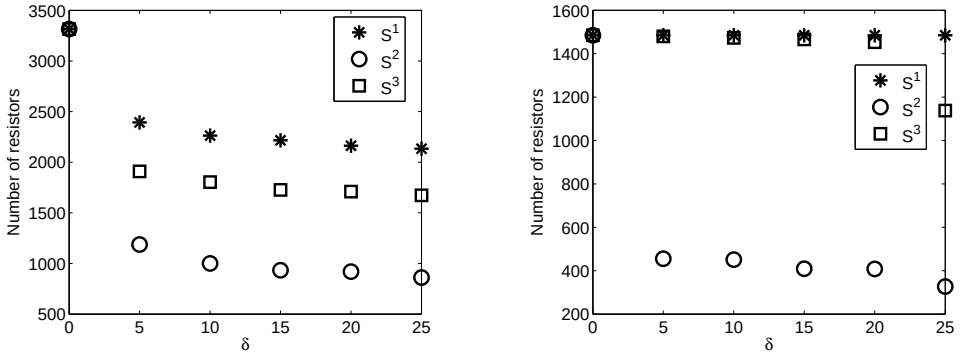

Figure 5.5: *Tolerance δ versus the number of resistors in the networks (I-left, III-right) after simplification (S^1 - by the estimation Err_{vc}, S^2 - by Err_{vs}, S^3 - by Err_{tpa}) applied after reduction by* ReduceR.

For the purpose of illustration, Figure 5.6 shows values of $Err_{vs} \equiv \sigma_{max} < 5\%$ applied after *ReduceR* and corresponding values of $Err_{pa} \equiv \lambda_{max}$ computed on the sequence of conductance matrices obtained during *Golden search 1*. One can observe that λ_{max} increases monotonically. This happens due to the fact that the path resistance is a monotonic function [30]. To show this, let g and $\widetilde{g}$ $(0 \leq g \leq \widetilde{g})$ be conductances and G, $\widetilde{G}$ be corresponding conductance matrices. Since $G \leq \widetilde{G}$ (i.e., $\widetilde{G} - G$ is positive semi-definite), it follows that $\mathbf{e}_{ij}^T G^{-1} \mathbf{e}_{ij} \geq \mathbf{e}_{ij}^T \widetilde{G}^{-1} \mathbf{e}_{ij}$, i.e., $R_{ij} \geq \widetilde{R}_{ij}$. Therefore, deleting each new resistor from G makes the relative error, Err_p, in (5.32) and its estimation, Err_{pa}, non-decreasing functions.

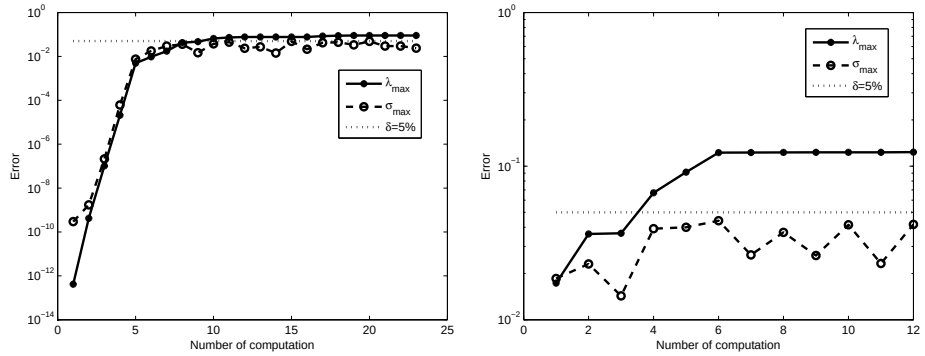

Figure 5.6: *For the network I (left) and II (right): values of $\sigma_{max} < 5\%$ and corresponding values of λ_{max} which were computed during performance of* Golden search 1.

5.8 Error estimation for $\frac{||\mathbf{v}_e-\widetilde{\mathbf{v}}_e||}{||\mathbf{v}_e||}$

In Section 5.6 we have derived two estimations for the error $Err_v = \frac{||\mathbf{v}-\widetilde{\mathbf{v}}||}{||\mathbf{v}||}$. In this section we will derive an estimation for $Err_{ve} = \frac{||\mathbf{v}_e-\widetilde{\mathbf{v}}_e||}{||\mathbf{v}_e||}$, which involves voltages only at external nodes. The insight behind this is that current can be injected only in external nodes; therefore, the error does not need to include internal nodes. However, such restriction makes more difficult to derive an estimation because instead of operating with conductance matrix, G, we have to operate with its Schur complement, G_s.

In this section we will show the main idea of derivation of an error estimation for Err_{ve}, which is independent of the current $\mathbf{i}_n$. Details of derivation and computational costs can be found in Appendix B.3.

Let a resistor $1/g_1$ be deleted from a grounded network with conductance matrix G, i.e., the new conductance matrix becomes $\widetilde{G} = G - \mathbf{m}g\mathbf{m}^T$. Eliminating all internal nodes from G and $\widetilde{G}$ leads to the equivalent conductance matrices $G_s = G_{11} - G_{12}G_{22}^{-1}G_{12}^T$ and $\widetilde{G}_s = \widetilde{G}_{11} - \widetilde{G}_{12}\widetilde{G}_{22}^{-1}\widetilde{G}_{12}^T$ respectively. Moreover,

$$Err_{ve} = \frac{||(G_s^{-1} - \widetilde{G}_s^{-1})\mathbf{i}_n||}{||G_s^{-1}\mathbf{i}_n||} \leq \frac{||(G_s^{-1} - \widetilde{G}_s^{-1})||}{||G_s^{-1}\frac{\mathbf{i}_n}{||\mathbf{i}_n||}||} \leq ||(G_s^{-1} - \widetilde{G}_s^{-1})|| \sup_{||\mathbf{i}_n||=1} \frac{1}{||G_s^{-1}\mathbf{i}_n||}. \tag{5.48}$$

Since G_s is symmetric positive definite, G_s^{-1} exists and

$$\sup_{||\mathbf{i}||=1} \frac{1}{||G_s^{-1}\mathbf{i}_n||} = \sup_{||\mathbf{w}||\neq 0} \frac{||\mathbf{w}||}{||G_s^{-1}\mathbf{w}||} = \sup_{||G_s\mathbf{z}||\neq 0} \frac{||G_s\mathbf{z}||}{||\mathbf{z}||} \tag{5.49}$$

$$= \sup_{||\mathbf{z}||\neq 0} \frac{||G_s\mathbf{z}||}{||\mathbf{z}||} = ||G_s||. \tag{5.50}$$

Here $||G_s||$ is the norm of matrix G_s induced by the vector norms $||G_s\mathbf{z}||$ and $||\mathbf{z}||$. Note that an arbitrary vector norm can be used. Substituting (5.50) into (5.48) leads to

$$Err_{ve} \leq ||(G_s^{-1} - \widetilde{G}_s^{-1})|| \cdot ||G_s|| \tag{5.51}$$

$$= ||(G_{11} - G_{12}G_{22}^{-1}G_{12}^T)^{-1} - (\widetilde{G}_{11} - \widetilde{G}_{12}\widetilde{G}_{22}^{-1}\widetilde{G}_{12}^T)^{-1}|| \cdot ||G_s|| \equiv Err_{ves}, \tag{5.52}$$

Thus, Err_{ves} is a bound of Err_{ve}, i.e., $Err_{ve} \leq Err_{ves}$, which is independent of the current $\mathbf{i}_n$.

Note that to compute Err_{ves} efficiently, we can try to delete resistors only between inter-

nal nodes. It means that upon deleting a resistor

$$G_{11} = \widetilde{G}_{11},\ G_{12} = \widetilde{G}_{12}. \tag{5.53}$$

Deleting the first resistor $1/g_1$ from G_{22} leads to

$$\widetilde{G}_{22} = G_{22} - \mathbf{m}_1 g_1 \mathbf{m}_1^T,$$

where $\mathbf{m}_1$ is defined according to the notation in (5.4). Applying the Sherman-Morrison formula (5.19) to $\widetilde{G}_{22}$ one obtains

$$\widetilde{G}_{22}^{-1} = G_{22}^{-1} - \mathbf{h}_1 c_1 \mathbf{h}_1^T, \tag{5.54}$$

where $\mathbf{h}_1 = G_{22}^{-1}\mathbf{m}_1$ and the scalar c_1 is defined as

$$c_1 = \left(\frac{1}{-g_1} + \mathbf{m}_1^T \mathbf{h}_1\right)^{-1}. \tag{5.55}$$

Substituting (5.53) and (5.54) into (5.52) and applying the Sherman-Morrison formula (5.19) leads to

$$\begin{aligned} Err_{ves} &= ||G_s^{-1} - \widetilde{G}_s^{-1}|| \cdot ||G_s|| \\ &= ||G_s^{-1} - \left(G_{11} - G_{12}\left(G_{22}^{-1} - \mathbf{h}_1 c_1 \mathbf{h}_1^T\right) G_{12}^T\right)^{-1}|| \cdot ||G_s|| \\ &= ||G_s^{-1} G_{12} \mathbf{h}_1 \widetilde{c}_1 \mathbf{h}_1^T G_{12}^T G_s^{-1}|| \cdot ||G_s||, \end{aligned} \tag{5.56}$$

where the scalar $\widetilde{c}_1$ is

$$\widetilde{c}_1 = \left(\frac{1}{c_1} + \mathbf{h}_1^T G_{12}^T G_s^{-1} G_{12} \mathbf{h}_1\right)^{-1}. \tag{5.57}$$

Note that Err_{ves} in (5.56) corresponds to a bound after deleting a single resistor. After that, the second resistor can be deleted from $\widetilde{G}_{22}$ and, thus, c_1, $\widetilde{c}_1$, and $\mathbf{h}_1$ have to be recomputed to obtain an updated error bound Err_{ves}. In Appendix B.3 we show a generalization of the error estimation (5.56) as well as the complexity to compute it.

A disadvantage of the presented error estimation, Err_{ves}, is that it allows us to delete resistors one-be-one. Deleting resistors by groups will increase complexity proportionally to the number of resistors in a group. Therefore, the use of the algorithms *Golden Search 1* and *Golden Search 2*, proposed in Section 5.5, is not required.

Example

Figure 5.7 shows sorted values of resistors in decreasing order and values of Err_{ve} and Err_{ves} calculated by deleting each resistor *individually*. To compute Err_{ve} we considered

Table 5.8: *Number of resistors after reduction by* ReduceR *(R), and after reduction and simplification by* Err_{ves} *(R+S). Parameters are* $\delta = 5\%$, $Tmax = 5$

	I	II	III
# resistors (R)	3315	702	1485
# internal nodes (R)	138	8	0
# resistors (R + S)	3074 (32 sec.)	697 (1 sec.)	1485

current sources with magnitude 1 connected to all ports. It can be seen that for both networks the error bound, Err_{ves}, is much more "strict" than Err_{ve}. Both Err_{ve} and Err_{ves} have similar behaviour and are shifted from each other. It is also can be noticed that removing resistors with larger values does not always deliver smaller errors. However, the tendency "the larger resistor, the smaller error" can be clearly observed.

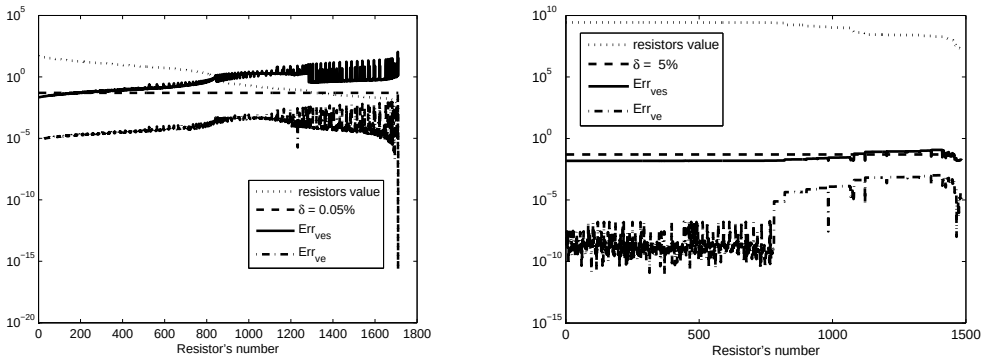

Figure 5.7: *Dependence between the deleted resistor and the value of exact relative error* Err_{ve}*, and its bound* Err_{ves}*.*

From Table 5.8 it can be seen that simplification applied after reduction by *ReduceR* appears time-consuming, if the number of deleted resistors is large (e.g., for the network I). It happens due to the high complexity of the algorithm. If the number of internal nodes is small, then simplification by Err_{ves} is poor (e.g., networks II and III). It happens because computation of Err_{ves} requires deleting resistors from the block G_{22}, which involves only internal nodes. Therefore, if this block does not contain many resistors, then reduction is not expected to be extensive. Comparing the above results with the results in Table 5.7, we conclude that Err_{ves} is the most expensive and non-sharpest estimation.

5.9 Relation with incomplete factorizations

In this section we compare the simplification procedure with incomplete factorizations. Incomplete factorizations play an important role as preconditioners for iterative solvers [18], [13]. Let $A \in \mathbb{R}^{n\times n}$ be a sparse matrix with elements a_{ij}. The problem of finding a preconditioner for a system $Ax = b$ is to find a matrix M in such a way that M is a good approximation to A in some sense and the system $Mx = b$ is much easier to solve than the original system. The preconditioned system $M^{-1}Ax = M^{-1}b$ is solved with an iterative method till the solution converges. Typically, the convergence rate depends on the condition number, $\kappa(M^{-1}A)$, of the preconditioned system [18]. Therefore, it is desirable to choose M such that the preconditioned iteration converges rapidly, while the cost of each iteration is not too expensive.

One of the ways of defining a preconditioner M is to perform an incomplete factorization of the original matrix A. Based on the fact that the incomplete factorizations are aimed on constructing approximate, sparser factors than the complete factorizations, we consider to use the incomplete factorizations as some sort of simplification.

Since in our particular case the conductance matrix (after grounding) is symmetric positive definite, we will consider the incomplete Cholesky factorization. The incomplete LL^T factorization process computes an approximate Cholesky factor, that is a lower triangular matrix L, such that $A = LL^T + E$, where E is an error matrix. Matrix L is computed as an complete factor by dropping some fill-in. Fill-in can be discarded based on several different criteria, such as position, value, or a combination of both [13].

Before going further, one should notice that simplification is not aimed at the construction of the factorization. Simplification works directly on G rather than with the factors. On the other hand, incomplete factorization does not guarantee that $\left|\left|\frac{\mathbf{v}-\tilde{\mathbf{v}}}{\mathbf{v}}\right|\right|$ or $\left|\frac{R_{ij}-\widetilde{R}_{ij}}{R_{ij}}\right|$ is small enough, but instead it is aimed on delivering $||G - LL^T||_F$ small, which is required for fast convergence of iterative solvers.

In practice, due to round-off errors, it might happen that the product LL^T is more dense than the original G. Performing the incomplete Cholesky factorization with the use of a drop tolerance [13], [75], one may compute such factor, L, which makes the product LL^T less dense than the original matrix G. For the implementation of the incomplete Cholesky factorization we used the build-in Matlab function *cholinc*.

Table 5.9 shows examples of the drop tolerances, which deliver smaller fill-in than the original conductance matrices and also the computed values of the errors $Err_v = \frac{||\mathbf{v}-\tilde{\mathbf{v}}||}{||\mathbf{v}||}$ and $Err_p = \left|\frac{R_{ij}-\widetilde{R}_{ij}}{R_{ij}}\right|$. The error Err_v was computed with random input currents injected

Table 5.9: *Comparison of the original conductance matrix G with the product of incomplete factors LL^T which have been computed for a given drop tolerance* droptol

network	nonzero elements of G	*droptol*	nonzero elements of LL^T	$\frac{\|\|G-LL^T\|\|_\infty}{\|\|G\|\|_\infty}$	Err_p	Err_v
I	59258	0.1	40126	0.58	1.08	0.99
II	6487	0.2	5161	0.73	0.85	0.98
II	171416	0.05	164098	0.09	0.9091	0.38

Table 5.10: *Comparison of the original conductance matrix G with the product of incomplete factors LL^T which have been computed for given drop tolerances $droptol_1$ and $droptol_2$. Notation nnz(A) denotes the number of nonzero elements of matrix A*

network	nnz(G)	$droptol_1$	nnz(LL^T)	Err_v	$droptol_2$	nnz(LL^T)	Err_p
I	59258	10^{-5}	100510	0.02	10^{-8}	102438	0.031
II	6487	10^{-4}	18091	0.01	10^{-5}	17883	0.02
III	171416	10^{-7}	1556516	0.03	$5 \cdot 10^{-8}$	1644386	0.03

to all terminals. We observe that the computed values of Err_v and Err_p are higher than the correspondent drop tolerances.

Table 5.10 suggests some tolerances which deliver $Err_v < 0.05$ or $Err_p < 0.05$. However, it can be seen that the number of nonzero entries (nnz) in the product LL^T is significantly higher that in G. To conclude, one does not know in advance which drop tolerance to choose in order to deliver $Err_v < \delta$ or $Err_p < \delta$. The above examples demonstrate that the incomplete Cholesky factorization with a drop tolerance, such that $Err_v < 5\%$ or $Err_p < 5\%$, cannot be used for simplification since it delivers more dense conductance matrices than the original G.

5.10 Summary of the error estimations

In this section we summarize the properties of the error estimations, which were used for simplification of resistor networks. Table 5.11 shows a comparison between the error estimations from different perspectives, such as possibility to delete resistors by groups,

Table 5.11: *Comparison of the error estimations from different perspectives*

Error	Estimation	Requirements to compute estimation	Deleting by groups	Ground	Application to large networks
$Err_v = \frac{\|\|\mathbf{v}-\widetilde{\mathbf{v}}\|\|}{\|\|\mathbf{v}\|\|}$	Err_{vc}	(1a) computing the condition number of G	yes	yes	yes
	Err_{vs}	(1b) computing the maximum singular value of $I - \widetilde{G}^{-1}G$	yes	yes	yes
$Err_{ve} = \frac{\|\|\mathbf{v}_e-\widetilde{\mathbf{v}}_e\|\|}{\|\|\mathbf{v}_e\|\|}$	Err_{ves}	resistors between internal nodes are candidates to be deleted	no	yes	no
$Err_p = \frac{\|R-\widetilde{R}\|}{\|\widetilde{R}\|}$	Err_{pa}	computing the largest magnitude eigenvalue of $I - \widetilde{G}^{-1}G$	yes	yes	yes
$Err_{tp} = \frac{\|R_{tp}-\widetilde{R}_{tp}\|}{\|\widetilde{R}_{tp}\|}$	Err_{tpa}	(2a) computing the largest magnitude eigenvalue of $\widetilde{G} - G$	yes	no	yes
		(2b) resistors which have no common nodes and satisfy (5.47) may be deleted	no	no	yes

necessity to ground a node before simplification, and applicability for reduction of large networks. Most estimations require grounding (i.e., positive definite G) and most of them can be incorporated with the algorithms *Golden search 1* and *Golden Search 2* to delete resistors by groups. The most attractive estimations are Err_{vs} and Err_{pa} since they have shown to be applicable for large networks while having a good reduction quality. In general, all considered estimations show better performance, when they are applied to the reduced networks, than when they are applied to the the original ones.

5.11 Concluding remarks

In this chapter we have considered approaches for reduction of the number of resistors in resistor networks. The suggested approaches, to which we collectively referred as

"simplification", can improve the sparsity of the original and/or reduced conductance matrix by neglecting resistors which do not contribute significantly to the behaviour of the circuit. Four criteria for measuring the quality of approximation have been suggested and corresponding error bounds have been derived. Obtained error bounds allow to keep a strict control on the accuracy of the reduced resistor networks and demand less computational effort than the direct error computations, especially if the number of ports is large. Based on these estimations, we suggested a few simplification algorithms, which allow to delete resistors by groups. We also demonstrated that the incomplete Cholesky factorization cannot be efficiently used for the role of simplification. The considered simplification algorithms, applied after reduction by *ReduceR*, improved total reduction up to 70%. Since the success of simplification also depends on the values of conductances in resistor networks, simplification can be considered as a complementary procedure to existing exact reduction techniques.

Chapter 6

Substrate extraction

In the design and fabrication of micro-electronic circuits, it is necessary to simulate and predict many kinds of effects, such as substrate crosstalk, interconnect delays and others. In order to simulate and predict properly these effects, accurate and efficient substrate modeling methods are required. Substrate resistance extraction involves finding a resistance network between the ports that correctly describes the behaviour of the substrate. In this chapter we consider the problem of resistance extraction of a substrate with a homogeneous doping profile. We solve the problem by means of two discretization methods, namely the finite element method (FEM) and the boundary element method (BEM) and discuss the advantages and disadvantages of each of these methods. We particularly addresses the problem of achieving grid-independent results and characterize the cases in which one technique is better than the other.

6.1 Mathematical formulation of the problem

In Chapter 2 we discussed that the problem of modeling a substrate can be transformed into the problem of finding a resistance network between the contacts. An example of a 3D substrate with 3 contacts is presented in Figure 2.6. For the goal of modeling substrate one has to solve the Laplace equation under appropriate boundary conditions. The Laplace equation for the potential u in the domain Ω with the boundary Γ is

$$\sigma \nabla^2 u = 0, \tag{6.1}$$

where σ is conductivity of the domain (we consider a homogenous domain, i.e., σ is constant). The boundary conditions are chosen such that current can enter or leave

the domain through the contact areas while the remaining boundaries have insulating properties. In other words, we have Dirichlet boundary conditions on the contact areas, i.e.,

$$u = \bar{u} \quad \text{on} \quad \Gamma_1, \tag{6.2}$$

and homogeneous Neumann boundary conditions on the remaining part of the boundary, i.e.,

$$\frac{\partial u}{\partial \mathbf{n}} = \bar{q} = 0 \quad \text{on} \quad \Gamma_2, \tag{6.3}$$

where $\mathbf{n}$ is the normal to the boundary $\Gamma = \Gamma_1 \bigcup \Gamma_2$ (note that $\Gamma_1 \bigcap \Gamma_2 = 0$). In fact, setting $\frac{\partial u}{\partial \mathbf{n}} = 0$, implies that the normal component of the current density J_n (through Γ_2) is zero as well. This can be concluded from the relation

$$J_n = \sigma \frac{\partial u}{\partial \mathbf{n}} \quad \text{on} \quad \Gamma_2.$$

Therefore, the homogenous Neumann boundary condition $\frac{\partial u}{\partial \mathbf{n}} = 0$ on Γ_2 implies $J_n = 0$, i.e., no current flows through the boundary Γ_2.

6.2 The Finite Element Method

FEM is a popular technique for substrate modeling [29] [12]. We will formulate the 2D boundary value problem (6.1)–(6.3) by FEM and will show the process of extraction of the resistance network. First, we construct the weak formulation of the problem by defining the weighted residual of (6.1) for a single element with domain Ω_h. The residual element takes the form:

$$r_h = \sigma \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right).$$

Since the numerical solution u is generally not identical to the exact solution, r_h is nonzero. Our objective is to minimize r_h in a weighted sense. To achieve this, we first multiply r_h with a weight function ω, then integrate the result over the area of the element, and then set the integral to zero:

$$\sigma \int_{\Omega_h} \omega \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) d\Omega_h = 0. \tag{6.4}$$

Taking into account that $\omega \frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}(\omega \frac{\partial u}{\partial x}) - \frac{\partial \omega}{\partial x} \frac{\partial u}{\partial x}$, (6.4) can be rewritten as

$$\sigma \int_{\Omega_h} \left(\frac{\partial}{\partial x} \left(\omega \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(\omega \frac{\partial u}{\partial y} \right) \right) d\Omega_h - \sigma \int_{\Omega_h} \left(\frac{\partial \omega}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial \omega}{\partial y} \right) d\Omega_h = 0. \tag{6.5}$$

The Green's theorem states that the area integral of the divergence of a vector quantity equals to the total outward flux of the vector quantity through the contour that bounds the area, i.e.,

$$\int_{\Omega_h} \left(\frac{\partial A_x}{\partial x} + \sigma \frac{\partial A_y}{\partial y} \right) d\Omega_h = \oint_{\Gamma_h} (A_x + A_y) \cdot \mathbf{a}_n dl,$$

where $\mathbf{a}_n = a_x \mathbf{n}_x + a_y \mathbf{n}_y$ is the outward unit vector that is normal to the boundary of the element. Applying Green's theorem to the first integral of (6.5), one obtains:

$$\int_{\Omega_h} \left(\frac{\partial \omega}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial \omega}{\partial y} \right) d\Omega_h = \oint_{\Gamma_h} \omega \left(\frac{\partial u}{\partial x} \mathbf{n}_x + \frac{\partial u}{\partial y} \mathbf{n}_y \right) dl. \tag{6.6}$$

According to the Galerkin approach, the weight function ω must belong to the same set of basis functions that are used to interpolate u. In general, we interpolate u with the set of Lagrange polynomials as

$$u = \sum_{i=1}^{n} N_i(x, y) u_i^e, \tag{6.7}$$

where N_j are the corresponding basis functions based on Lagrange polynomials (interpolation functions), n is the number of local nodes per element, and u_i^e are the unknown coefficients of a single mesh element. Substituting (6.7) into (6.6), and setting

$$\omega = N_i, \quad i = 1, \ldots n,$$

the weak form of the differential equation becomes

$$\int_{\Omega_h} \left(\frac{\partial N_i}{\partial x} \sum_{j=1}^{n} u_j^e \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \sum_{j=1}^{n} u_j^e \frac{\partial N_j}{\partial y} \right) d\Omega_h =$$

$$\oint_{\Gamma_h} N_i \left(\frac{\partial u}{\partial x} \mathbf{n}_x + \frac{\partial u}{\partial y} \mathbf{n}_y \right) dl, \quad i = 1, \ldots, n.$$

This equation can be rewritten in matrix form as follows

$$Y_h \mathbf{u}_h = \mathbf{i}_h, \tag{6.8}$$

where

$$Y_{h,ij} = \int_{\Omega_h} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) d\Omega_h, \tag{6.9}$$

$$\mathbf{u}_h = \begin{pmatrix} u_{h,1} & \ldots & u_{h,n} \end{pmatrix}^T, \tag{6.10}$$

$$\mathbf{i}_{h,i} = \oint_{\Gamma_h} N_i \left(\frac{\partial u}{\partial x} \mathbf{n}_x + \frac{\partial u}{\partial y} \mathbf{n}_y \right) d\Omega_h, \quad i, j = 1, \ldots, n. \tag{6.11}$$

The above Y_h, $\mathbf{u}_h$ and $\mathbf{i}_h$ correspond to a stamp which represents a single finite element. For example, if the finite element is triangular with linear basis functions, then Y_h be-

comes a 3×3 matrix, and the vector $\mathbf{u}_h$ contains the voltages at the three nodes in the corners of the element. If a finite element has no edges at the boundary, then the integral in (6.11) equals zero. This follows from the conservation current law, which states that $\oint_{\Gamma_h} \mathbf{n} \cdot \mathbf{J} dl = \int_{\Omega_h} \nabla \cdot \mathbf{J} ds = 0$. When we use linear basis functions, each edge of the FEM discretization is equivalent to a resistance branch in the equivalent network. This is schematically presented in Figure 6.1.

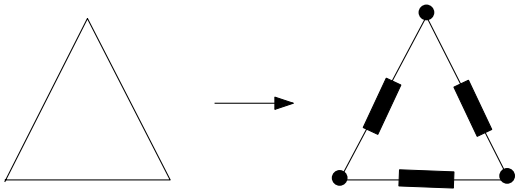

Figure 6.1: *Left triangular from FEM discretization represents resistance network on the right.*

To evaluate the integrals in (6.9) and (6.11), it is necessary to make a change of variables. In other words, instead of integrating over the triangular element on the regular coordinate system, it is more convenient to carry out the integration on a master triangle which lies on the natural coordinate system. Details about the evaluation of the integrals in (6.9) and (6.11) for linear triangular elements can be found in [29] [71].

Based on the stamp for a single element, the global matrix for the whole domain Ω can be constructed. Let N denote the number of nodes in the global discretization. In this case we can define the following system

$$Y\mathbf{U} = \mathbf{I}, \tag{6.12}$$

where $Y \in \mathbb{R}^{N \times N}$ is a symmetric matrix with elements defined as in (6.9), $\mathbf{I} \in \mathbb{R}^N$ contains the components defined in (6.11), and $\mathbf{U} \in \mathbb{R}^N$ denotes the vector of potentials at the nodes. Note, that for the interior edges of the finite elements, the contribution of (6.11) is zero.

Since each element, in a triangular mesh, interacts only with neighboring elements, not all entries of Y are filled. This makes Y a sparse matrix. Therefore, the resulting resistance network is large and sparse. However, only a small number of FEM nodes belong to the contact areas, while the rest of FEM nodes are internal nodes. Since the internal nodes are not connected to other physical structures, the internal nodes can be eliminated in order to obtain a resistance network, where only the nodes at the contacts remain. Below we show how this can be done.

Conductance matrix

Since not all nodes belong to the contact areas, we subdivide the nodes into two subsets. The first subset includes the nodes at the contacts, i.e., external nodes or ports (index p). The second subset includes all other nodes, i.e., internal nodes (index i). Thus, we rewrite equation (6.12) in a block matrix form as

$$\begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_p \\ \mathbf{U}_i \end{pmatrix} = \begin{pmatrix} \mathbf{I}_p \\ \mathbf{I}_i \end{pmatrix}, \tag{6.13}$$

where $Y_{11} \in \mathbb{R}^{N_p \times N_p}$, $Y_{12} \in \mathbb{R}^{N_p \times N_i}$, and $Y_{22} \in \mathbb{R}^{N_i \times N_i}$, N_p denotes the number of nodes located at the contacts, and N_i the number of remaining nodes. To define a total potential at each contact area, we introduce an incidence matrix $F \in \{1, -1, 0\}^{N_p \times N_c}$ (where N_c denotes the number of contacts) as follows [29]

$$F_{ij} = \begin{cases} 1, & \text{if node } i \text{ belongs to the } j\text{-th contact} \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathbf{U}_c \in \mathbb{R}^{N_c}$ denote the potentials at the contact areas, and $\mathbf{U}_i \in \mathbb{R}^{N_i}$ denote the potentials at the nodes located outside of the contacts. Then, (6.13) can be rewritten as

$$\begin{pmatrix} F^T Y_{11} F & F^T Y_{12} \\ Y_{21} F & Y_{22} \end{pmatrix} \begin{pmatrix} \mathbf{U}_c \\ \mathbf{U}_i \end{pmatrix} = \begin{pmatrix} \mathbf{I}_c \\ \mathbf{0} \end{pmatrix}, \tag{6.14}$$

where $\mathbf{I}_c \in \mathbb{R}^{N_c}$ is the total current at the contacts. Eliminating $\mathbf{U}_i$ one obtains

$$\mathbf{I}_c = \underbrace{\left(F^T Y_{11} F - F^T Y_{12} Y_{22}^{-1} Y_{21} F\right)}_{G_s} \mathbf{U}_c,$$

where G_s is a symmetric positive semidefinite conductance matrix, which contains the path resistances of the network.

6.3 The Boundary Element Method

Now we will solve the 2D boundary value problem (6.1)–(6.3) by BEM and show the process of extraction of the resistance network. BEM is based on an integral form of the Laplace equation [17] [82]. Taking into account Dirichlet and Neumann boundary conditions, the integral form of the Laplace equation has the form

$$\int_\Omega (\nabla^2 u)\omega d\Omega - \int_{\Gamma_2} \left(\frac{\partial u}{\partial \mathbf{n}} - \bar{q}\right) \omega d\Gamma + \int_{\Gamma_1} (u - \bar{u}) \frac{\partial \omega}{\partial \mathbf{n}} d\Gamma = 0, \tag{6.15}$$

where ω denotes an arbitrary weight function, which is continuous up to the second derivative. Further we will choose ω equal to the Green's function. The Green's function $G(p,q)$ is the fundamental solution of the Laplace equation:

$$\nabla^2 G(p,q) + \Delta^p = 0, \tag{6.16}$$

where Δ^p represents a Dirac Delta function, which tends to infinity at the point $x = x^p$ and is equal to zero everywhere else. The Green's function $G(p,q)$ can be viewed as the function describing the potential at a position p, resulting from a unit point charge placed at position q. For the 2D homogeneous case of the Laplace equation, the Green's function is

$$G(p,q) = \frac{1}{2\pi} \ln \frac{1}{r},$$

and for the 3D case

$$G(p,q) = \frac{1}{4\pi r},$$

where $r = |p - q|$. Let the weight function ω be chosen as the Green's function. Substituting $\omega = G(p,q)$ into (6.15), and integrating by parts twice, we obtain

$$\int_\Omega (\nabla^2 G) u d\Omega = \int_{\Gamma_2} u \frac{\partial G}{\partial \mathbf{n}} d\Gamma_2 - \int_{\Gamma_2} \bar{q} G d\Gamma_2 - \int_{\Gamma_1} \frac{\partial u}{\partial \mathbf{n}} G d\Gamma_1 + \int_{\Gamma_1} \bar{u} \frac{\partial G}{\partial \mathbf{n}} d\Gamma_1. \tag{6.17}$$

From (6.16) it follows that

$$\int_\Omega (\nabla^2 G(p,q)) u d\Omega = -\int_\Omega \Delta^p u d\Omega = -u^p. \tag{6.18}$$

Substituting (6.18) into (6.17) we obtain

$$\alpha u^p = -\int_{\Gamma_2} u \frac{\partial G}{\partial \mathbf{n}} d\Gamma_2 + \int_{\Gamma_2} \bar{q} u d\Gamma_2 + \int_{\Gamma_1} \frac{\partial u}{\partial \mathbf{n}} G d\Gamma_1 - \int_{\Gamma_1} \bar{u} \frac{\partial G}{\partial \mathbf{n}} d\Gamma_1, \tag{6.19}$$

where $\alpha = 0.5$ if $p \in \Gamma$, $\alpha = 1$ if $p \in \Omega$, and $\alpha = 0$ if $p \notin \Omega$ [17]. Equation (6.19) shows that potential at the point p, i.e., u^p, can be computed as a sum of integrals over the boundaries Γ_1 and Γ_2. This equation is the integral formulation of the Laplace equation, and it is the base of BEM for the Laplace equation.

Since we consider the case of homogeneous Neumann boundaries, i.e., $\bar{q} = \frac{\partial u}{\partial \mathbf{n}} = 0$, the second integral of the right hand-side in (6.19) vanishes. Thus, to solve equation (6.19), integration over the *whole* boundary of Ω, i.e., $\Gamma_1 \bigcup \Gamma_2$ is required. The solution procedure of (6.19) will be considered in the next subsection.

By demanding an extra condition on the fundamental solution $G(p,q)$, some improve-

ment in solving (6.19) is possible. The condition is

$$\frac{\partial G(p,q)}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \Gamma_1 \text{ and } \Gamma_2. \tag{6.20}$$

Then, equation (6.19) is transformed into a simpler equation:

$$\alpha u^p = \int_{\Gamma_1} qG(p,q)d\Gamma_1. \tag{6.21}$$

It is important to notice that (6.21) involves integration only over the contact regions, and not over the whole boundary. A Green's function which satisfies (6.20) has been derived in [82], and it has been already incorporated into the advanced layout-to-circuit extractor SPACE [61] [28].

Discretization and Solution

We will subdivide the boundary of the substrate (including the contacts) into N elements with constant basis functions. It is possible to use elements with piecewise linear or quadratic basis functions; however, these approaches are more complex from the point of view of implementation; therefore, we will not consider them in this thesis. The values of u and $\frac{\partial u}{\partial \mathbf{n}}$ are assumed to be constant over each element. Taking into account the boundary conditions (6.2)–(6.3), the integral equation (6.19) can then be written for a given point $p \in \Gamma$ as follows

$$\frac{u^p}{2} + \sum_{j=1}^{N} \int_{\Gamma_{2,j}} u \frac{\partial G(p,q)}{\partial \mathbf{n}} d\Gamma = \sum_{j=1}^{N} \int_{\Gamma_{1,j}} G(p,q) \frac{\partial u}{\partial \mathbf{n}} d\Gamma. \tag{6.22}$$

Since u and $\frac{\partial u}{\partial \mathbf{n}}$ are constant over each element, they can be taken out from the integrals. Thus (6.22) takes the form

$$\frac{u^p}{2} + \sum_{j=1}^{N} \bar{H}^{pj} u^j = \sum_{j=1} K^{pj} q^j, \tag{6.23}$$

where

$$\bar{H}^{pj} = \int_{\Gamma_j} \sigma \frac{\partial G(p,q)}{\partial \mathbf{n}} d\Gamma, \quad K^{pj} = \int_{\Gamma_j} \sigma G(p,q) d\Gamma, \quad \text{and} \quad q = \frac{\partial u}{\partial \mathbf{n}}.$$

We can rewrite (6.23) in more compact form

$$\sum_{j=1}^{N} H^{pj} u^j = \sum_{j=1} A^{pj} q^j, \tag{6.24}$$

where $H^{pj} = \bar{H}^{pj}$ if $i \neq j$ and $H^{pj} = \bar{H}^{pj} + \frac{1}{2}$ if $i = j$. Thus (6.24) becomes a system of the form:

$$H\mathbf{U} = M\mathbf{Q}, \tag{6.25}$$

where $H \in \mathbb{R}^{N\times N}$ and $M \in \mathbb{R}^{N\times N}$, $\mathbf{U} \in \mathbb{R}^N$ contains the potentials at Γ_2, and $\mathbf{Q} \in \mathbb{R}^N$ contains the current fluxes at Γ_1. At some nodes, the values of u and $\frac{\partial u}{\partial \mathbf{n}}$ are known from the boundary conditions; therefore, we rearrange the system of equations in such a way that all unknowns are on the left side. The system takes the form

$$A\mathbf{X} = \mathbf{B}, \tag{6.26}$$

where X contains the unknown values of u and $\frac{\partial u}{\partial \mathbf{n}}$ on the boundary Γ. Solving this system of equations, we can obtain the values of u and $\frac{\partial u}{\partial \mathbf{n}}$ at the boundary nodes, where the boundary conditions $\bar{q}$ and $\bar{u}$, respectively, were specified. Once the system is solved, it is also possible to compute the values of u and its derivatives at any point p inside of the domain Ω. The value of u at any internal point p can be computed by the formula (6.19), which can be written as

$$u^p = \int_{\Gamma_1} \frac{\partial u(w)}{\partial \mathbf{n}} G(p,w) d\Gamma_1 - \int_{\Gamma_2} u(w) \frac{\partial G(p,w)}{\partial \mathbf{n}} d\Gamma_2. \tag{6.27}$$

Here, w denotes the integration variable. The partial derivatives, for instance, in the direction x, i.e., $\frac{\partial u}{\partial x}$, can be calculated by differentiating (6.27), i.e.,

$$\left(\frac{\partial u}{\partial x}\right)_p = \int_{\Gamma_1} \frac{\partial u(w)}{\partial \mathbf{n}} \frac{\partial G(p,w)}{\partial x} d\Gamma_1 - \int_{\Gamma_2} u(w) \frac{\partial}{\partial x}\left(\frac{\partial G(p,w)}{\partial \mathbf{n}}\right) d\Gamma_2. \tag{6.28}$$

However, for the sake of resistance extraction, it is not required to compute values of u and its derivatives at points inside of the domain Ω. We only require the values of u and $\frac{\partial u}{\partial \mathbf{n}}$ at the boundaries, and this can be achieved by solving the system (6.26).

Note, that the matrix A in (6.26) has the size of the number of boundary elements, and it is dense, while in the case of FEM discretization, the matrix Y in (6.12) is sparse and has the size of the number of interior nodes of the finite elements.

Admittance matrix

In the case of substrate modeling, the goal is to obtain an admittance matrix, which describes the behaviour of the substrate with respect to the contact areas. Here we will show how to construct the admittance matrix from the discretized equations obtained with BEM. This admittance matrix can be also interpreted as a resistance network. From (6.25) we obtain

$$\mathbf{Q} = M^{-1}H\mathbf{U},$$

where **Q** collects the current densities, **U** collects the potentials. The matrix

$$Y_e = M^{-1}H \tag{6.29}$$

denotes the admittance matrix between all boundary elements with respect to a virtual reference node, which represents the potential at infinity.

Computing $Y_e \in \mathbb{R}^{N\times N}$ requires solving N times a system of linear equations of the form $M\mathbf{x} = \mathbf{h}_i$ for $\mathbf{x}$, where $\mathbf{h}_i$ denotes the i-th column of H matrix. Using the LU decomposition of M, an approximate complexity to compute Y_e is $O(N^3) + NO(N^2)$. However, using the windowing technique, which is based on the Schur inversion algorithm [82] [60] and takes into account only influences between boundary elements that are relatively close to each other, it is possible to compute an approximate inverse of M in only $O(N)$ time.

To extract the admittance matrix for the contacts, we introduce an incidence matrix $F \in \{1, -1, 0\}^{N\times N_c}$, where N_c denotes the number of contacts, such that:

$$F_{ij} = \begin{cases} 1, & \text{if } i\text{-th boundary element belongs to the } j\text{-th contact} \\ 0, & \text{otherwise.} \end{cases}$$

Thus, the admittance matrix which represents the network between all contacts with respect to the virtual reference node becomes $Y \in \mathbb{R}^{N_c\times N_c}$:

$$Y = F^T Y_e F,$$

which is a symmetric dense matrix with positive diagonal entries and negative off-diagonal entries. By eliminating the reference node, one obtains a conductance matrix for the resistance network which contains the path resistances between each pair of contacts.

Example

Let two contacts be defined at the top of the 2D substrate as depicted in Figure 6.3. Thus, Y is a 2×2 symmetric admittance matrix:

$$Y = \begin{pmatrix} y & -y_s \\ -y_s & y \end{pmatrix},$$

which corresponds to the network presented in Figure 6.2. Eliminating the reference node, we obtain a conductance matrix, which describes the network between the contacts:

$$G = \frac{1}{2}\begin{pmatrix} y+y_s & -y-y_s \\ -y-y_s & y+y_s \end{pmatrix}.$$

Therefore, the path resistance between the contacts equals $\frac{2}{y+y_s}$.

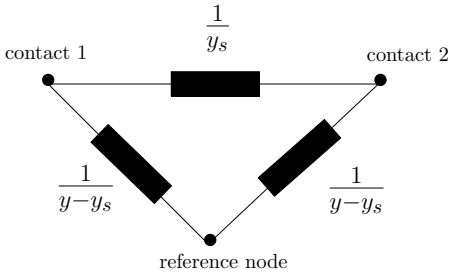

Figure 6.2: *Resistor network with two substrate contacts and a reference node.*

We note that there is another way to compute the path resistance between the contacts. By setting $\overline{u} = u_0$ at the left contact, $\overline{u} = 0$ at the right contact, and solving the Laplace equation, the path resistance, R, between the contacts can be computed as

$$R = \frac{u_0}{J_n} = \frac{u_0}{\sigma \sum_k l_k p_k},$$

where l_k denotes the length of the k-th boundary element at the right contact, and p_k denotes the current flux flowing through the k-th boundary element.

6.4 A 2D case study

Now we will compare the performance and characteristics of FEM and BEM through the following example.

We consider a 2D substrate as in Figure 6.3, of size 1000×350 μm, with two contacts 10 μm in length at a distance of 30 μm on the top the substrate. The conductivity of the substrate is 10 S/m. A unit voltage was applied to the left contact, zero voltage was set at the right contact, and no current flow through non-contact areas, i.e., $\frac{\partial u}{\partial \mathbf{n}} = 0$.

If one is interested to know the values of the potential inside of the domain at many points, then FEM is the most suitable method because it discretizes the whole domain, and the solution already contains this information. If one is only interested in the solu-

tion at the boundary, which is required, for instance, for the extraction of the resistance network, then BEM is an excellent alternative. We will study the performance of BEM and FEM for extracting a resistance network for the substrate presented in Figure 6.3.

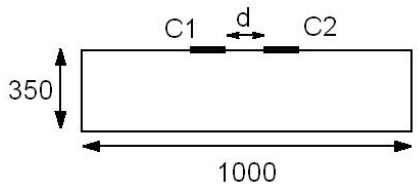

Figure 6.3: *Substrate with two contacts C1 and C2, each of the size* $10\mu m$*. Distance values are in* μm*.*

First, we solve the Laplace equation (6.1) with corresponding boundary conditions by FEM with a very fine discretization over the whole domain. We obtain the solution for the potential presented in Figure 6.4. It can be seen that it contains a high voltage zone below the left contact and a low voltage zone around the second contact. The figure also includes the current streamlines, which show that the current flows from the left contact to the right one. We note, that conductivity σ does not influence on the behaviour of the potential, but it influences on the normal component of the current J_n, going through the contacts:

$$\frac{J_n}{\sigma} = \frac{\partial u}{\partial \mathbf{n}}.$$

Second, we solve the Laplace equation (6.1) with corresponding boundary conditions by BEM implemented in Matlab [59]. We recall that for a 2D homogeneous domain, the Green's function has the form

$$G(p,q) = \frac{1}{2\pi} \ln\left(\frac{1}{r}\right).$$

Since the normal derivative of $G(p,q)$ does not vanish on the boundary, the left sum of integrals in (6.22) cannot be neglected and, therefore, we have to discretize the whole boundary of the substrate. The mesh at the contacts and between the contacts has been made finer than at the other parts of the substrate.

Figure 6.5 shows a comparison of the BEM and FEM solutions for the potential on the upper side of the substrate (where the contacts are located). To capture the behaviour of the potential by BEM, a coarse mesh at the boundaries was sufficient. From the figure we observe that both solutions by BEM and FEM coincide well.

In Figure 6.6 we show a comparison of the BEM and FEM solutions of the normal component of the current J_n at the top part of the substrate (at other parts current stays zero due to the boundary conditions). Due to the very sharp variations of the current at the contacts, we required a fine mesh for both BEM and FEM to obtain an agreement bet-

ween the solutions. In general, we observe that the potential does not require a very

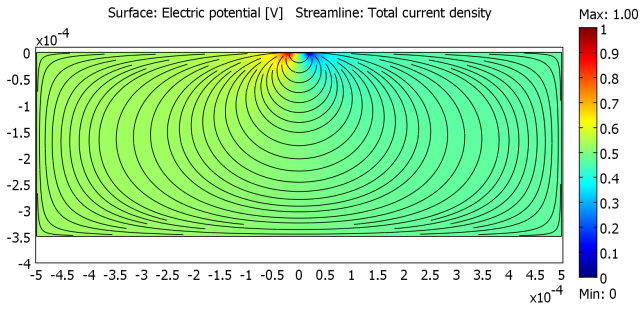

Figure 6.4: *Potential distribution and streamlines of current in the 2D substrate*

fine mesh for an adequate description, while current requires fine discretization, especially close to the contacts. An accurate value for the current at the boundary is crucial to determine the path resistance between the contacts.

Now we will proceed by analyzing the requirements of both methods (BEM and FEM) for having an accurate solution. To this extend, we use the following technique. We analyze the dependence of the computed value of the path resistance on the mesh size. When a mesh refinement conducts to a very small change in the resistance value, we can expect the solution to be mesh independent.

Figure 6.7 shows a relation between the number of FEM and BEM mesh elements and the computed value of the path resistance between the contacts. The number of mesh elements is directly related to the size of the linear system to be solved for both cases. From this plot we can notice that BEM requires a much smaller amount of elements than FEM for the computed value of path resistance to converge. This is, of course, due to the fact that BEM only requires to discretize the boundary, while FEM discretizes the whole 2D domain. Note, that we could not experience convergence of path resistance obtained by FEM with uniform refinement due to the lack of memory at the machine where the computations were performed.

To be able to compare the performance of BEM and FEM (with adaptive refinement) we measured the relative variation of path resistance based on the data of Figure 6.7. For instance, if a further refinement provides a variation in the path resistance of less than 1 %, then it means that we have reached two significant digits. This relative variation can be considered as a measure of the quality of the numerical solution. In Figure 6.8, we

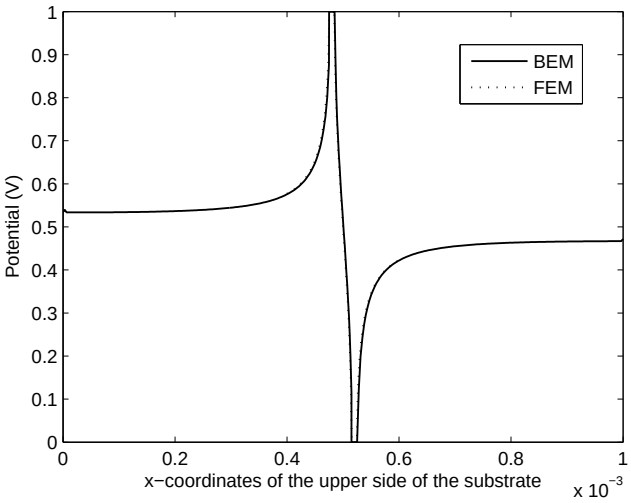

Figure 6.5: *Potential at the upper boundary of the substrate.*

show a plot of the relative variation for the path resistance versus the computation time. It can be seen that the solution computed with BEM converges at least five times faster than the solution computed with FEM. For example, to achieve a relative variation in path resistance of the order 10^{-1}, BEM requires 0.66 sec., while FEM requires 3.7 sec.

Though BEM already performs better than FEM for extracting the resistance network of homogeneous substrate, there is room for interesting improvements. Some possibilities are among the following.

As we mentioned before, one of the most important drawbacks of BEM is the fact that the discretized matrices in (6.25) are dense and, therefore, computing the inverse of M is expensive ($O(n^3)$). However, windowing technique [82] [60], which is based on the Schur algorithm for approximate matrix inversion, can help to reduce the complexity. The Schur algorithm requires the matrix M to be known only partly, in a staircase (band) around the main diagonal. The band structure corresponds to interactions between closely coupled boundary elements. The approximate inversion then implicitly estimates the entries of the matrix inside the band structure, such that the resulting M^{-1} contains zeros outside of the band structure.

Another important issue is the use of special Green's function [82] which satisfies:

$$\frac{\partial G(p,q)}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \Gamma_1 \text{ and } \Gamma_2. \tag{6.30}$$

The use of such Green's function results into having to integrate only over the contact regions as in (6.21) and not over the whole boundary. This means that the substrate

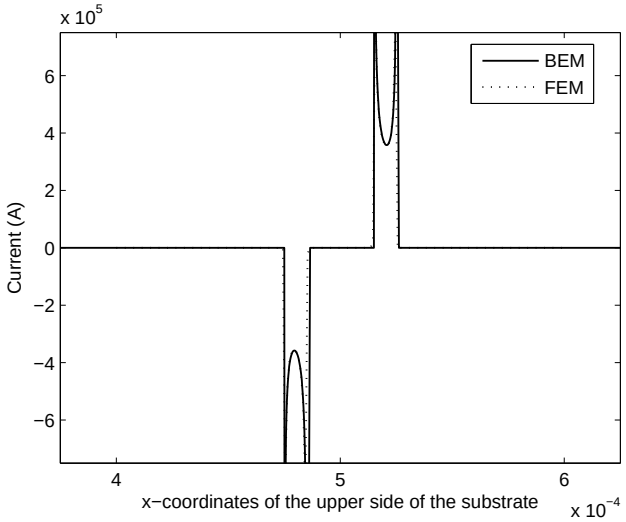

Figure 6.6: *Solutions for the current at the top boundary of the substrate by BEM and FEM.*

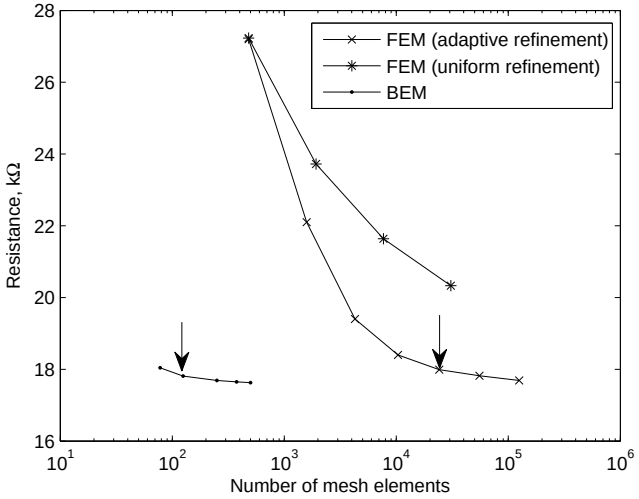

Figure 6.7: *Dependence between the resistance value and the number of mesh elements. Vertical arrows indicate the points in which a further refinement will provide a relative variation of less than 1% in the path resistance value.*

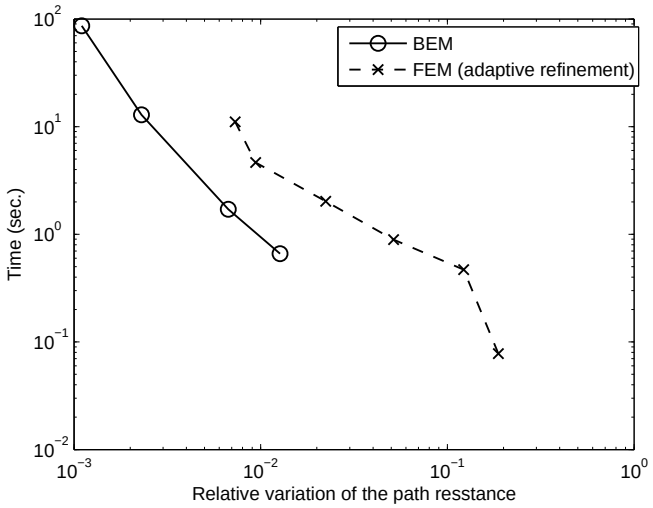

Figure 6.8: *Dependence between the relative variations of path resistance and time.*

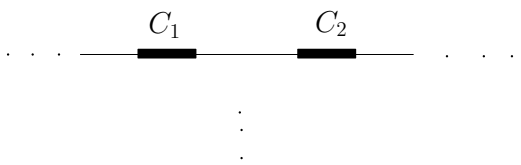

Figure 6.9: *Semi-infinite substrate with two contacts.*

modeling is performed in a semi-infinite domain as shown in Figure 6.9. In the next section we consider the extraction of a 3D homogeneous substrate, and we will show how the above techniques influence on a solution obtained by BEM in comparison with a solution obtained by FEM.

6.5 Modeling of 3D substrate by BEM and FEM

In this section we will compare BEM and FEM for the simulation of a 3D homogeneous substrate. We discuss features of modeling the substrate by existing tools based on BEM and FEM and investigate the convergence of both methods.

We consider a uniformly doped (10 S/m) substrate of size $1000 \times 1000 \times 380$ μm, which is presented in Figure 6.10. On the top layer it has two contacts of dimension 5×1.5 μm, and the distance between these contact is 60 μm. To extract the equivalent resistance between two contacts, we will use the following modeling tools:

- SPACE [61] - layout-to-circuit extractor with implementation of BEM,
- Comsol [21] - FEM-based multiphysics modeling tool.

6.5.1 Comparison between the methods

There are some main differences between BEM and FEM. The substrate modeling by BEM implemented in SPACE requires to discretize only the contact areas, while FEM discretizes the whole domain into tetrahedral elements. BEM assumes the domain to be a semi-infinite half-space, while FEM requires a finite domain. Therefore, FEM can approximate BEM by defining the FEM domain as large as possible [78].

As in the 2D case, the extraction of the 3D substrate with two contacts by BEM leads to a resistance network with 3 nodes: two nodes correspond to the contacts (C1 and C2) and one node corresponds to a reference node SUB. After elimination of the reference node, the network can be compared to the one obtained by FEM which does not contain the reference node.

In Comsol we have chosen a mesh with as much refinement as possible, given the available amount of memory in the machine on which Comsol runs. For SPACE we have used the nominal extraction settings.

Table 6.1 demonstrates that the computed path resistances by BEM and FEM networks are close. The remaining differences between BEM and FEM results are caused by the

Table 6.1: *Resistance values in kΩ extracted from the 3D substrate with two contacts. The bottom row indicates resistance between the contacts C1 and C2 after elimination of the reference node SUB*

	BEM	FEM
R(C1,C2)	854.54	33.81
R(C1,SUB)	15.12	-
R(C1,SUB)	15.12	-
	29.22	33.81

fact that path resistance by FEM has not been yet converged to the final path resistance, while the path resistance computed by BEM already converged. This fact will be clarified in the next two sections. One can think that for the considered 3D substrate, BEM does not approximate FEM well. This is not the case because the size of the contacts are small compared to the size of the substrate.

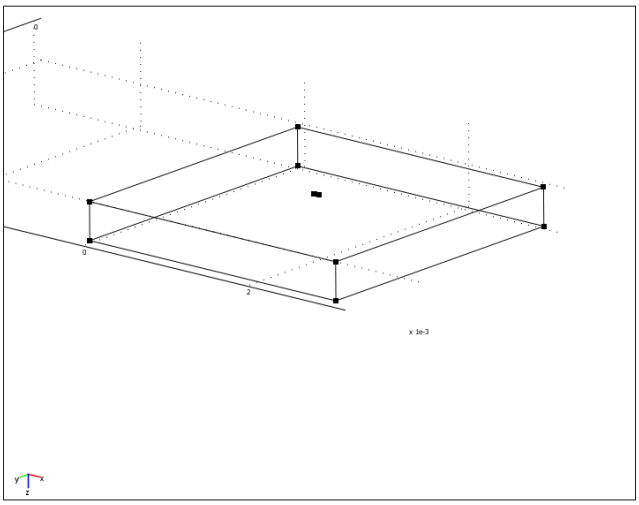

Figure 6.10: *Example of the substrate with two contacts. Image taken from the Comsol user interface.*

6.5.2 Convergence of FEM

This section presents a practical study of the convergence of FEM by varying the maximum mesh size at the contacts and the total number of mesh elements.

Table 6.2: *Statistics about solutions by FEM with adaptive mesh refinement*

# meshes at C1	112	136	358
# meshes (total)	28066	94753	178190
path resistance ($k\Omega$)	38.74	36.09	33.81
time (s)	6	73.6	218.4

We note that a fine discretization of the contact areas is required to compute accurately the extracted resistance. Indeed, the path resistance value, R, between the contacts depends on the value of the current flux, I, through the first contact and on the applied voltage, V, at the second contact. The resistance can be computed as

$$R = \frac{V}{I},$$

The current flux requires computing the integral over the boundary of the contact area, i.e., $I = \oint_\Gamma J_n dl$; therefore, the discretization plays an important role. The finer the discretization, the more accurate the approximation of the resistance value.

Table 6.2 shows that an adaptive mesh refinement decreases the resistance value. However, comparing the last value of the path resistance (33.81 $k\Omega$) with the previous one (36.09 $k\Omega$) shows that convergence has not been achieved yet. Further refinement was not possible due to the lack of memory. From this we conclude that the substrate extraction for large domains may require a lot of computer memory to make enough refinements.

6.5.3 Convergence of BEM

In this section we study the convergence of BEM through the parameters available in the layout-to-circuit extractor SPACE. For this we choose nominal settings, and vary only individual parameters, while keeping other parameters in the nominal settings. The nominal settings are

- number of BEM meshes per contact = 68,
- size of BEM window equals ∞, i.e., all entries of M^{-1} in (6.29) are taken into account.

Table 6.3: *Extraction statistics and resistance values for increasing refinement in the BEM mesh*

#BEM elements	time (s)	mem (Mb)	R ($k\Omega$)
4	2	0.12	30.77
8	2	0.13	30.76
16	2	0.18	30.45
68	2.1	1.05	29.22
296	4.3	16.6	29.01
332	5	20.85	29.00

Table 6.4: *Extraction statistics and resistance values for increasing size of the BEM window*

BEM window size (μm)	time (s)	mem (Mb)	R ($k\Omega$)
1	2.1	0.188	25.58
5	2.1	0.370	29.73
30	2.3	0.408	29.73
50	2.3	1.057	29.22
∞	2.3	1.057	29.22

BEM mesh

The initial BEM mesh consists of 68 BEM elements per contact. Table 6.3 demonstrates that refinement has very little influence on the resistance value between the contacts, while it is relatively costly with respect to the extraction time and memory usage.

Size of BEM window

SPACE allows sparsification through the windowing technique [78] [82], which can make the method more efficient at the cost of losing some accuracy. Table 6.4 shows that the window size has little influence on the resistance value. The larger the window size, the more memory is required. The differences in time are not noticeable due to the relatively small amount of contacts and meshes at each contact.

6.6 Concluding remarks

In this chapter we have considered the problem of homogeneous substrate modeling by the boundary element method (BEM) and the finite element method (FEM). Depending on the characteristics of the particular task, BEM and FEM techniques have advantages and disadvantages. BEM finds the solution on the boundary, and to know the solution at some internal points, BEM requires postprocessing. On the other hand, FEM finds solution at each discretization point of the domain. Therefore, if one is interested in behaviour of the whole domain FEM is more suitable for this task. For other problem BEM proves to be a much better alternative.

For instance, for extraction of a resistance network, which describes the behaviour of a homogeneous substrate, one requires to know the solution only at the contact areas. From this perspective, BEM is more attractive since solutions at the inner nodes are not required and, therefore, we spare time and do not capture details which are not necessary. In this sense BEM is a model order reduction technique on the operator level. Furthermore, there is another aspect to consider. Solving a 3D problem by FEM may be time and memory-consuming due to necessity of having fine discretization for obtaining grid independent results, while BEM is out of this problem.

As we saw in this chapter, BEM relies on the Green's function and requires discretization of the whole boundary. We have explored the usage of BEM for modeling of 2D and 3D substrates. A special kind of Green's function allows us to discretize only the contact areas, and not the whole boundary. In this case BEM approximates substrate as a semi-infinite half space. As a result, FEM solution at the contacts can approximate the BEM solution by making the FEM domain as large as possible.

BEM leads to dense matrices, which have the size of the number of elements at the boundary. Therefore, we also addressed the problem of BEM delivering dense matrices. One of the alternatives we studied was the use of windowing technique, which sparsifies the matrix by taking into account only influences between boundary elements that are relatively close to each other. We have observed that sparsification has little influence on the quality of extraction.

We note that the class of problems in which BEM can be applied is limited. For example, modeling of substrates involving layout-dependent doping patterns is more challenging than modeling of homogeneous substrates [79] [78]. In this case, finding analytical Green's function becomes extremely difficult, which restricts the kind of problems in which BEM can be applied efficiently. To overcome this problem, FEM or combined BEM/FEM methods can be considered. Nonetheless, BEM performs better than FEM, when applied to homogeneous substrate extraction.

Chapter 7

Industrial test case: simulation of power MOS transistors

In this chapter we present a few industrial problems related to modeling of MOS transistors. We suggest an efficient algorithm for computing output current at the top ports of the power MOS transistors for given voltage excitations. The suggested algorithm exploits the connection between the resistor and transistor networks and benefits from the sparsity of the conductance matrix. We also investigate a large resistor network, which is a part of the power MOS transistor model, and find out which existing reduction methods for resistor networks deliver significant reduction in the amount of resistors.

7.1 MOS transistor model

A power MOS transistor model [16] consists of a three-level architecture such as the one presented in Figure 7.1. On the top level two external contacts (top ports) are located. The body level contains a resistor network. On the bottom level the resistor network is connected to 768 bottom contacts (bottom ports). These bottom ports are the fingers of the non-linear transistors. Some of the bottom ports may simultaneously belong to two different transistors. Each transistor contains two bottom ports. The voltage difference between the transistor's ports is nonlinearly related to the current flowing through the transistor as shown in Figure 7.6.

7.2 Reduction of large resistor networks

We consider the resistor network, which is a part of the power MOS transistor model presented in Figure 7.1. The resistor network can be schematically subdivided into 5 layers which are connected to each other through resistors. The top layer contains 2 ports, and the bottom layer contains 768 ports that are shown in Figure 7.2. The layers, located between the top and the bottom, contain the internal nodes connected through the resistors.

To summarize, the network contains 110065 resistors, 41045 internal nodes, and 770 ports (external nodes). The aim is to reduce the resistor network up to the order of 10^4 resistors to reuse it efficiently.

As we saw in Chapter 3, reduction process of resistor networks by *ReduceR* [74] starts from identifying independent parts of the network, i.e., strongly connected components. If a strongly connected component (scc) has no ports, then it can be removed from the network. First, we have decided to perform a preprocessing step to analyze all scc's of the given network. The network contains 4 scc's: the first scc contains 28224 nodes, the second, the third scc contains 72 nodes each, and the forth scc contains 8 nodes. Since the last three scc's do not contain ports, we will not further take them into consideration.

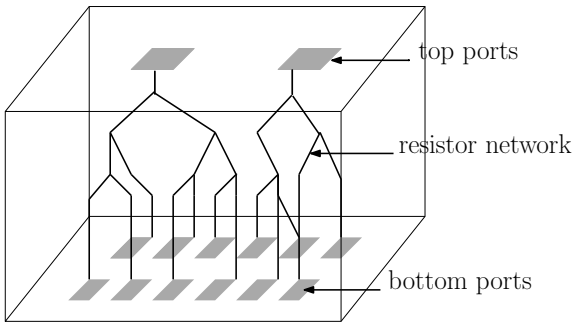

Figure 7.1: *Schematic representation of MOS transistor.*

In Chapter 5, we saw that simplification applied after reduction may lead to significant improvements in sparsity of conductance matrix in reasonable time. Following this result, we have applied *ReduceR* and simplification to the resistor network of the MOS transistor. Table 7.1 demonstrates that reduction by *ReduceR* alone and together with simplification by Err_{pa} with $\delta = 0.05$ (more details about this simplification can be found in Section 5.6.3) delivers resistor networks with the number of resistors of the order 10^5 which are still very large.

Table 7.1: *Results of reduction by the full elimination of internal nodes, by ReduceR (R), and by the simplification (S).*

	origin.	full elimin.	R	R+S	full elim.+S
#resistors	110413	148656	110065	109973	79484
int. nodes	41205	0	41045	41045	0
CPU time	-	9.5 s.	1844 s.	2196 s.	340 s.

An attractive feature is that the time required to perform full elimination is orders faster than to perform *ReduceR*. Figure 7.3 demonstrates the sparsity pattern of the conductance matrix obtained after full elimination of internal nodes. This is a not totally dense conductance matrix because for computing the Schur complement as

$$G_s = G_{11} - G_{12}G_{22}^{-1}G_{12}^T = G_{11} - QQ^T,$$

with $Q = L^{-1}G_{12}$, we fist have reordered G_{22} with AMD and then computed the Cholesky factor L, such that $G_{22} = LL^T$. As a result, the network contains 148656 resistors instead of 296065 resistors as it would happen without reordering. Applying simplifi-

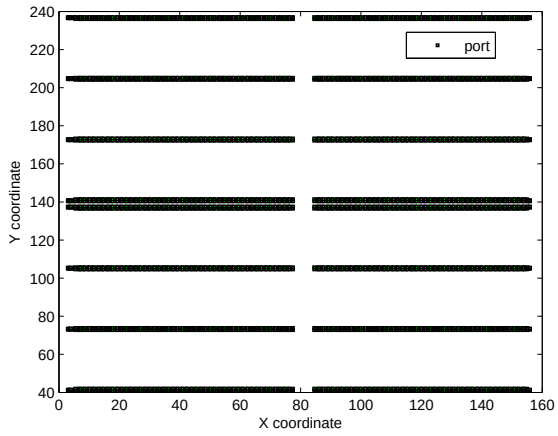

Figure 7.2: *Locations of the ports on the bottom layer.*

cation by Err_{pa} with $\delta = 5\%$ *after* elimination of all internal nodes, we obtain a reduced network with the number of elements of the order 10^4, see Table 7.1. The sparsity pattern of the new conductance matrix is shown in Figure 7.4 that is significantly sparser than the matrix pattern obtained after the elimination of all internal nodes presented in Figure 7.3.

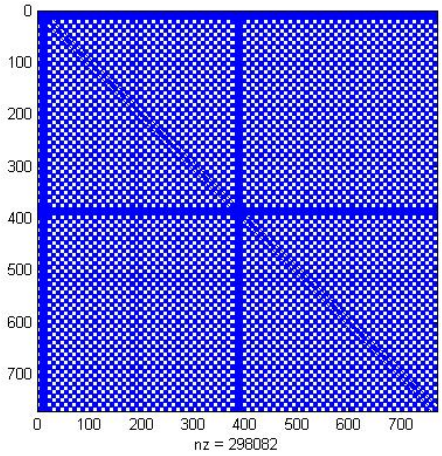

Figure 7.3: *The sparsity pattern of the conductance matrix obtained after elimination of all internal nodes.*

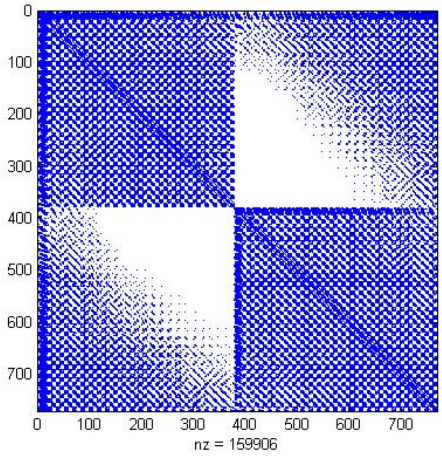

Figure 7.4: *The sparsity pattern of the conductance matrix obtained after elimination of all internal nodes and simplification.*

Figure 7.5 shows the quality of reduction: the path resistances of the reduced network differ from the path resistances of the original network by no more than a predefined tolerance $\delta = 0.05$.

We remind that this resistor network is the same as we used in the example of Section 5.4.2 to show that neglecting small entries of the conductance matrix was not sufficient to obtain a network of desired accuracy. The reader is referred to Table 5.1 which demonstrates that by dropping off-diagonal entries of the conductance matrix that are smaller than 0.0001% and 0.00001% of the corresponding diagonal terms, the maximum relative error between path resistances is larger than 5%. It follows that controlling error during simplification is a very important advantage of the simplification algorithms [87] [86] that we have developed. These algorithms allow to delete resistors and simultaneously to keep a strict control on the error due to approximation.

This example of reduction of a resistor network demonstrates that the order in which reduction methods are applied may significantly influence the quality of the reduced model.

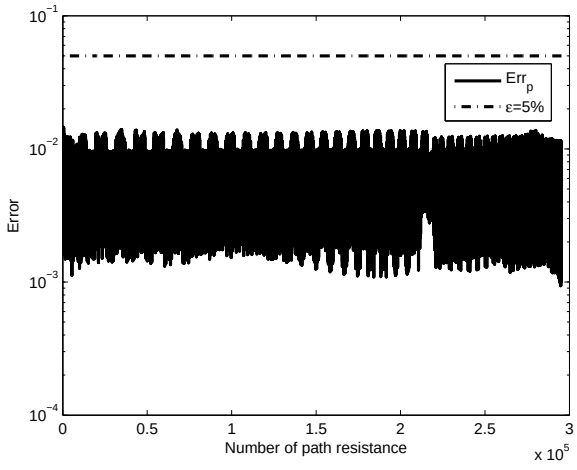

Figure 7.5: *Comparison of the computed error* $Err_p = \frac{|R-\widetilde{R}|}{|R|}$ *with* 5% *error for each path resistance.*

7.3 Simulation of MOS transistor

7.3.1 Modeling problem 1

The modeling problem is the following. Provided

- voltage excitation on the external top contacts ($\mathbf{v}_e$),
- nonlinear relation between the voltage difference at the transistor's ports and the current flowing through the transistors ($\mathbf{i}_T(\Delta\mathbf{v}_t)$),
- resistor network with conductance matrix G,

we have to find

- total current at the external top contacts ($\mathbf{i}_e$),
- voltages at the internal nodes, $\mathbf{v}_i$, of the resistor network and voltages at the bottom ports, $\mathbf{v}_t$.

Resistor network

In a matrix form, equation describing the resistor network of the power MOS transistor is

$$\begin{pmatrix} G_{11} & G_{12} & G_{13} \\ G_{12}^T & G_{22} & G_{23} \\ G_{13}^T & G_{23}^T & G_{33} \end{pmatrix} \begin{pmatrix} \mathbf{v}_e \\ \mathbf{v}_t \\ \mathbf{v}_i \end{pmatrix} = \begin{pmatrix} \mathbf{i}_e \\ \mathbf{i}_t \\ \mathbf{0} \end{pmatrix}, \tag{7.1}$$

where $G_{11} \in \mathbb{R}^{n_e \times n_e}$ denotes a block corresponding to the external contacts, $G_{22} \in \mathbb{R}^{n_t \times n_t}$ denotes a block corresponding to the bottom contacts, $G_{33} \in R^{n_i \times n_i}$ is a block corresponding to the internal nodes. The vector $\mathbf{i}_e$ denotes the currents injected into the external top contacts, and vector $\mathbf{i}_t$ denotes the currents flowing into the resistor network through the bottom ports. Note, there are no currents injected into internal nodes of the resistor network.

First we notice that once $\mathbf{v}_e$, $\mathbf{v}_t$, and $\mathbf{v}_i$ are known, $\mathbf{i}_e$ can be computed as follows

$$\mathbf{i}_e = G_{11}\mathbf{v}_e + G_{12}\mathbf{v}_t + G_{13}\mathbf{v}_i. \tag{7.2}$$

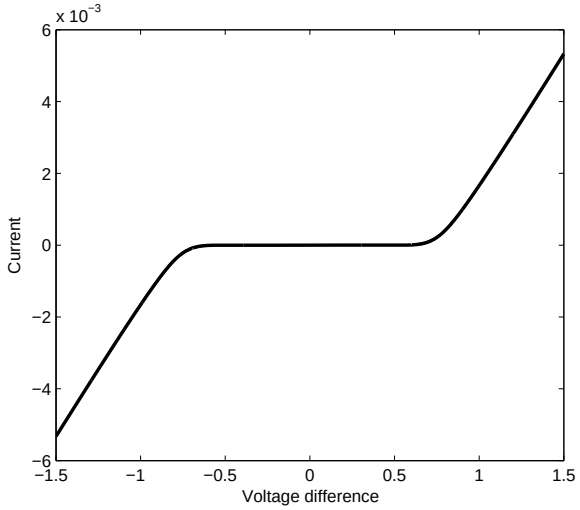

Figure 7.6: *Nonlinear relation between voltage difference at the transistor's ports and current flowing through the transistor.*

To find $\mathbf{v}_t$ and $\mathbf{v}_i$, first we separate all known and unknown variables in (7.1) which leads to the following system

$$\begin{pmatrix} G_{22} & G_{23} \\ G_{23}^T & G_{33} \end{pmatrix} \begin{pmatrix} \mathbf{v}_t \\ \mathbf{v}_i \end{pmatrix} = \begin{pmatrix} \mathbf{i}_t \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} G_{21}\mathbf{v}_e \\ G_{31}\mathbf{v}_e \end{pmatrix}, \tag{7.3}$$

where $i_t = f(\mathbf{v}_t)$ is a function representing the currents going into the resistor network from the transistor network.

Transistor network

Similar to the resistor network, we define an incidence matrix for the transistor network as $P \in \mathbb{R}^{n_{tr} \times n_t}$, where n_{tr} is the number of transistors connected to the bottom ports of the MOS transistor, and n_t is the number of bottom ports.

For example, for two transistors defined between three nodes as in Figure 7.7, the incidence matrix is defined as

$$P = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}.$$

Thus, we can define the vector of currents, $\mathbf{i}_T$, which flows through all the transistors as follows

$$\mathbf{i}_T = f(P\mathbf{v}_t).$$

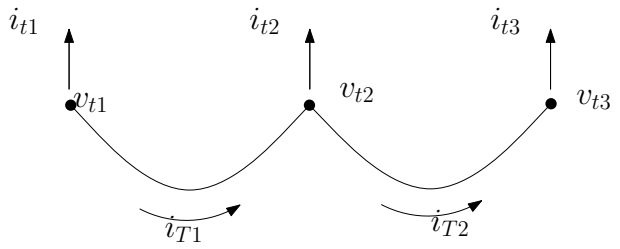

Figure 7.7: *Schematic representation of two transistors.*

On the other hand, $\mathbf{i}_t$ represents the communication between the resistor network and the transistor network. By analyzing the transistor network and keeping the orientation of $\mathbf{i}_t$, we realize that $\mathbf{i}_t$, i.e., current flowing from the transistor network into the resistor network, can be written as

$$\mathbf{i}_t = -P^T f(P\mathbf{v}_t). \tag{7.4}$$

In other words, the vector $\mathbf{i}_t$ is determined by the vector $\mathbf{i}_T$, which can be computed by evaluating f from the table containing the nonlinear relation between $\mathbf{v}_t$ and $\mathbf{i}_T$. Thus (7.3) can be rewritten as

$$\underbrace{\begin{pmatrix} G_{22} & G_{23} \\ G_{23}^T & G_{33} \end{pmatrix}}_{G} \underbrace{\begin{pmatrix} \mathbf{v}_t \\ \mathbf{v}_i \end{pmatrix}}_{\mathbf{v}} = \underbrace{\begin{pmatrix} -P^T f(P\mathbf{v}_t) \\ \mathbf{0} \end{pmatrix}}_{\mathbf{b}_1} - \underbrace{\begin{pmatrix} G_{21}\mathbf{v}_e \\ G_{31}\mathbf{v}_e \end{pmatrix}}_{\mathbf{b}_2}. \tag{7.5}$$

This is a nonlinear equation with respect to $\mathbf{v}_t$. Since we do not count on an explicit expression for f, we cannot use derivative-based solvers such as Newton's method. Nevertheless, the equation can be solved, for instance, by Broyden's method [40]. However, there exists another alternative. We observe that it is better to keep the linear and the nonlinear parts separate from each other in such a way that we could benefit from the computation of the Cholesky decomposition of the linear system. Therefore, we will proceed in an iterative manner, requiring only solutions of the factorized systems and evaluations of the nonlinear function f. This method is some sort of block Gauss-Seidel algorithm.

To take full advantage of the sparsity pattern of G, we applied the reordering strategy (AMD [7]) before solving the nonlinear system (7.5). The steps to solve the nonlinear system are described in the suggested Algorithm 6. Computing $\mathbf{v} = \begin{pmatrix} \mathbf{v}_t & \mathbf{v}_i \end{pmatrix}^T$ at each iteration requires computing $\mathbf{i}_t$ based on (7.4). The algorithm stops when the residual is smaller than a prescribed tolerance.

Algorithm 6 Algorithm to solve nonlinear system (7.5)

INPUT: G, $\mathbf{b}_2$, P, table which describes f, tolerance ϵ
OUTPUT: $\mathbf{v}$
1. Reorder G with AMD;
2. Compute Cholesky factorization $G = LL^T$;
3. Set initial guess $\mathbf{i}_t^{(1)}$;
4. **for** $i = 1, \ldots$
5. Define $\mathbf{b}_1 = \begin{pmatrix} \mathbf{i}_t^{(i)} & 0 \end{pmatrix}^T$;
6. Solve $G\mathbf{v}^{(i)} = (\mathbf{b}_1 - \mathbf{b}_2)$ for $\mathbf{v}^{(i)}$
7. Compute residual $\mathbf{r}^{(i)} = \mathbf{i}_t^{(i)} + P^T f(P\mathbf{v}_t^{(i)})$;
8. **if** $||\mathbf{r}^{(i)}|| \leq \epsilon$ **then** stop **else**
9. $\mathbf{i}_t^{(i+1)} = -P^T f(P\mathbf{v}_t^{(0)})$;
10. go to step 4;
11. **endif**
12. **endfor**

The advantage of this approach, is that the Cholesky factorization is performed only once. Therefore, once we have solved the system for a fixed input, computing the solution for a different input equires only solvng two lower-triangular systems and evaluations of the nonlinear function f.

Since the data for the nonlinear function f, presented in Figure 7.6, are provided in the form of a table, computing $f(a)$, where a is a vector, can be performed as follows. For each element a_i one has to evaluate its location. If $a_i \in (-1.5; 1.5)$, then $f(a_i)$ is computed by the interpolation between the closest neighboring points. If $a_i > 1.5$ or $a_i < 1.5$, then $f(a_i)$ is computed by extrapolating corresponding linear parts of the curve.

In our experiments, the algorithm usually converges in less than 5 iterations. But this could take longer, if the function f would have stronger nonlinearities.

Example

Figure 7.8 demonstrates how the difference in voltages, $\mathbf{v}_{e,1} - \mathbf{v}_{e,2}$, at the top contacts influences the total current flowing out of the left contact. It can be seen that the current in the range $(0.5, 1.5)$ is nonlinear which is expected due to the nonlinear behavior of transistors in the same range. The time spent on solving the nonlinear system (7.5) for 400 different inputs, $\mathbf{v}_e$, is 401 sec. in a PC with a processor 1.6 GHz and with 2MB RAM.

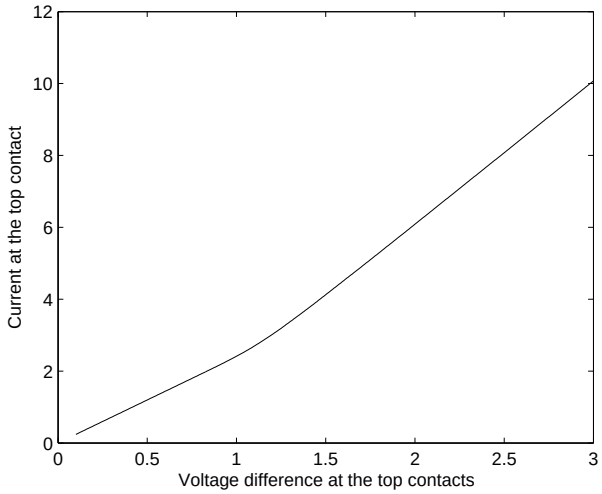

Figure 7.8: *Dependence between the voltage difference at the top contacts and the current flowing through one of the top contacts.*

7.3.2 Modeling problem 2

If, in a particular case, we are not interested in finding $\mathbf{v}_i$, then we deal with a similar modeling problem as in Section 7.3.1. The only difference is that in this case we have to find

- total current at the external top contacts ($\mathbf{i}_e$),
- voltages at the bottom ports ($\mathbf{v}_t$).

There are a few alternatives to solve this task. The first alternative is to solve the system (7.5) by Algorithm 6, though $\mathbf{v}_i$ is not needed any more. The second alternative is first to eliminate all internal nodes and then to solve the corresponding system by using Algorithm 6. The main difference between these approaches is the size of the matrix G (number of equations) and its sparsity. In the first case G is large and sparse, in the second case G is smaller and more dense.

We will consider the second approach. Eliminating $\mathbf{v}_i$ from (7.1) leads to the system of the form

$$\begin{pmatrix} \widetilde{G}_{11} & \widetilde{G}_{12} \\ \widetilde{G}_{12}^T & \widetilde{G}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{v}_e \\ \mathbf{v}_t \end{pmatrix} = \begin{pmatrix} \mathbf{i}_e \\ \mathbf{i}_t \end{pmatrix}. \tag{7.6}$$

Table 7.2: *Comparison of the systems (7.5) and (7.7)*

	system (7.5)	system (7.7)	system (7.7) with simplified $\widetilde{G}_{22}$
size	28222	768	768
# resistors	110413	148656	79484
CPU time (s.)	401	297	267
preprocessing time (s.)	-	9.5	340

Thus, $\mathbf{v}_t$ can be found by solving the following system:

$$\underbrace{\widetilde{G}_{22}}_{G}\underbrace{\mathbf{v}_t}_{\mathbf{v}} = \underbrace{\mathbf{i}_t}_{\mathbf{b}_1} - \underbrace{\widetilde{G}_{12}^T\mathbf{v}_e}_{\mathbf{b}_2}, \tag{7.7}$$

where $\widetilde{G}_{22} \in \mathbb{R}^{n_t \times n_t}$ has a sparsity pattern similar to the one in Figure 7.3. Note, that $\widetilde{G}_{22}$ has smaller dimension but larger fill-in than $G \in \mathbb{R}^{(n_t+n_i)\times(n_t+n_i)}$ in (7.5). Nevertheless, we expect that solving (7.7) by Algorithm 6 will take less time than solving (7.5). As soon as $\mathbf{v}_t$ has been found, $\mathbf{i}_e$ can be easily obtained from (7.6).

Table 7.2 shows a comparison between the time costs of solving the systems (7.5) and (7.7). The table includes the extra time required for preprocessing of G. As expected, even though the system (7.7) is less sparse than the one in (7.5), solving the system (7.7) is faster than solving the system (7.5) because the second system has smaller size. In this case, sparsity does not play a very important role. Table (7.7) also shows the results of solving (7.7), when $\widetilde{G}_{22}$ has been simplified beforehand by Err_{pa} with $\delta = 5\%$. It can be seen that the result looks less profitable due to the preprocessing time which is needed for simplification. Nevertheless, preprocessing can be useful if one has to solve (7.7) several times (for more than thousand different inputs $\mathbf{v}_e$, in this particular case). This may happen, for instance, if a power MOS transistor model is to be reused in other simulations.

7.4 Concluding remarks

In this chapter we have considered a large resistor network of a power MOS transistor. Reduction of this network by the method developed in Chapter 5 combined with full elimination of internal nodes showed to be much more successful, than reduction by *ReduceR*, or by only full elimination of the internal nodes. After that, we have developed a strategy to compute the behaviour of the power MOS transistor. It includes

computing the current, which flows out of the top contacts, for given input voltages at the top contacts. A key observation here is the construction of an incidence matrix, which connects the resistor and the transistor networks. The solution is found in an iterative manner, requiring only solutions of the factorized linear systems and evaluations of a nonlinear function. This strategy can also be applied for modeling MOS transistors with larger amount of top contacts, and it is expected to be very helpful when the dependence between the voltage difference and the current at the transistor contacts is highly nonlinear.

Chapter 8

Conclusions

In this thesis we addressed a set of challenging industrial problems originating from modeling of integrated circuits. The main results of this thesis are following:

- In Chapter 2, we discussed existing methods used for modeling of interconnect and substrate. In particular, we have reviewed a method for the modeling of interconnect structures used in the tool Fasterix. We also presented a mathematical formulation for the modeling of a homogeneous substrate. Since result of interconnect and substrate modeling are circuits consisting of resistors, inductors and capacitors, we have included and discussed the main properties of the circuit equations such as stability, passivity and important types of circuit analysis.

- In Chapter 3, we briefly discussed the goals and popular existing methods of model order reduction. Then, we concentrated on the reduction method used in the tool Fasterix and methods for reduction of large resistor networks which are important ingredients for the methods developed in this thesis. It was shown that preservation of sparsity is a crucial condition for the reduction of many-terminal networks.

- Chapter 4 contains a detailed analysis of the super node algorithm (reduction method used in Fasterix). We have proved that the admittance matrix of the reduced circuit is not positive real, which causes the super node algorithm to deliver non-passive circuits. To overcome this problem, we have included a passivity enforcement technique based on quadratic programming in the super node algorithm. The modified version of the super node algorithm always delivers passive reduced circuits, while the extra computational time is moderate.

- Challenges in the reduction of resistor networks are explained in Chapter 5. It

was shown that reduction methods based on elimination of internal nodes in the resistor networks may deliver a poor reduction.

First, we have proved that if an original network contains positive resistances, then the corresponding reduced network, obtained from the original one by eliminating any amount of internal nodes, consists of positive resistances as well. This property implies that the reduced network is passive.

Then, we have suggested a novel method to reduce the amount of resistors in large resistor networks. The idea of our approach is to improve the sparsity of the conductance matrix by replacing the original network by an approximate one with less resistors, while keeping the error within some given margin. The key of the approach is the analytical derivation of error bounds, which with the adequate algorithms can be computed efficiently. Error bounds have been derived for the relative error of voltages and the maximum relative error of path resistances. In several numerical experiments, the proposed technique performs very well and appears very promising, especially for the case of multi-terminal resistor networks. The approach is versatile in the sense that it can be used in combination with already existing reduction techniques, for obtaining better reduced models.

- In Chapter 6, we considered extraction of resistor network from a homogeneous substrate. This problem requires solving the Laplace equation with appropriate boundary conditions. In this chapter we have studied two discretization methods to solve this equation, namely the finite element method (FEM) and the boundary element method (BEM). We have noted that BEM can be seen as a model order reduction technique at the level of operator because the Green's function makes possible to discretize only the contact areas instead of having to discretize the whole substrate which is necessary for FEM. Since BEM usually leads to dense matrices, we also considered opportunity of sparsification by taking into account only influences between boundary elements that are relatively close to each other. It was observed that sparsification has little influence on the quality of extraction. We conclude that modeling of homogeneous substrate by BEM is significantly faster because it reaches mesh-independent results in cases where FEM can not.

- Chapter 7 contains a challenging industrial example related to the simulation of a power MOS transistor. First, we have reduced the large resistor network of a power MOS transistor by full elimination of internal nodes combined with the method developed in Chapter 5. At the cost of some accuracy, the result of reduction in this case is much more successful, than by applying existing reduction methods for resistor networks. Secondly, we have developed a strategy to model the power MOS transistor. This strategy can be applied for modeling MOS transistors with larger amount of top contacts, and it is expected to be even more helpful when the dependence between the voltage difference and the current at the transistor contacts is highly nonlinear.

8.1 Suggestions for future work

Nowadays, there exist a vast variety of model order reduction methods to reduce large RLC circuits, which often represent the interconnect. It is of great importance to have well-based efficient methods which preserve stability, passivity, and sparsity. The sparsity condition becomes important because real life interconnects are multy-terminal structures. Therefore, sparsity preserving methods are very likely to be a good candidate for future generation of reduction methods.

As we discussed, the substrate is often modeled by large resistor networks. Therefore, it is often required to have reduced models which are exact or accurate enough within a given margin. It is, therefore, desirable to have efficient methods for reduction of resistor networks, which are able to reduce not only the amount of internal nodes but also the number of resistors (i.e. the sparsity of the conductance matrix). With this goal in mind, we have suggested simplification algorithms, which are aimed at reducing the amount of resistors. In order to warranty the quality of the solution, we have created analytical error estimations. Nonetheless, there is room for improvement.

Simplification of resistor networks based on the relative error between voltages at external nodes is more attractive than the relative error between voltages at external *and* internal nodes because only the external nodes are accessed by designers. Therefore, deriving a computationally efficient estimation for this error is desirable and can be considered as a future direction.

Simplification based on merging nodes between relatively small resistances can be the next step towards decreasing the number of resistors in the network. Since varying the number of nodes changes the dimension of conductance matrix, not all considered criteria to simplify networks can be used. For instance, estimations for the error, $\frac{||\mathbf{v}-\tilde{\mathbf{v}}||}{||\mathbf{v}||}$, where ,$\mathbf{v}$ denotes the vector of voltages at all nodes of the original network, and $\widetilde{\mathbf{v}}$ denotes the vector of voltages at all nodes after simplification, cannot be used. Another possibility would be to study the performance of the simplification procedures developed in this thesis, when applied to large networks obtained by BEM, and also to consider the possibility of extending these methods for simplification of RC, RL, and RLC networks.

Appendix A

Appendices for Chapter 4

A.1 Computation of Y_R, Y_L, Y_G and Y_C for high frequency range

For this goal quantities $\mathbf{I}_0$, $\mathbf{V}_0$, $\mathbf{I}_1$ and $\mathbf{V}_1$ can be presented through the frequency independent quantities $\mathbf{I}_{00}$, $\mathbf{V}_{00}$, $\mathbf{I}_{01}$, $\mathbf{V}_{01}$, $\mathbf{I}_{10}$, $\mathbf{V}_{10}$, $\mathbf{I}_{11}$ and $\mathbf{V}_{11}$ as follows [20] [93]:

$$\mathbf{I}_0 = \frac{1}{-i\omega}\mathbf{I}_{00} + \frac{1}{(-i\omega)^2}\mathbf{I}_{01}, \tag{A.1}$$

$$\mathbf{V}_0 = \mathbf{V}_{00}^n + \frac{1}{-i\omega}\mathbf{V}_{01}, \tag{A.2}$$

$$\mathbf{I}_1 = -i\omega\mathbf{I}_{10}^n + \mathbf{I}_{11}, \tag{A.3}$$

$$\mathbf{V}_1 = (-i\omega)^2\mathbf{V}_{10}^n + (-i\omega)\mathbf{V}_{11}, \tag{A.4}$$

The pairs $(\mathbf{I}_{00}, \mathbf{V}_{00})$, $(\mathbf{I}_{01}, \mathbf{V}_{01})$, $(\mathbf{I}_{10}, \mathbf{V}_{10})$, $(\mathbf{I}_{11}, \mathbf{V}_{11})$ can be found from the four sets of equations (A.5)–(A.8). To get these equations we should substitute (A.1)–(A.4) into (4.75)–(4.78) and gather terms with s in the same power. In order to obtain $\mathbf{I}_{00}$ and $\mathbf{V}_{00}$, terms with s^0 and $\frac{1}{s}$ are gathered:

$$L\mathbf{I}_{00}^n - P_{N'}\tilde{\mathbf{V}}_{00}^n = P_N\mathbf{V}_N \tag{A.5}$$

$$-P_{N'}^T\tilde{\mathbf{I}}_{00}^n = 0$$

In order to obtain $\mathbf{I}_{01}$ and $\mathbf{V}_{01}$, terms with $\frac{1}{s}$ and $\frac{1}{s^2}$ are gathered:

$$L\mathbf{I}_{01}^{n} - P_{N'}\mathbf{V}_{01}^{n} = -R\mathbf{I}_{00} \tag{A.6}$$
$$-P_{N'}^{T}\tilde{\mathbf{I}}_{01}^{n} = 0$$

In order to obtain $\mathbf{I}_{10}$ and $\mathbf{V}_{10}$, terms with s^2 and s are gathered:

$$L\mathbf{I}_{10} - P_{N'}\mathbf{V}_{10} = 0 \tag{A.7}$$
$$-P_{N'}^{T}\mathbf{I}_{10}^{n} = C_{N'N'}\mathbf{V}_{00} + C_{N'N}\mathbf{V}_{N}$$

In order to obtain $\mathbf{I}_{11}$ and $\mathbf{V}_{11}$, terms with s^2 and s are gathered:

$$L\mathbf{I}_{11} - P_{N'}\mathbf{V}_{11} = -R\mathbf{I}_{10} \tag{A.8}$$
$$-P_{N'}^{T}\mathbf{I}_{11}^{n} = C_{N'N'}\mathbf{V}_{01}.$$

Substituting (A.1)–(A.4) into $\tilde{Y}_1(s)$, one obtains (4.80):

$$Y_3(s) = (s)^{-2}Y_R + (s)^{-1}Y_L + Y_G + (s)Y_C + O((ik_0h)^2), \tag{A.9}$$

where

$$\begin{aligned} Y_R &= P_N^T\mathbf{I}_{01}, \\ Y_L &= P_N^T\mathbf{I}_{00}, \\ Y_G &= C_{NN'}^T\mathbf{V}_{01} + P_N^T\mathbf{I}_{11}, \\ Y_C &= C_{NN'}^T\mathbf{V}_{00} + C_{NN}\mathbf{V}_N + P_N^T\mathbf{I}_{10}. \end{aligned}$$

The error due to approximation yields

$$Y_3(s) - Y_1(s) = O\left((ik_0h)^2\right). \tag{A.10}$$

A.2 Construction of the matrix M

We consider a rational approximation of the form

$$Y(s,\mathbf{x}) = \sum_{m=1}^{n} \frac{c_m}{s - a_m} + se, \tag{A.11}$$

where $\mathbf{x}$ is a vector which includes the residues c_m. Let vector $\mathbf{y}(s,\mathbf{x})$ contains columns of $Y(s,\mathbf{x})$. Denoting differentials of $\mathbf{x}$ and $\mathbf{y}(s,\mathbf{x})$ as $\Delta\mathbf{x}$, $\Delta\mathbf{y}(s,\mathbf{x})$, one can define the

following incremental relation

$$\Delta\mathbf{y}(s,\mathbf{x}) = M\Delta\mathbf{x}. \tag{A.12}$$

To show the construction of matrix M, let $Y(s,\mathbf{x})$ be 2×2 matrix and $n=2$. In this case M is constructed as

$$M(s) = \begin{pmatrix} \frac{1}{s-a_{1,11}} & \frac{1}{s-a_{1,12}} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{s-a_{2,11}} & \frac{1}{s-a_{2,12}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{s-a_{1,21}} & \frac{1}{s-a_{1,22}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{s-a_{2,21}} & \frac{1}{s-a_{2,22}} \end{pmatrix}.$$

A.3 Construction of the matrix F

Let $\mathbf{g}$ contains the columns of $Re(Y(s,\mathbf{x}))$, and let λ denote a vector of eigenvalues of $Re(Y(s,\mathbf{x}))$. We shall consider $\Delta\mathbf{g}$, $\Delta\mathbf{x}$ and $\Delta\lambda$ as differentials. Thus the following linear relations can be defined

$$\Delta\mathbf{g} = Re(M)\Delta\mathbf{x}, \tag{A.13}$$

$$\Delta\lambda = Q\Delta\mathbf{g}, \tag{A.14}$$

where the matrix M is defined in Appendix A.2, and matrix Q is defined below. Note, that (A.13) can be obtained from (A.12) by taking the real part. Combining (A.13) and (A.14), one obtains the following expression for matrix F:

$$\Delta\lambda = \underbrace{QRe(M)}_{F}\Delta\mathbf{x}. \tag{A.15}$$

In order to define F we have to construct Q. Below we show how to do it. Let G denote $Re(Y(s,\mathbf{x}))$, and suppose that all eigenvalues of G are distinct. We consider the eigenvalue problem of G for a particular eigenvalue λ_i and its corresponding eigenvector $\mathbf{v}_i$:

$$G\mathbf{v}_i = \lambda_i\mathbf{v}_i. \tag{A.16}$$

We shall consider ΔG, $\Delta\lambda_i$ and $\Delta\mathbf{v}_i$ as differentials, therefore, according to [25] (p. 229)

$$(\Delta G)\mathbf{v}_i + G(\Delta\mathbf{v}_i) = (\Delta\lambda_i)\mathbf{v}_i + \lambda_i(\Delta\mathbf{v}_i). \tag{A.17}$$

Multiplying (A.17) with the corresponding left (row) eigenvector $\mathbf{w}_i$ of G, and taking into account that $\mathbf{w}_i G = \lambda\mathbf{w}_i$, we obtain

$$\Delta\lambda_i = \frac{\mathbf{w}_i\Delta G\mathbf{v}_i}{\mathbf{w}_i\mathbf{v}_i}. \tag{A.18}$$

Since G is symmetric, $\mathbf{w}_i = \mathbf{v}_i$. When the eigenvectors have been normalized to unit length we get

$$\Delta\lambda_i = \mathbf{v}_i^T \Delta G \mathbf{v}_i. \tag{A.19}$$

Let ΔG consist of column vectors $\mathbf{g}_i$, i.e., $\Delta G = \begin{pmatrix} \mathbf{g}_1 & \mathbf{g}_2 & \dots & \mathbf{g}_N \end{pmatrix}$. We remind that $\Delta\mathbf{g}$ contains columns of ΔG, i.e. $\Delta\mathbf{g} = \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_N \end{pmatrix}$. Thus, the vector $\Delta\lambda$ can be constructed through (A.19) as follows

$$\Delta\lambda = \begin{pmatrix} \mathbf{v}_1 \Delta G \mathbf{v}_1 \\ \mathbf{v}_2 \Delta G \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_N \Delta G \mathbf{v}_N \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^N \mathbf{v}_1^T \mathbf{g}_k (\mathbf{v}_1)_k \\ \sum_{k=1}^N \mathbf{v}_2^T \mathbf{g}_k (\mathbf{v}_2)_k \\ \vdots \\ \sum_{k=1}^N \mathbf{v}_N^T \mathbf{g}_k (\mathbf{v}_N)_k \end{pmatrix} \tag{A.20}$$

$$= \begin{pmatrix} \mathbf{v}_1^T(\mathbf{v}_1)_1 & \mathbf{v}_1^T(\mathbf{v}_1)_2 & \dots & \mathbf{v}_1^T(\mathbf{v}_1)_N \\ \mathbf{v}_2^T(\mathbf{v}_2)_1 & \mathbf{v}_2^T(\mathbf{v}_2)_2 & \dots & \mathbf{v}_2^T(\mathbf{v}_2)_N \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{v}_N^T(\mathbf{v}_N)_1 & \mathbf{v}_N^T(\mathbf{v}_N)_2 & \dots & \mathbf{v}_N^T(\mathbf{v}_N)_N \end{pmatrix} \begin{pmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_N \end{pmatrix} = Q\Delta\mathbf{g}. \tag{A.21}$$

Thus, the elements of the matrix Q are defined through the normalized eigenvectors of $Re(Y(s, \mathbf{x}))$. For a fixed frequency s, the resulting matrix F can be computed as

$$F(s) = Q(s) Re(M(s)), \tag{A.22}$$

where $M(s)$ is described in Appendix A.2.

Appendix B

Appendices for Chapter 5

B.1 Theorems related to M-matrices

Theorem 9. *If A_{11} is a principal submatrix of A, then $\rho(A_{11}) \leq \rho(A)$, where $\rho(\cdot)$ stands for the spectral radius of a matrix.*

Theorem 10. *Let $A \in Z_n$, and let us write A as $A = \alpha I - P$ with $\alpha \in \mathbb{R}$ and $P \succ 0$, where "$\succ$" applies element-wise, i.e. all the elements of P are non-negative. Then, A is an M-matrix if and only if $\alpha \geq \rho(P)$.*

The following Theorem 11 follows from the two previous theorems.

Theorem 11. *Let A be an M-matrix. Then, each principal submatrix of A is also an M-matrix.*

Proof. Since A is an M-matrix, by Theorem 2 we can write $A = \alpha I - P$, where $P \succ 0$ and $\alpha \geq \rho(P)$. Since A_{11} is a principal submatrix of A, one can write $A_{11} = \alpha I - P_{11}$, where P_{11} is the correspondent principal submatrix of P. Applying Theorem 1, one obtains that $\alpha \geq \rho(P) \geq \rho(P_{11})$. By Theorem 2, we conclude that A_{11} is an M-matrix. □

Theorem 12. *Provided that $A \in Z_n$, the following statements are equivalent:*

- *A is positive stable, that is, A is an M-matrix.*
- *A is nonsingular and $A^{-1} \succ 0$.*

Proofs of Theorem 9, Theorem 10, and Theorem 12 can be found in [47].

B.2 Theorem related to perturbation theory

Let A be a Hermitian matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, and let $\Delta A = A + E$ be a Hermitian perturbation of A with eigenvalues $\widetilde{\lambda}_1 \geq \widetilde{\lambda}_2 \geq \cdots \geq \widetilde{\lambda}_n$.

Theorem 13. *Let the eigenvalues of E be*

$$\epsilon_1 \geq \epsilon_2 \geq \cdots \geq \epsilon_n, \tag{B.1}$$

and let $i_1, \ldots i_k$ be distinct integers between 1 and n inclusive. Then

$$\lambda_{i_1} + \ldots \lambda_{i_k} + \epsilon_{n-k+1} + \cdots + \epsilon_n \leq \widetilde{\lambda}_{i1} + \cdots + \widetilde{\lambda}_{ik} \tag{B.2}$$

$$\leq \lambda_{i_1} + \ldots \lambda_{i_k} + \epsilon_1 + \cdots + \epsilon_k. \tag{B.3}$$

Corollary 1

$$\widetilde{\lambda}_i \in \left[\lambda_i + \epsilon_n, \lambda_i + \epsilon_1\right], \quad \text{for} \quad i = 1, \ldots, n. \tag{B.4}$$

Corollary 2

$$\max\left\{|\widetilde{\lambda}_i - \lambda_i|\right\} \leq ||E||_2, \quad \text{for} \quad i = 1, \ldots, n. \tag{B.5}$$

Note that $||E||_2 = \max\{|\epsilon_1|, |\epsilon_n|\}$. Proof of the theorem and corollaries can be found in [83].

B.3 Generalization of the error estimation Err_{ves}

Err_{ves} in (5.56) is the upper bound of the relative error Err_{ve} after the first resistor has been deleted. We will generalize Err_{ves} for the case when k resistors are deleted one by one from the network. Deleting the second resistor $1/g_2$ form the block G_{22} leads to

$$\widetilde{\widetilde{G}}_{22} = \widetilde{G}_{22} - \mathbf{m}_2 g_2 \mathbf{m}_2^T. \tag{B.6}$$

Applying the Sherman-Morrison formula [40], one obtains

$$\widetilde{\widetilde{G}}_{22}^{-1} = \widetilde{G}_{22}^{-1} - \mathbf{h}_2 c_2 \mathbf{h}_2^T, \tag{B.7}$$

where $\mathbf{h}_2 = \widetilde{G}_{22}^{-1}\mathbf{m}_2$, and c_2 is defined similarly as in (5.55). After deleting the second resistor, the error bound (6.18) becomes

$$||G_s^{-1} - \widetilde{\widetilde{G}}_s^{-1}|| \cdot ||G_s|| = ||G_s^{-1} - \left(G_{11} - G_{12}\left(\widetilde{G}_{22}^{-1} - \mathbf{h}_2 c_2 \mathbf{h}_2^T\right) G_{12}^T\right)^{-1}|| \cdot ||G_s|| \tag{B.8}$$

$$= ||G_s^{-1} - \widetilde{G}_s^{-1} + \widetilde{G}_s^{-1} G_{12} \mathbf{h}_2 \widetilde{c}_2 \mathbf{h}_2^T G_{12}^T \widetilde{G}_s^{-1}|| \cdot ||G_s||, \tag{B.9}$$

where

$$\widetilde{G}_s = G_{11} - G_{12} \widetilde{G}_{22}^{-1} G_{12}^T, \tag{B.10}$$

$$\widetilde{c}_2 = \left(\frac{1}{c_2} + \mathbf{h}_2^T G_{12}^T \widetilde{G}_s^{-1} G_{12} \mathbf{h}_2 \right)^{-1}. \tag{B.11}$$

At this point we note that $G_s^{-1} - \widetilde{G}_s^{-1}$ in (B.9) comes from the previous iteration, i.e., when the first resistor was deleted. This helps us to generalize Err_{ves} in (5.51) for the case when k resistors have been deleted one by one:

$$Err_{ves} = || \sum_{j=1}^{k} G_{s,(j-1)}^{-1} G_{12} \mathbf{h}_j \widetilde{c}_j \mathbf{h}_j^T G_{12}^T G_{s,(j-1)}^{-1} || \cdot ||G_s||, \tag{B.12}$$

where

$$\mathbf{h}_j = G_{22,(j-1)}^{-1} \mathbf{m}_j, \tag{B.13}$$

$$\widetilde{c}_j = \left(\frac{1}{c_j} + \mathbf{h}_j^T G_{12}^T G_{s,(j-1)}^{-1} G_{12} \mathbf{h}_j \right)^{-1}, \tag{B.14}$$

$$c_j = \left(\frac{1}{-g_j} + \mathbf{m}_j^T \mathbf{h}_j \right)^{-1}, \tag{B.15}$$

$$G_{s,(j)} = G_{11} - G_{12} G_{22,(j)}^{-1} G_{12}^T, \tag{B.16}$$

$$G_{22,(j)} = \widetilde{G}_{22,(j-1)} - \mathbf{m}_j g_j \mathbf{m}_j^T, \tag{B.17}$$

and $G_{s,(0)} = G_{11} - G_{12} G_{22}^{-1} G_{12}^T$, $G_{22,(0)} = G_{22}$. Formula (B.12) clearly shows that deleting each new resistor adds a new term to the error bound Err_{ves}.

Implementation issues and complexity

Based on (B.12), we suggest an algorithm for simplifying resistor networks. According to the algorithm, all resistors are subdivided into three parts: the first part contains only resistors between external nodes (conductance block G_{11}), resistors between internal nodes (conductance block G_{22}), and resistors between external and internal nodes (conductance block G_{12}). Resistors in G_{22} are sorted in decreasing order, and the error bound (B.12) is computed for each resistor. If Err_{ves} is less than the predefined tolerance δ, then such resistor is deleted from G_{22}, otherwise the next resistor is considered. If it happens that the computed error bound is larger than δ for more than T_{max} resistors or all possible resistors have been considered, then the process is stopped.

We can estimate the cost of the algorithm by calculating the number of times the algo-

rithm requires to solve linear systems and the number of times it requires to downdate the Cholesky factorizations. Suppose k_1 resistors have been checked to be deleted, and only k_2 resistors have been deleted, i.e. $k_2 \leq k_1$. In total, $2k_1 + k_2 n_e$ linear systems with lower triangular *sparse* matrices and $2k_1$ linear systems with *dense* lower triangular matrices have to be solved, and k_2 Cholesky factorizations $G_{22} = LL^T$ and $G_s = L_s L_s^T$ have to be downdated. In total, the complexity is $(2k_1 + k_2 n_e)O(n_i^\beta) + k_2 O(2n_i n_e^2) + k_2 O(n_i^2)$, $\beta \leq 2$.

For a dense Cholesky factorization (e.g., $L_s L_s^T$), rank-n_i downdating costs roughly $2n_i n_e^2 + 4n_i n_e$ floating-point operations. However, there are available algorithms for multiple-rank modifications of a sparse Cholesky factorization [24] which lead to improved complexity. These algorithms exploit the pattern of entries in the Cholesky factor which can be changed. For sparse symmetric positive definite matrices, the sparse Cholesky factorization, rank-1, and rank-r update/downdate functions are available in the set of routines CHOLMOD [19], which we used for efficient computing Err_{ves} in the examples from Section 5.8.

Bibliography

[1] *Fasterix, Package for the Electromagnetic Simulation of PCB Layout from Philips Electronic Design and Tools/Analogue Simulation*.

[2] *NXP Semiconductors, Pstar, ver. 6.0*.

[3] R. Achar and M. S. Nakhla. Simulation of High-Speed Interconnects. In *Proc. of the IEEE*, pages 693–728, 2001.

[4] O. Alac, B. Scott, and W.F. Tinney. Sparcity oriented compensation methods for modified network solutions. *IEEE Transactions on Power Apparatus and Systems*, PAS-102:1050–1060, May 1983.

[5] D. Aldous. *Reversible Markov Chains and Random Walks on Graphs*. Book in preparation, Available at www.stat.berkeley.edu/ aldous/RWG/book.html, 2003.

[6] O. Alsac, B. Scott, and W.F. Tinney. Sparcity-oriented compensation methods for modified network solutions. *IEEE Trans. on Power Apparatus and Systems*, PAS-102:1050–1060, May 1983.

[7] P.R Amestoy, T.A. Davis, and I.S. Duff. An approximate minimum degree ordering algorithm. *SIAM J. Matrix Anal. Appl.*, 17:886–905, 1996.

[8] B.D.O. Anderson and S. Vongpanitlerd. *Network Analysis and Synthesis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1973.

[9] A. C. Antoulas. Approximation of Large-Scale Dynamical Systems: An Overview. In *Large Scale Systems 2004: Theory and Applications. A Proceedings Volume from the 10th IFAC/IFORS/IMACS/IFIP Symposium*. Osaka, Japan, 26-28 July 2004.

[10] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Agebraic Equations*. Siam, 1998.

[11] J. Baglama and L. Reichel. Augmented implicitly restarted Lanczos bidiagonalization methods. *SIAM J. Sci. Comp.*, 27:19–42, 2005.

[12] E. Barke. Resistance calculations from mask artwork data by finite element method. In *Proc. 22nd Design Automation Conference*, pages 305–311, 1985.

[13] M. Benzi. Preconditioning techniques for large linear systems: A survay. *Journal of Computtional Physics*, 182(2):418–477, 2002.

[14] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Science*. Academic Press, New York, San Francisco, London, 1979.

[15] W. E. Boyce and R. C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*. Wiley, 2005.

[16] L. E. M. Brackenbury. *Design of VLSI Systems - A Paractical Introduction*. Macmillan, Houndmills, Basingstoke, Hampshire RG21 2XS, London, 1987.

[17] C. A. Brebbia and J. Dominguez. *Boundary Elements. An Introductory Course*. Computational Mechanics Publications, 1989.

[18] T. F. Chan and H. A. van der Vorst. Approximate and incomplete factorizations. In Venkatakrishnan V. Keyes A., Sameh V., editor, *ICASE/LaRC Interdisciplinary Series in Science and Engineering*, pages 167–202. Kluwer, Dordrecht, 1997.

[19] Y. Chen, T.A. Davis, W.W. Hager, and S. Rajamanickam. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software*, 35, 2008.

[20] R.Du Cloux, G.P.J.F.M Maas, and A.J.H Wachters. Quasi-static boundary element method for electromagnetic simulations of PCBs. *Philips J. Res.*, 48:117–144, 1994.

[21] Comsol. Comsol multiphisics ver. 3.4, http://www.comsol.com, september 2010.

[22] T. A. Davis. Suit Sparce: A Suit of Sparce Matrix Packages ver. 3.4.0, http://www.cise.ufl.edu/research/sparse/SuitSparse, june 2010.

[23] T.A. Davis and W.W. Hager. Modifying a Sparse Cholesky Factorization. *SIAM J. Matrix Anal. Appl.*, 20:606–627, 1999.

[24] T.A. Davis and W.W. Hager. Multiple-Rank Modifications of a Saprse Cholesky Factorization. *SIAM J. Matrix Anal. Appl.*, 22:997–1013, 2001.

[25] D. K. Faddeev and V. N. Faddeeva. *Computational Methods of Linear Algebra*. W. H. Freeman and Company, 1963.

[26] D.R. Fokkema, G.L.G. Sleijpen, and van der H.A. Vorst. Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pensils. *SIAM J. Sci. Comp.*, 20:94–125, 1998.

[27] R.V. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *Journal of Computational and Applied Mathematics*, 123:395–421, 1999.

[28] A. J. van Genderen, N. P. van der Meijs, and T. Smedes. Space substrate resiastance extraction user's manual. Report et-ns 96-03, Delft University of Technology, 2006.

[29] A.J. van Genderen. *Reduced Models for the Behavior of VLSI Circuits*. PhD thesis, Delft University of Technology, Delft, 1991.

[30] A. Ghosh, S. Boyd, and A. Saberi. Minimizing effective resistance of a graph. *SIAM Rev.*, 50:37–66, 2008.

[31] D. Gleich. Matlabbgl: A matlab library, http://www.stanford.edu/dgleich/programs/matlab-bgl, april 2010.

[32] G. H. Golub and C. F. van Loan. *Matrix Computations*. John Hopkins University Press, 1996.

[33] H.C. de Graaf and F.M. Klaassen. *Compact Modeling for Circuit Design*. Springer Verlag, 1990.

[34] M. Green and D. J. N. Limebeer. *Linear Robast Control*. Prentice-Hall, 1995.

[35] E. Grimme. *Krylov projection methods for model reduction*. PhD thesis, Coordinated-Science Laboratory, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 1997.

[36] S. Gugercin and A. C. Antoulas. Model reduction of large-scale systems by least squares. *Lin. Alg. Appl.*, 415(2-3):290–321, 2006.

[37] E. A. Guillemin. *Synthesis of Passive Networks*. Wiley, New York, 1957.

[38] B. Gustavsen and A. Semlyen. Enforcing passivity for admittance matrices approximated by rational functions. *IEEE Trans. on Power Systems*, 16:97–104, 2001.

[39] G. D. Hachtel, R. K. Brayton, and F. G. Gustavson. The Sparse Tableau Approach to Network Analysis and Design. *IEEE Trans. on Circuit Theory*, 1971.

[40] M. T. Heath. *Scientific Computing. An Introductory Survey*. McGraw-Hill, New York, 2002.

[41] P. Heres and W. Schilders. Model order reduction of interconnect structures in Fasterix. Technical note, Philips Research, 2003.

[42] P. J. Heres. *Robust and Effcient Krylov Subspace Methods for Model Order Reduction*. PhD thesis, Eindhoven University of Technology, Eindhoven, 2005.

[43] C-W Ho, A.E. Ruehli, and P.A. Brennan. The modified nodal appoach to network analysis. *IEEE Transactions on circuits and design*, CAS-22:504–509, 1975.

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

[45] M. E. Hochstenbach. Harmonic and refined extraction methods for the singular value problem, with applications in least squares problems. *BIT Numerical Mathematics*, 44:721–754, 2004.

[46] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[47] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1991.

[48] R. Ionutiu and J. Rommes. Circuit synthesis of reduced order models. Technical note NXP-TN-2008/00316, NXP Semiconductors, 2009.

[49] R. Ionutiu and J. Rommes. Model order reduction for multi-terminal circuits. CASA-Report 09-29, Eindhoven University of Technology, 2009.

[50] R. Ionutiu and J. Rommes. SparseRC: sparsity preserving model reduction for RC circuits with many terminals. CASA-Report 11-05, Eindhoven University of Technology, 2011.

[51] J. D. Jackson. *Classical Electrodynamics*. Wiley, Chichester, 1999.

[52] T. Kailath. *Linear Systems*. Prentice Hall, Englewood Cliffs, N.J., 1980.

[53] M. Kamon, A. Marques, L. M. Silveira, and J. White. Automatic Generation of Accurate Circuit Models of 3-D Interconnect. *IEEE Transactions on Components, Packaging, and Manufacturing Technology - Part B*, 21(3):225–240, 1998.

[54] G. Karypis and V. Kumar. *METIS, A software package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. http://glaros.dtc.umn.edu/gkhome/metis/.

[55] R.B. Lehoucq and D.C. Sorensen. Deflation techniques within an implicitly restarted Arnoldi iteration. *SIAM J. Matrix Anal. Appl.*, 17:789–821, 1996.

[56] P. Lenaers. Model order reduction for large resistive networks. Final report, Technische Universiteit Eindhoven, 2008.

[57] A. C. S. Lima, B. Gustavsen, and A. B. Fernandes. Inaccuracies in network realization of rational models due to finite precision of RLC branches. In *Proc. Int. Conf. Power Syst. Transients*. Lyon, France, June 4-7 2007.

[58] P. Liu, Z. Qi, and S. X. D. Tan. Passive hierarchical model order reduction and realization of RLCM circuits. In *Proc. of the sixth International Symposium on Quality Electronic Design (ISQED'05)*, 2005.

[59] Mathworks. MATLAB ver. 7.5.0.338, http://www.mathworks.com, april 2010.

[60] N. P. van der Meijs. *Accurate and Efficient Extraction*. PhD thesis, Delft University of Technology, 1992.

[61] N. P. van der Meijs, A. J. van Genderen, F. Beefrink, and P. J. H. Elias. Space user's manual. Report et-nt 92.21, Delft University of Technology, 2005.

[62] P. Miettinen, M. Honkala, and J. Roos. Using METIS and HMETIS algorithms in circuit partitioning. Ciruit theory laboratory report series ct-49, Helsinki University of Technlogy, 2006.

[63] R.F. Milsom. RF simulation of passive ics using Fasterix. Report rp3506, Philips Electronics, 1996.

[64] R.F. Milsom, K.J. Scott, and P.R. Simons. Reduced equivalent circuit model for PCB. *Philips J. Res.*, 48:9–35, 1994.

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

[66] A. Odabasioglu, M. Celik, and Pileggi L.T. PRIMA: Passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 8:645–654, 1998.

[67] B. N. Parlett. *The Symmetric Eigenvalue Problem, Ser. Classics in Applied Mathematics*. SIAM, Philadelphia, PA, 1998.

[68] B. N. Parlett and J. K. Reid. Tracking the Progress of the Lanczos Algorithm for Large Systems Eigenproblems. *IMA Journal of Numerical Analysis*, 1:135–155, 1981.

[69] C. R. Paul. *Analysis of Multiconductor Transmission Lines*. Wiley, Hoboken, New Jersey, second edition, 2008.

[70] J.R. Phillips and L.M. Silveria. Poor man's TBR: A simple model reduction scheme. *IEEE Trans. CAD Circ. Syst.*, 24:283–288, 2005.

[71] A.C. Polycarpou. *Introduction to the Finite Element Method in Electromagnetics*. Morgan & Claypool, 2006.

[72] J. Rommes. *Methods for Eigenvalue Problems with Application in Model Order Reduction*. PhD thesis, Universiteit Utrecht, Utrecht, 2007.

[73] J. Rommes and N. Martins. Efficient computation of multivariable transfer function dominant poles using subspace acceleration. *IEEE Trans. Power Syst.*, 21(4):1471–1483, Nov 2006.

[74] J. Rommes and W.H.A. Schilders. Efficient methods for large resistor networks. *IEEE Trans. on CAD Circ. Syst.*, 29:28–39, 2010.

[75] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, second edition, 2003.

[76] D. Saraswat, R. Achar, and M. Nakhla. A fast algorithm and practical considerations for passive macromodeling of measured/simulated data. *IEEE Trans. Advanced Packaging*, 27:57–70, 2004.

[77] W. H. A. Schilders, H. A. van der Vorst, and J. Rommes. *Model Order Reduction. Theory, Research Aspects and Applications*. Springer, Berlin, 2008.

[78] E. Schrik. *A combined BEM/FEM Method for IC Substrate Modeling*. PhD thesis, Delft University of Technology, Delft, 2006.

[79] E. Schrik and N. P. van der Meijs. Combined BEM/FEM Substrate Resistance Modeling. In *Proceedings of the 39th Design Automation Conference*, pages 771–776. New Orleans, LA, August 2002.

[80] E. Schrik and N.P. van der Meijs. Combined BEM/FEM vs. 3DFEM substrate resistance modeling. Technical report, Delft University of Technology.

[81] L. M. Silveira, M. Kamon, and J. White. Efficient Reduced-Order Modeling of Frequency-Dependent Coupling Inductances associated with 3-D Interconnect Structures. In *Proc. 32nd Design Automation Conf.*, pages 376–380. San Francisco, CA, June 1995.

[82] T. Smedes, N.P. van der Meijs, and A.J. van Genderen. Boundary element methods for 2D capacitance and substrate resistane calculations in inhomogeneous media in a vlsi layout verification package. *Advances in Engineering Software*, 20:19–27, 1994.

[83] G. W. Stewart and Ji-guang Sun. *Perturbation Theory*. Academic Perss, INC., Boston, San Diego, New York, London, Sydney, Tokyo, Toronto, 1990.

[84] M. Stoll. A Krylov-Schur approach to the truncated SVD. Preprint submitted to Elsevier, 2010.

[85] S.X.D. Tan and L. He. *Advanced Model Order Reduction Techniques in VLSI Design*. Cambridge University Press, 2007.

[86] M. V. Ugryumova, J. Rommes, and W. H. A. Schilders. Error bounds for reduction of multi-port resistor networks. Submitted to Int. J. Numer. Model., 2010.

[87] M. V. Ugryumova, J. Rommes, and W. H. A. Schilders. On approximate reduction of multi-port resistor networks. To appear in Scientific Computing in Electrical Engineering, Proceedings of SCEE 2011, 2011.

[88] M. V. Ugryumova and W. H. A. Schilders. On passivity of the super node algorithm for EM modeling of interconnect systems. In *Proc. Design, Automation and Test in Europe (DATE 2010)*. Dresden, Germany, 2010.

[89] M. V. Ugryumova and W. H. A. Schilders. Stability and passivity of the super node algorithm for EM modeling of ICs. In *Scientific Computing in Electrical Engineering (SCEE 2008)*. Springer, 2010.

[90] M. E. Verbeek. Patrial element equivalent circuit (PEEC) models for on-chip passives and interconnects. *Int. J. Numer. Model.*, (2):61–84, 2004.

[91] J. Vlach and K. Singhal. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Reinhold: Ney York, 1994.

[92] A. J. H. Wachters. Mathematical methods used in Fasterix. Technical note 408/96, Philips Electronics, 1996.

[93] A.J.H. Wachters and W.H.A. Schilders. Simulation of EMC Behaviour. In P.G. Girlet, editor, *Handbook of Numerical Analysis*, volume VIII, pages 661–753. Elsevier, North Holland, 2005.

[94] J. S. van Welij. Basis functions matching tangential components on element edges. In *Proc. SISDEP-2 Conf.* Swansea, UK, 1986.

[95] F. Yang, Y. Zeng, Y. Su, and D. Zhou. RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI. *Commun. comput. Phys.*, 3:376–396, 2008.

[96] Z. Ye, D. Vasilyev, Z. Zhu, and J. Phillips. Sparse Implicit Projection (SIP) for Reduction of General Many-Terminal Networks. In *Proc. of the 2008 IEEE/ACM Int. Conf. on Computer-Aided Design*, pages 736–743, 2008.

[97] M. Zahn. *Electromagnetic Field Theory: a problem solving approach*. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, 1979.

Summary

The electronic industry is concerned with many different areas. One of those areas is the modeling of integrated circuits (ICs) which are used in all kinds of electronic devices. Due to the fact that the overall geometry sizes of ICs in general become smaller and the frequencies of the signals used in these structures are increasing, many kinds of effects are tending to influence the behaviour of the ICs. For instance, interconnects are not ideal conductors and this may cause delays, transistors behave as not ideal switches, and the substrate may cause crosstalk between different parts of the IC. These effects may be dramatic if they are not well-understood and precautions are not taken. To be able to couple these inherently different structures of IC, the generally large models of interconnect and substrate have to be replaced by smaller equivalent models which have approximately the same behavior. For this goal, model order reduction is required. To fulfill the demands of industrial applications, model order reduction methods should be accurate, efficient and flexible.

In this thesis we considered several realistic industrial problems arising from IC modeling which required improved model order reduction methods. The results of the research can be summarized as follows. Firstly, it is proved that a model order reduction technique, the super node algorithm, used in the electromagnetic tool Fasterix, delivers stable but not always passive models. For ICs applications it is very important to preserve properties of the system, such as stability and passivity, in order to provide physically reliable models. We handled this problem by considering a few approaches for passivity enforcement which we incorporated into the super node algorithm. An approach which is based on calculating a correction to the rational approximation of admittance matrix based on linearization and constrained minimization by Quadratic Programming delivers passive circuits which also have exactly the same size as the reduced circuit obtained after applying the super node algorithm.

Secondly, we propose a novel approach for reduction of multi-terminal resistor networks which often arise during substrate extraction. The basic idea is to improve the sparsity of the conductance matrix by neglecting resistors, which do not contribute significantly to the behavior of the circuit. In order to do this precisely and to be able to

keep a control on the quality of approximation, we derived explicit analytical error estimations which demand less computational effort than the direct error computations, especially when the number of ports is large. Based on these error estimations we designed algorithms which allow to delete resistors by groups and can be combined with existing reduction methods to obtain further reduction.

In order to simulate and predict properly substrate crosstalk, accurate and efficient substrate modeling methods are required. Substrate extraction often involves finding a resistance network between ports which describes the behavior of the substrate. In this thesis, we also considered the problem of resistance extraction of a substrate with homogeneous doping profile. We solved the problem by means of two discretization methods: the finite element method (FEM), and the boundary element method (BEM). We showed that for extraction of homogeneous substrate, BEM works faster than FEM and achieves grid-independent results earlier.

In addition, we consider a challenging industrial problem related to modeling of MOS transistors. For given voltage excitations at the top ports, we suggested an algorithm to compute output current at these ports. The solution is found in an iterative manner, requiring only solutions of the factorized linear systems and evaluations of a nonlinear function. This strategy can also be applied for modeling of MOS transistors with larger amount of top contacts and it is expected to be very helpful in a case of a highly nonlinear dependence between the voltage difference and the current at the bottom contacts.

From the point of view of real applications, the results in this thesis are a contribution to the development of model order reduction which remains a very interesting area with lot of potential for further improvements.

Samenvatting

De elektronische industrie heeft belang bij een groot aantal onderzoeksgebieden; zon gebied behelst geïntegreerde schakelingen (integrated circuits, ICs), die in veel verschillende toepassingen worden gebruikt. Omdat de afmetingen van ICs in het algemeen steeds kleiner worden, terwijl de frequenties van gebruikte signalen almaar toenemen, kunnen er effecten optreden die het functioneren van de ICs beïnvloeden. Zo kunnen er bijvoorbeeld signalen vertragen omdat de verbindingen niet perfect geleiden, of de transistoren gedragen zich wellicht niet als ideale schakelingen, of er kan, door het substraat heen, overspraak tussen verschillende delen van het IC ontstaan. Dergelijke effecten kunnen grote gevolgen hebben als men ze niet goed begrijpt en er geen overeenkomstige maatregelen genomen worden. Om goed te kunnen begrijpen hoe de verschillende onderdelen van een IC met elkaar samenhangen moeten de - in het algemeen bijzonder grote - modellen van verbindingen en substraat vervangen worden door kleinere modellen die min of meer hetzelfde gedrag vertonen. Hiertoe zijn modelorde-reductie methoden vereist. Deze methoden moeten nauwkeurig, efficiënt en flexibel zijn, zodat ze kunnen worden gebruikt voor industriële toepassingen.

In dit proefschrift hebben we een aantal realistische problemen uit de industrie bekeken, betrekking hebbend op het modelleren van ICs waarbij modelorde-reductie methoden nodig zijn. De resultaten van dit onderzoek kunnen als volgt worden samengevat. Ten eerste is bewezen dat het super node-algoritme, een modelorde-reductie techniek gebruikt in het simulatiepakket voor electromagnetisme Fasterix, stabiele maar niet altijd passieve modellen oplevert. Voor toepassingen van ICs is het van belang om eigenschappen zoals stabiliteit en passiviteit te behouden. Daartoe is een aantal aanpakken om passiviteit te vergroten bestudeerd, en hebben we die aanpakken ingebed in het super node-algoritme. Een zon aanpak, waarbij een correctie op de rationale benadering van de admittantiematrix wordt berekend aan de hand van linearisatie en optimalisatie van kwadratische problemen onder restricties, levert passieve circuits op die bovendien precies even groot zijn als de circuits die verkregen worden met het super node-algoritme.

Ten tweede stellen wij een nieuwe aanpak voor om weerstandsnetwerken met meer-

dere contactpunten, die vaak ontstaan tijdens de substraatextractie, te reduceren. Het idee is om de ijlheid van de geleidingsmatrix te vergroten, door weerstanden te negeren die geen significante invloed op het gedrag van het circuit hebben. Om dit nauwkeurig te doen en de kwaliteit van de benadering te kunnen beheersen, hebben we expliciete analytische foutafschattingen afgeleid die minder rekenkracht nodig hebben dan directe foutberekeningen, met name als er veel contactpunten zijn. Gebruikmakend van deze afschattingen hebben we algoritmen ontworpen waarmee groepen van weerstanden kunnen worden verwijderd uit het model. Deze algoritmen kunnen nog gecombineerd worden met bestaande reductiemethoden om verdere reductie te verkrijgen. Om overspraak door het substraat goed te kunnen simuleren en voorspellen, zijn nauwkeurige en efficiënte modelleermethoden voor het substraat vereist. Substraatextractie behelst dikwijls het vinden van een weerstandsnetwerk tussen contactpunten dat het gedrag van het substraat beschrijft. In dit proefschrift is ook bestudeerd hoe dergelijke netwerken gevonden kunnen worden voor een substraat met homogene materiaaleigenschappen. Dit probleem hebben we opgelost met behulp van twee verschillende discretisatiemethoden: de eindige elementenmethode (finite element method, FEM) en de randelementenmethode (boundary element method, BEM). Voor de extractie van homogene substraten hebben we aangetoond dat BEM sneller is dan FEM, en dat BEM eerder roosteronafhankelijke resultaten oplevert.

Daarnaast hebben we een uitdagend probleem uit de industrie bestudeerd, dat betrekking heeft op het modelleren van MOS-transistoren. Wij stellen een algoritme voor om, bij gegeven spanningsimpuls op de bovenste contacten, de uitgaande stroom op deze contactpunten te berekenen. De oplossing wordt op iteratieve wijze verkregen, waarbij slechts oplossingen van het gefactoriseerde lineaire systeem en het uitrekenen van een niet-lineaire functie benodigd zijn. Deze strategie kan ook worden toegepast op MOS-transistoren met een groot aantal contacten aan de bovenkant van het substraat. Wij verwachten dat het ook bruikbaar zal zijn in het geval van een sterk niet-lineaire afhankelijkheid tussen spanningsverschillen en stroom op de onderste contacten.

Vanuit het oogpunt van de toepassingen dragen de resultaten in dit proefschrift bij aan de ontwikkeling van modelorde-reductiemethoden, wat nog steeds een interessant onderzoeksgebied is met veel mogelijkheden voor verdere verbeteringen.

Curriculum vitae

Maria Vladimirovna Ugryumova was born on 25th June 1983 in Novosibirsk, Russia. After graduating from the Technical Lyceum with the major in physics and mathematics in 2000 she became a student at the Department of Applied Mathematics and Computer Science at Novosibirsk State Technical University, Novosibirsk, Russia. In 2004 she obtained a Bachelor degree (BCh), and in 2006 she obtain a Master degree (MSc) in applied mathematics and computer science from Novosibirsk State Technical University. Between 2004 and 2006 Maria worked as a developer and programmer of user functions for Unigraphics at Novosibirsk Aviation Industrial Company, Novosibirsk, Russia. In 2007 Maria began a PhD project at the Eindhoven University of Technology and NXP Semiconductors, Eindhoven, The Netherlands, the results of which are presented in this dissertation.

Acknowledgments

While writing this thesis I had a lot of support from many people. I would like to express my gratitude to all those who have contributed to this thesis in many different ways. First and foremost I would like to thank my supervisor Prof. Wil Schilders for giving me the opportunity to do my PhD research at the CASA group of the Eindhoven University of Technology, and also for his guidance and stimulating support during the past four years. I also would like to thank him for the comfortable work atmosphere, and the opportunity to visit conferences where I have met many people from the field of my research.

I am very thankful to my copromotor Dr. Michiel Hochstenbach who helped me with challenging questions related to my research and who has read my thesis and gave me many helpful comments for improving the content. I am also very grateful to my advisor Dr. Joost Rommes from NXP Semiconductors who invested his time to check my papers, presentations and gave me many useful advices and support during my research.

I would like to express my sincere gratitude to the members of the promotion committee including Prof. Luis Miguel Silveira, Prof. Kees Vuik, Prof. Bob Mattheij and Prof. Arjeh Cohen, together with my supervisor Prof. Wil Schilders, copromotor Dr. Michiel Hochstenbach and the members of the extended promotion committee including Prof. Anton Tijhuis and Dr. Joost Rommes. I would like to thank them for the time to read my thesis and their willingness to judge my work.

I would like to thank Dr. Nick van der Meijs and Dr. Simon de Graaf for the help with the layout-to-circuit extractor SPACE which I used to obtain the results in Chapter 6 of this thesis.

What made these four years especially bright and enjoyable was the great working atmosphere within CASA and NXP Semiconductors. I would like to thank my current and former colleagues for many social events we have shared together, all of which made my life in Eindhoven special. Thank you Ali Etaati, Tasnim Fatima, Yves van Gennip,

Christina Giannopapa, Shruti Gumaste, Andriy Hlod, Bart Janssen, Jan Willem Knopper, Mark van Kraaij, Kundan Kumar, Bas van der Linden, Temesgen Markos, Oleg Matveichuk, Lena Filatova, Peter int Panhuis, Laura Astola, Evgeniya Balmashnova, Sinatra Canggih, Miguel Patricio Dias, Remco Duits, Maxim Pisarenco, Valeriu Savcenco, Berkan Sesen, Volha Shchetnikava, Antonino Simone, Mayla Bruso, Shona Yu, Sudhir Srivastava, Jurgen Tas, Ronald Rook, Arie Verhoeven, Venkat Viswanathan, Erwin Vondenhoff, Fan Yabin, Luiza Bondar, Willem Dijkstra, Rostyslav Polyuga, Michiel Renger, Prasad Perlekar, Badr Kaoui, Arpan Ghosh, Steffen Arnrich, David Bourne, Lucia Scardia, Martien Oppeneer, Nicodemus Banagaaya, Jan Niehof, Bratislav Tasic, Michael Striebel, Jan ter Maten, Kasra Mohaghegh, Maryam Saadvandi, Tamara Bechtold, Luciano de Tomassi, Davit Harutyunyan and many more. Special thanks to the people with whom I have shared offices in the past: Mirela Darau, Hans Groot, Qingzhi (Darcy) Hou, Roxana Ionutiu, Godwin Kakuba, Evelyne Knapp, Agnieszka Lutowska, Zoran Ilievski, Corien Prins, Luis Caballero, Ardy van der Berg, and Lavanya Jagan. I am appreciate the help and assistance given to me by Enna van Dijk, Marèse Wolfs-van de Hurk and Irene Andringa Portela who helped me with wide variety of administrative issues.

I am also very grateful to Michiel Renger for translating the Summary of this thesis in to Dutch, and my paranimfs Maria Rudnaya and Neda Sepasian who agreed to be with me during the defence ceremony.

On more personal level, I would like to thank my dearest parents Irina Ugryumova and Vladimir Ugryumov, my sister Anna Karpova, my cousins Natalya Abrosimova and Elena Pynzar, my grandparents, and other family members who are very important for me and who supported me every day being far away from me.

Last, but definitely not least, I would like to thank my special one, Patricio Rosen Esquivel, for his love and unlimited support in all aspects of life.

Maria Ugryumova

Eindhoven, March 2011.