Identification of Electrical Circuits for Realization of Sparsity Preserving Reduced Order Models

Christof Kaufmann

25th March 2010

Abstract

Nowadays very-large scale integrated circuits contain a large portion of interconnect. The parasitics of this network of interconnect can affect among other things the maximum data rate of the internal data bus. Therefore simulation of a model of these networks is necessary to predict the behaviour of the interconnect. Such a model can be so large that its simulation can take days or might even be unfeasible. Model order reduction techniques can decrease the number of unknowns of a model, thus reducing simulation time considerably. Special model order reduction methods yield reduced models that can be correspond to a real circuit smaller than the original one.

This work starts with a description of the modelling of electrical circuits. Examples are given to illustrate the procedure. Then, the basics of model order reduction are explained to give a basic understanding why and how methods are used to reduce models. The main focus of this work is on the recently developed model order reduction method SparseMA. The method is described in detail and practical aspects as well as examples are given for many steps of this method. SparseMA is capable to reduce RC circuits and yields reduced models that can be formed by RC circuits again. These resulting circuits are analyzed to gain new insights. It is illustrated that SparseMA cannot be applied to every RC circuit. The insights gained provide valuable information towards identification of electrical circuits suited for SparseMA.

Acknowledgements

I would like to thank Prof. Dr. rer. nat. Michael Knorrenschild, Prof. Dr. rer. nat. Michael Günther and Kasra Mohaghegh for their great support. Then I want to mention Christian Jelenowski and Srinivasan Natarajan because of a lot of fruitful discussions on and off the topic. Additionally I appreciate the helpful comments of Joost Rommes and Roxana Ionutiu. Without all these people it would not have possible to write this work.

Declaration

The work described in this report is the result of my own investigations. All sections of the text and results that have been obtained from other work are fully referenced.

....

(Date) (Signed)

Contents

1 Introduction

The development of circuits and integrated circuits (ICs) becomes more and more complicated. Electrical devices need to be small, cheap and efficient. These requirements lead to highly integrated circuits (chips); the process of creating these kind of ICs is called very-large-scale integration (VLSI). New philosophies have established themselves in development of ICs.

One is the usage of highly configurable ICs. These ICs are called field-programmable gate arrays (FPGAs) and are widely used in flexible, non-mass products. They contain many logical blocks, that can be connected to each other by a programmable configuration to build complex functions, e.g. decoders and even whole (relatively simple) microcontrollers. For this functionality the blocks must have physical connections, which are called *interconnect*. Since there is a large number of the logical blocks there is a lot of interconnect.

Another philosophy is the system on chip (SoC) concept, which leads also to large portions of interconnect. SoC means, a whole device (or system) is as far as possible integrated on a single chip and this concept is used for most mass products, as for example USB memory sticks or complex microcontrollers as used e.g. in handheld devices. The engineers that design these ICs buy components as intellectual property (software) from special companies where applicable (e.g. layouts for memory) and develop only the remaining components by themselves. This reduces time-to-market duration significantly. When all components are finished they must get connected. This *interconnect* is used by very fast bus protocols to transport the data in form of energy between the different components.

Now, every non-ideal wire (or interconnect in general) has parasitic effects (in short: parasitics). One is series inductance (L), because a current has always a magnetic field around it, in which energy in form of magnetic flux is saved. A change of current requires a change of energy in the magnetic field. Another effect is capacitance (C), which arises because of the ability for a material between two connections to save energy in the form of electrical charge. So, a change of voltage requires a change of energy. The third effect is resistance (R). This is not an energy saving effect, but an energy consuming by dissipating heat, i.e. converting electrical energy to heat energy. Certainly, the described parasitic effects depend on the dimensions and material properties (specific conductance, permeability, permittivity) of the interconnect and hence on the layout of an IC. As passive filters consist of resistors (R), capacitors (C) and inductors (L), one can understand why interconnect induces a frequency selective behaviour. This behaviour causes a non-ideal signal propagation. If high data rates are required for the information that travels on the interconnect, their parasitic effects must be considered carefully by the designer, since the frequency selective behaviour limits the bandwidth of the interconnect. Furthermore, perturbations in one signal can propagate to another nearby signal (”crosstalk”), by any combination of the described effects. That is why the connections have to be considered together, instead separately. So a signal that is sent from one end of an interconnect causes crosstalk to other interconnects and suffers additionally from frequency selective behaviour due to parasitics of the network of interconnects.

Therefore it is necessary that IC designers use simulation tools to predict the behaviour of the parasitics of the interconnect network before the IC is physically produced. Simulation saves time as well as costs, compared to (physical) production and testing. In order to simulate the behaviour, these simulation tools have to extract an electrical system (circuit) from the interconnect by considering dimensions and material properties of the layout of the IC. Such an electrical system could be a passive linear network with lumped R, C and L components. The extracted circuit can only approximate the interconnect, which is obvious with the following example. A model of a two-wire line would require infinitely many lumped components. However, to describe its behaviour with reasonable accuracy, the models can become very large. Their simulation could take too long or might be unfeasible. Now, so-called model order reduction techniques can be used to find a smaller model with a similar behaviour. Simulation of this smaller model takes usually less time than simulation of the original circuit. In this way the simulation time can be reduced or a simulation can become possible. However, there are several model order reduction methods. There is probably not one method, that is well-suited for reduction of all kind of circuits. So it would be desirable to be able to decide which method is best-suited for which circuit. Criterions for different methods could be employed on a circuit that indicate whether a method is suitable or not.

This paper is organized as follows. Chapter 2 introduces the reader into modelling of electrical RCL circuits and illustrates the procedure with examples. Different ways of a standard approach for modelling are discussed. Additionally, some important properties are also explained. Then, Model order reduction techniques are brought in in Chapter 3. After giving reasons for model order reduction, an overview of some widely used methods is provided. Some of their aspects should give a basic idea why other methods have been developed and are preferable. Another method is described to give a basic understanding for the method described in the following chapter: Chapter 4 focuses on model order reduction with a recently developed method, called *SparseMA*[12]. That chapter describes how SparseMA can be applied to reduce RC circuits. The reduced circuits are analyzed to gain new insights. It is illustrated that SparseMA cannot be applied to every RC circuit. The gained insights provide valuable information towards identification of electrical circuits suited for SparseMA. Additionally practical aspects are issued. Chapter 5 concludes and motivates further work with several explicit ideas.

2 Modelling of electrical circuits

An electrical circuit can be mathematically modelled by a system of some differential and some algebraic equations. Such systems are called differential algebraic equations (DAEs). In many situations these systems can numerically be solved by ordinary differential equation (ODE) solvers. These solvers compute the unknown states in time domain. The results of this simulation will of course depend on the input sources and on initial conditions. For more information on time domain simulations and a discussion on properties of circuits (and their mathematical model) important for time domain simulations, see e.g. [9, 14].

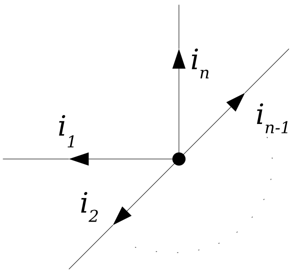
A circuit's behaviour in frequency domain, on the other hand, does not depend on the input signal. For the mathematical description in the frequency domain typically the transfer function (i.e. the ratio of outputs to inputs in frequency domain) is used. There are different kinds of transfer functions, as for example voltage amplification (or attenuation) if outputs and inputs are voltages at different nodes or as another example impedance parameters if outputs are voltages and inputs are currents at the same nodes. So, the result of simulations in frequency domain depends on the kinds and the positions of inputs and outputs.

In this chapter we will explain how the mathematical model is derived. At first Kirchhoff's laws have to be employed, also using voltage/current relationships for the various circuit elements. Then a systematic way to set up the differential-algebraic system, the so-called modified nodal approach (MNA) is described. The matrices arising in the modelling process can also be derived directly from the circuit in a so-called stamping process. Finally we will transfer the model from the time domain into the frequency domain and discuss the use of the transfer function.

2.1 Kirchhoff's Laws

Kirchhoff's laws are the basic tools for analysis of electrical circuits. There exist two Kirchhoff's laws, concerning nodes and loops.

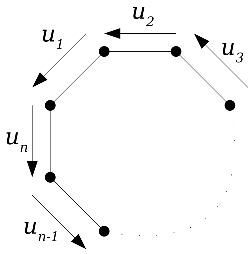
Kirchhoff's current law (KCL) (given by (2.1) and Figure 2.1) is the law for nodes. It states, that the sum of all currents leaving an arbitrary node is zero. Alternatively: The sum of all currents leaving an arbitrary node is equal to the sum of all currents entering the same node. This law is fundamental for circuit analysis and gives for each node one equation in the form of (2.1). It can be used by automatic software to build these so-called *node equations*.



$$\sum_{k=1}^{n} i_k = 0 \tag{2.1}$$

Figure 2.1: Currents leaving a node.

The other law, Kirchhoff's voltage law (KVL), states that the sum of all branch voltages along any closed loop is zero, see (2.2) and Figure 2.2. Automatic software can use this law to relate node voltages[1] to branch voltages.



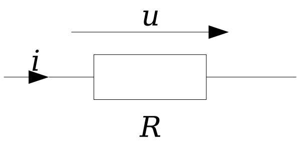
$$\sum_{k=1}^{n} u_k = 0 \tag{2.2}$$

Figure 2.2: Voltages along a loop.

2.2 Branch constitutive relationships

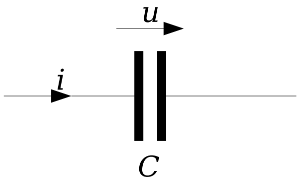
With Kirchhoff's laws the topology of a circuit can be described, but there are no information about the used electrical components included yet. Branch constitutive relationships (BCR) give the relationships between the current passing through an element to the voltage across its terminals (or dependent on the element their derivatives). The properties (e.g. resistance) of the elements are included as parameters. Since in this thesis the analysis is restricted to ideal resistors, capacitors and inductors, the BCRs are given only for these elements. See Figures 2.3, 2.4, 2.5 and (2.3), (2.4), (2.5), respectively. For the resistor the BCR is called Ohm's Law. Note that i is the current through the element and $\dot{i}$ its derivative.

[1] The term *node voltage* is throughout used in this document as *node voltage measured against ground*



$$u = R\,i \tag{2.3}$$

Figure 2.3: A resistor with resistance R.



$$i = C\,\dot{u} \tag{2.4}$$

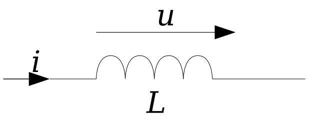
Figure 2.4: A capacitor with capacitance C.

Resistors (R), capacitors (C) and inductors (L) are linear elements, because the BCRs are linear equations. Circuits made of linear elements only and independent sources are called linear circuits.

2.3 Modelling in the time domain: modified nodal analysis

Modified nodal analysis is the conventional method to describe an electrical circuit as mathematical system. We consider circuits with R, C and L elements and only independent, grounded sources, i.e. sources which depend only on the input and with one end connected to ground. The notations are taken from [6]. MNA is done like described in the following.

First all nodes except ground get assigned a number. Then KCL is applied to all numbered nodes, while BCRs are used to describe the currents of resistors and capacitors by node voltages (or their time-derivatives). To complete the *MNA*, two kinds of equations have to be added and outputs have to be chosen. One equation has to be formulated for each inductor to describe its BCR, where the voltage across its terminals is given by node voltages. And finally, one equation has to be added for each (grounded) voltage source, to specify to which node the voltage source is connected. Note, that the currents of inductors and voltage sources will remain as currents in the node equations.



$$u = L\,\dot{i} \tag{2.5}$$

Figure 2.5: An inductor with inductance L.

This means in particular, that these currents are unknowns. To bring the described system of equations in a matrix-vector form, the coefficients of the node voltages and their derivatives have to be collected as well as the incidents of the inductor and voltage source currents. The outputs of the circuit can be chosen from the state vector of unknowns by forming a second matrix equation.

A small example circuit illustrates the MNA procedure. Then, the system of equations will be shown in a generalized form, which will be described extensively. From this general description a faster way of finding the system matrices (and vectors) will be considered, the so-called stamping procedure. Next, an important property of the system matrices will be described. Finally, a procedure to renumber circuit nodes is presented.

2.3.1 An Example

Now, MNA will be applied to the example circuit in Figure 2.6.

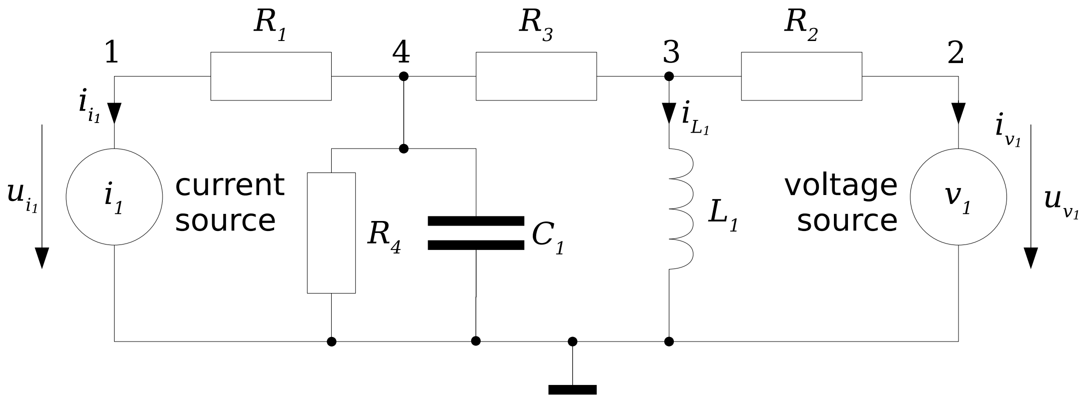


Figure 2.6: Example circuit with four numbered nodes, including one current source, one voltage source, four resistors, one capacitor and one inductor.

Firstly KCL is applied to the four numbered nodes and BCRs are used for the currents of resistors and capacitors. These four equations yield (2.6.1) to (2.6.4), where node voltage u_k denotes the voltage at node k measured against ground. Then, the BCR for the inductor L_1 has to be added and yields (2.6.5). Finally, (2.6.6) for the voltage source assigns the node voltage u_2 to the voltage of the voltage source u_{v_1}.

$$
\begin{aligned}
G_1\,(u_1 - u_4) + i_{i_1} &= 0 && (2.6.1)\\
G_2\,(u_2 - u_3) + i_{v_1} &= 0 && (2.6.2)\\
G_3\,(u_3 - u_4) + G_2\,(u_3 - u_2) + i_{L_1} &= 0 && (2.6.3)\\
C_1\,\dot{u}_4 + G_1\,(u_4 - u_1) + G_3\,(u_4 - u_3) + G_4\,u_4 &= 0 && (2.6.4)\\
L_1\,\dot{i}_{L_1} &= u_3 && (2.6.5)\\
u_{v_1} &= u_2 && (2.6.6)
\end{aligned}
$$

where $G_k = R_k^{-1}(k = 1, \dots, 4)$.
By collecting the coefficients of the node voltages (and their derivatives) and leaving on the right hand side only the sources (inputs) with negative sign (2.6.1) to (2.6.6) give

$$
\begin{array}{rrrrrrll}
 & G_1\,u_1 & & & -G_1\,u_4 & & = -i_{i_1} & (2.7.1)\\
 & & G_2\,u_2 & -G_2\,u_3 & & +i_{v_1} & = 0 & (2.7.2)\\
 & & -G_2\,u_2 & +(G_3 + G_2)\,u_3 & -G_3\,u_4 & +i_{L_1} & = 0 & (2.7.3)\\
C_1\,\dot{u}_4 & -G_1\,u_1 & & -G_3\,u_3 & +(G_1 + G_3 + G_4)\,u_4 & & = 0 & (2.7.4)\\
L_1\,\dot{i}_{L_1} & & & -u_3 & & & = 0 & (2.7.5)\\
 & & -u_2 & & & & = -u_{v_1} & (2.7.6)
\end{array}
$$

Now it is obvious to write it in matrix-vector form as

$$
\left(\begin{array}{cccc|c|c}
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & C_1 & 0 & 0\\
\hline
0 & 0 & 0 & 0 & L_1 & 0\\
\hline
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\left(\begin{array}{c}
\dot{u}_1\\ \dot{u}_2\\ \dot{u}_3\\ \dot{u}_4\\ \hline \dot{i}_{L_1}\\ \hline \dot{i}_{v_1}
\end{array}\right)
+
\left(\begin{array}{cccc|c|c}
G_1 & 0 & 0 & -G_1 & 0 & 0\\
0 & G_2 & -G_2 & 0 & 0 & 1\\
0 & -G_2 & G_3 + G_2 & -G_3 & 1 & 0\\
-G_1 & 0 & -G_3 & G_1 + G_3 + G_4 & 0 & 0\\
\hline
0 & 0 & -1 & 0 & 0 & 0\\
\hline
0 & -1 & 0 & 0 & 0 & 0
\end{array}\right)
\left(\begin{array}{c}
u_1\\ u_2\\ u_3\\ u_4\\ \hline i_{L_1}\\ \hline i_{v_1}
\end{array}\right)
$$

$$
= \left(\begin{array}{c|c}
1 & 0\\
0 & 0\\
0 & 0\\
0 & 0\\
\hline
0 & 0\\
\hline
0 & -1
\end{array}\right)
\left(\begin{array}{c}
-i_{i_1}(t)\\ \hline u_{v_1}(t)
\end{array}\right). \qquad (2.8)
$$

The output can be chosen according to the requirements. For this simple example the outputs are chosen to be the voltage at node 1 (at the current source) and the current at node 2 (at the voltage source). The equation for the output $y(t)$ follows.

$$y(t) = \left(\begin{array}{c} u_1 \\ \hline -i_{v_1} \end{array}\right) = \left(\begin{array}{cccc|c|c} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -1 \end{array}\right) \left(\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \hline i_{L_1} \\ \hline i_{v_1} \end{array}\right) = \left(\begin{array}{c|c} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & -1 \end{array}\right)^T \left(\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \hline i_{L_1} \\ \hline i_{v_1} \end{array}\right) \tag{2.9}$$

This example serves in the next paragraph as example for some blocks. In Section 2.4 the chosen output will be discussed.

2.3.2 General form of equations

The previous example had all kinds of elements and sources that will be considered in this work. So, the general form of equations should be easy to understand. Let us consider a circuit with N nodes, n_L inductors, n_G resistors, n_C capacitors, n_V voltage sources and n_i current sources. The state equation and the equation for the outputs are:

$$\mathcal{E}\,\dot{x} + \mathcal{A}\,x = \mathcal{B}\,u(t), \tag{2.10}$$

$$y(t) = \mathcal{C}^T\,x, \tag{2.11}$$

where

$$\mathcal{E} = \begin{pmatrix} C & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \mathcal{A} = \begin{pmatrix} G & A_L & A_V \\ -A_L^T & 0 & 0 \\ -A_V^T & 0 & 0 \end{pmatrix}, \qquad \mathcal{B} = \begin{pmatrix} A_i & 0 \\ 0 & 0 \\ 0 & -I \end{pmatrix}, \tag{2.12}$$

$$x = \begin{pmatrix} u_n \\ i_L \\ i_s \end{pmatrix}, \qquad u(t) = \begin{pmatrix} -i_i(t) \\ u_v(t) \end{pmatrix} \tag{2.13}$$

where $C = A_C\,\tilde{C}\,A_C^T$, $\quad G = A_G\,\tilde{G}\,A_G^T \in \mathbb{R}^{N\times N}$,
$\qquad \tilde{C} = \operatorname{diag}(C_1,\dots,C_{n_C}), \qquad \tilde{G} = \operatorname{diag}(G_1,\dots,G_{n_G}), \qquad L = \operatorname{diag}(L_1,\dots,L_{n_L})$.

Let $S := \{-1,0,1\}$. Then, $A_L \in S^{N\times n_L}$, $A_G \in S^{N\times n_G}$, $A_C \in S^{N\times n_C}$, $A_V \in S^{N\times n_V}$ and $A_i \in S^{N\times n_i}$ are the incidence matrices for inductors, resistors, capacitors, voltage sources and current sources, respectively. They are defined as described in the following. Let $a_{j,k}$ denote the entry in row j and column k in the incidence matrix for the specific kind of the element which is considered now. Then the respective matrix is defined by $a_{j,k} = 1$ if the current through the element which column k corresponds to is leaving node j and $a_{j,k} = -1$ if it is entering node j. All remaining entries are zero. For resistors, capacitors and inductors the direction of the current can be chosen arbitrarily and for sources it can be defined to flow into the source.

I is the $n_V \times n_V$ identity matrix. Therefore, $\mathcal{E}$ and $\mathcal{A}$ are $(N+n_L+n_V)\times(N+n_L+n_V)$ matrices and $\mathcal{B}$ is a $(N+n_L+n_V) \times (n_i+n_V)$ matrix.

$u(t)$ is the vector of inputs and is partitioned into the vector of currents that flow out of current sources $-i_i(t)$ and the vector of voltages across voltage sources $u_v(t)$.

x is the vector of unknown states and is partitioned in the vector of (unknown) node voltages u_n, the vector of (unknown) inductor currents i_L and the vector of (unknown) currents that flow into voltage sources i_s.

$y(t)$ is the vector of outputs. $\mathcal{C}$ describes the relation between states and outputs. The element of $\mathcal{C}$ in row n and column m is denoted by $c_{n,m} \in S$. Each column of $\mathcal{C}$ has got exactly one non-zero entry, according to the vector of unknown states x. So, if $c_{n,m} = 1$ state n (i.e. the n^{th} entry of x) is output m and if $c_{n,m} = -1$ it is the negative of it. The outputs can be node voltages (including voltages at current sources), inductor currents or currents flowing out of voltage sources. To illustrate this we consider the output of the previous example (Section 2.3.1). In that example $\mathcal{C}$ defines two outputs (see (2.9)). Output 1 is the voltage at node 1 u_1, which was the first state in x, i.e. $c_{1,1} = 1$. The output 2 was the current flowing out of the voltage source, which was the negative of the last state (state 6) i_{v_1}, i.e. $c_{6,2} = -1$. A common choice for $\mathcal{C}$ is $\mathcal{C} = \mathcal{B}$, which is also the case in the example. This is the special case when the outputs are chosen to be at the inputs, i.e. for current sources the output is the voltage at the source and for voltage sources the current leaving the source.

Now, that everything in (2.10) and (2.11) is defined, we want to remark that a system with only one input (i.e. $n_V + n_i = 1$) and one output (i.e. $\mathcal{C} \in S^{N+n_V+n_i}$) is called SISO (single input, single output). If a system has multiple inputs and multiple outputs, it is called MIMO. Hence a system with multiple inputs and only one output it is called MISO; one input and multiple outputs is called SIMO. Since MIMO systems are the more general systems, theory is derived for MIMO systems throughout this work.

Note, that G, C and L are symmetric matrices and G and C are positive semidefinite [16] while L is positive definite.

Now let A be the incidence matrix for all different kinds of circuit elements (R, C and L) and sources (voltage and current sources). It can be defined as follows.

$$A = \begin{pmatrix} A_G & A_C & A_L & A_V & A_i \end{pmatrix} \tag{2.14}$$

Let us consider A for the previous example (Section 2.3.1). Each column corresponds to one circuit element. These are written above the matrix (where v_1 is the voltage source and i_1 the current source). Each row corresponds to one circuit node, which are written to the right hand side of the matrix.

$$A = \begin{array}{cc}
\begin{array}{cccccccc} R_1 & R_2 & R_3 & R_4 & C_1 & L_1 & v_1 & i_1 \end{array} & \text{node} \\
\left(\begin{array}{cccc|c|c|c|c}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & -1 & 1 & 1 & 0 & 0 & 0
\end{array}\right) & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}$$

One entry is explained to clarify the example: $a_{1,8} = 1$ since the current from node 1 is leaving the node and flowing into the source i_1 (see Figure 2.6). Now it should be clear how the incidence matrix is defined.

The incidence matrix A enables us to describe the Kirchhoff's laws from Section 2.1 in a shorter way. By denoting the vector of node voltages u_n, i.e., the k-th entry of u_n is the voltage at node k (measured against GND), the vectors of edge (branch) currents i_e and the vector of edge (branch) voltages u_e and (2.14) i.e.,

$$i_e = \begin{pmatrix} i_G \\ i_C \\ i_L \\ i_V \\ i_i \end{pmatrix}, \qquad u_e = \begin{pmatrix} u_G \\ u_C \\ u_L \\ u_V \\ u_i \end{pmatrix}.$$

Assuming the same order for the columns in A and the entries in i_e and u_e with respect to the electric elements. Then KCL (2.1) and KVL (2.2) can be formulated more shortly [8]:

$$A\, i_e = 0 \quad \text{(KCL)}, \qquad\qquad A^T\, u_n = u_e \quad \text{(KVL)}$$

Note that the procedure of finding the system matrices $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{B}$ can be done automatically by first generating the incident matrices and then using the block-form in (2.10).

2.3.3 Stamping and unstamping procedure

Another approach of finding G and C directly from the circuit, which is also suitable for automation, is the so-called *stamping* procedure. Let $g_{n,m}$ be the entry of G in row

n and column m. Then

$$g_{n,m} = g_{m,n} = -G_{n,m} \;\forall\; n \neq m, \tag{2.15}$$

where $G_{n,m}$ is the conductance of the resistors that connect node n directly to node m. If there is no resistor connecting the respective nodes, the conductance is zero. Furthermore,

$$g_{n,n} = G_{n,0} + \sum_{k=1,k\neq n}^{N} G_{n,k} = G_{n,0} - \sum_{k=1,k\neq n}^{N} g_{n,k}, \tag{2.16}$$

where $G_{n,0}$ is the conductance of the resistors that connect node n directly to GND. Again, if there is no resistor connecting the respective node to GND, the conductance is zero.

The stamping procedure can also be done with C, by taking the capacities of the corresponding capacitors instead of the conductance of the corresponding resistors. The remaining blocks that complete (2.10) have to be found as before.

The *unstamping* procedure is the opposite of the stamping procedure. The process of finding a circuit from given matrices is called *synthesizing* or *realizing*. So the unstamping procedure is a synthesizing method. It can be applied as described in the following.

First we find resistors and capacitors for the given symmetric matrices G and C. Assume the notations from above. Then the elements can be found by using (2.15) and (2.16) and similar equations for capacitors. For non-diagonal entries (2.15) gives

$$G_{n,m} = -g_{n,m} = -g_{m,n},$$

which are the elements, that are not connected to ground, i.e. connections from node n to node m (without implying a direction). The elements that connect node n to ground can be found from (2.16):

$$G_{n,0} = \sum_{k=1}^{N} g_{n,k} = \sum_{k=1}^{N} g_{k,n}$$

This can be done in an analogous manner for C.

Note, that the entries of G and C are not assumed to be as they would from an MNA formulation. So, there can be positive non-diagonal entries, that result in resistors and capacitors with negative parameter (value). The column and row sums can also be negative, with the same consequences. However, simulation tools usually do not have problems to simulate circuits with elements that have negative parameters.

The previous part dealt with resistors and capacitors. However, current sources can be added for a given incident matrix A_i, where the element in row j and column k is denoted by $a_{j,k} \in S$. According to its definition one current source will be placed for each column of A_i. If $a_{j,k} = 1$ the k^{th} current source is connected to node j and the input current flows into the source. For $a_{j,k} = -1$ it flows out of the source. The other end is connected to ground.

This synthesizing procedure is quite simple, but only applicable for RC circuits with current sources. We restrict our analysis on realizations of these kind of circuit, i.e. symmetric $\mathcal{A} = G$ and $\mathcal{E} = C$, and $\mathcal{B} = A_i$, where $a_{j,k} \in S$. For a generalization to RCL realizations, we refer the reader to [22, 10].

2.3.4 Sparse systems

We want to consider an important property of the systems yielded by MNA. It is called *sparsity*. A sparse matrix is a matrix with most entries zero. The system matrices $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{B}$ generated by the MNA approach are in many cases sparse, see (2.12): Obviously these matrices contain mostly blocks of zero matrices. Furthermore, even the block matrices C, L, G, A_L, A_V and A_i occurring in $\mathcal{E}$, $\mathcal{A}$ and $\mathcal{B}$ contain many zero elements (see (2.8)). This effect becomes even more apparent if we consider circuits with a larger number of elements. The number of non-zero entries and zeros in the system matrices of the example in Section 2.3.1 are given in the following table.

	non-zero entries	zeros
$\mathcal{B}$	2	10
$\mathcal{C}$	2	10
$\mathcal{A}$	14	22
$\mathcal{E}$	2	34

Note, that this example circuit is very small and thus the system matrices are very small, too. Still, the matrices are sparse.

A system is called sparse, if it has sparse system matrices. A sparse system is usually faster to simulate, than a dense (non-sparse) system and thus preferred in circuit simulation.

2.3.5 Permutation

Sometimes a change in node number ordering turns out to be useful. This can be done by operations directly on the system matrices thus avoiding a new stamping procedure. These operations are multiplications by permutation matrices. Let P be the permutation matrix (see [2]) corresponding to the reordering. By creating a matrix

$$\tilde{P} = \begin{pmatrix} P & 0 & 0 \\ 0 & I_{n_L} & 0 \\ 0 & 0 & I_{n_V} \end{pmatrix},$$

where I_{n_L} is the $n_L \times n_L$ identity matrix and I_{n_V} is the $n_V \times n_V$ identity matrix, the reordering can be done by left multiplying (2.10) by $\tilde{P}^T$ and changing the variable $x = \tilde{P}\tilde{x}$ in (2.10) and (2.10):

$$\begin{array}{ccccccc} & \tilde{P}^T \mathcal{E} \tilde{P} & \dot{\tilde{x}} & + & \tilde{P}^T \mathcal{A} \tilde{P} & \tilde{x} & = & \tilde{P}^T \mathcal{B} & u(t) \\ \Longleftrightarrow & \tilde{\mathcal{E}} & \dot{\tilde{x}} & + & \tilde{\mathcal{A}} & \tilde{x} & = & \tilde{\mathcal{B}} & u(t) \end{array}$$

$$y(t) \;=\; \mathcal{C}^T \tilde{P} \tilde{x} \;=\; (\tilde{P}^T \mathcal{C})^T \tilde{x} \;=\; \tilde{\mathcal{C}}\, \tilde{x},$$

where $\tilde{\mathcal{E}} = \begin{pmatrix} \tilde{C} & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{pmatrix}$, $\tilde{\mathcal{A}} = \begin{pmatrix} \tilde{G} & \tilde{A}_L & \tilde{A}_V \\ -\tilde{A}_L^T & 0 & 0 \\ -\tilde{A}_V^T & 0 & 0 \end{pmatrix}$, $\tilde{\mathcal{B}} = \begin{pmatrix} \tilde{A}_i & 0 \\ 0 & 0 \\ 0 & -I \end{pmatrix}$,

$$\tilde{x} = \begin{pmatrix} \tilde{u}_n \\ i_L \\ i_s \end{pmatrix},$$

$$\begin{array}{llll} \text{with} & \tilde{C} = P^T C\, P, & \tilde{A}_L = P^T A_L, & \tilde{A}_i = P^T A_i, \\ & \tilde{G} = P^T G\, P, & \tilde{A}_V = P^T A_V, & \tilde{u}_n = P^T u_n. \end{array}$$

The resulting system is the same system that would result from permuting first and applying a full new MNA afterwards. The operations done are called congruence transformation and will be defined in Section 3.2.

Example: As simple example assume an RC circuit with three nodes and no voltage source. The MNA of this circuit yields $\mathcal{A} = G, \mathcal{E} = C \in \mathbb{R}^{3\times 3}$ and $\mathcal{B} = A_i \in S^{3\times n_i}$. The permutation matrix P to renumber the nodes according to the following table

from	to
1	3
2	1
3	2

would be

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

Now assume a circuit with three nodes, which includes inductors and voltage sources. To renumber the nodes in the same manner as before, the same permutation matrix P could be used.

2.4 Modelling in the frequency domain: transfer functions

The transfer function H shows the behaviour of the system; it is the ratio of the outputs to the inputs in Laplace domain. It can be found, by first applying the Laplace transform to (2.10) and (2.11), which gives[2]

$$(\mathcal{E}\, s + \mathcal{A})\; X = \mathcal{B}\, U(s) \tag{2.17}$$

$$Y(s) = \mathcal{C}^T\, X, \tag{2.18}$$

where X, $U(s)$ and $Y(s)$ are the Laplace transforms of the vector of unknown states x, the vector of voltage and current sources $u(t)$ and the vector of outputs $y(t)$, respectively. The matrix pencil $\mathcal{A} + \lambda\,\mathcal{E}$ is assumed to be regular, i.e. $\mathcal{A} + \lambda\,\mathcal{E}$ is only singular for a finite number of $\lambda \in \mathbb{R}$. This is no big limitation to the circuits, that can be analyzed, see e.g. [9]. Now, (2.17) can be solved for X

$$X = (\mathcal{A} + s\,\mathcal{E})^{-1}\; \mathcal{B}\, U(s)$$

and then by (2.18)

$$Y(s) = \mathcal{C}^T\; (\mathcal{A} + s\,\mathcal{E})^{-1}\; \mathcal{B}\, U(s). \tag{2.19}$$

So, the transfer function is defined as

$$H(s) := \mathcal{C}^T\; (\mathcal{A} + s\,\mathcal{E})^{-1}\; \mathcal{B}. \tag{2.20}$$

The computation of $H(s)$ is called simulation in frequency domain and of course not done by calculating $(\mathcal{A} + s\,\mathcal{E})^{-1}$. However, it is still expensive for large systems.

Example: Just to give a tiny example, the well-known RC low-pass filter (see Figure 2.7) can be described with the system matrices next to Figure 2.7. Note, that since there is no inductor and no current source in this circuit, L, A_L and A_i as well as the corresponding zero matrices are not present (see [14]) in the system matrices.

If we choose the output to be the voltage at node 2, i.e.,

$$\mathcal{C} = (0\ 1\ 0)^T \qquad (\text{since } x = (u_1 \;\; u_2 \;\; i_{v_1})^T),$$

[2] assuming $x(0) = 0$

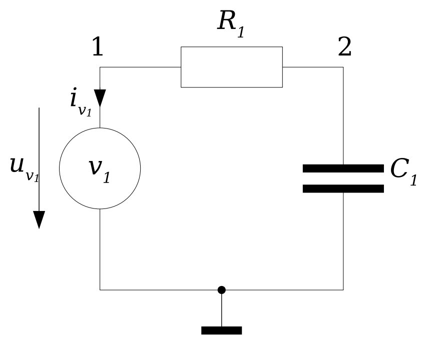


Figure 2.7: An RC low-pass with voltage source.

$$N = 2,\, n_V = 1,\, A_V = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$C = \begin{pmatrix} 0 & 0 \\ 0 & C_1 \end{pmatrix},\, G = \begin{pmatrix} G_1 & -G_1 \\ -G_1 & G_1 \end{pmatrix},$$

Therefore the system matrices are

$$\mathcal{E} = \left(\begin{array}{cc|c} 0 & 0 & 0 \\ 0 & C_1 & 0 \\ \hline 0 & 0 & 0 \end{array}\right),\, \mathcal{A} = \left(\begin{array}{cc|c} G_1 & -G_1 & 1 \\ -G_1 & G_1 & 0 \\ \hline -1 & 0 & 0 \end{array}\right),$$

$$\mathcal{B} = \left(\begin{array}{c} 0 \\ 0 \\ \hline -1 \end{array}\right).$$

the transfer function is the ratio of U_2 to U_1, i.e.,

$$H(s) = \frac{U_2(s)}{U_1(s)} = \mathcal{C}^T\, (\mathcal{A} + s\,\mathcal{E})^{-1}\, \mathcal{B} = (1 + s\,R_1\,C_1)^{-1}\,.$$

Y- and Z-parameters There are many types of transfer functions. Two important ones have Y- or Z-parameters as entries, respectively, which are admittance- or impedance-parameters, respectively. These are of minor importance for electrical engineers, but often used in model order reduction techniques. Let

$$\mathcal{C} = \mathcal{B}. \tag{2.21}$$

To get the Y-parameters as entries of transfer function matrix $H(s)$, only voltage sources have to be used and (2.21) must hold, then $y(t)$ is the output vector of currents flowing out of voltage sources and $u(t)$ the vector of source voltages, so $H(s)$ in $Y(s) = H(s)\,U(s)$ represents the Y-parameter matrix.

Similarly, if only current sources are used and (2.21) holds, $H(s)$ represents the Z-parameters. However, if (2.21) holds, but voltage *and* current sources are present in the circuit the transfer function entries are mixed. Then the entries represent voltage and current gains, impedance and admittance. These mixed parameters occur in the MNA example (circuit in Figure 2.6), because in (2.9) the outputs are chosen to be at the inputs (so (2.21) holds) and a current and a voltage source are used.

Example: If in the circuit shown in Figure 2.7 the voltage source is replaced by a current source, it can be illustrated quite easily how to find the Z-parameter. Note that there is no inductor and no voltage source in the circuit, so L, A_L, A_V and $-I$ are not present (see [14]) in the system matrices. Then $N = 2$, $n_i = 1$.

$$\mathcal{E} = \begin{pmatrix} 0 & 0 \\ 0 & C_1 \end{pmatrix}, \qquad \mathcal{A} = \begin{pmatrix} G_1 & -G_1 \\ -G_1 & G_1 \end{pmatrix}, \qquad \mathcal{C} = \mathcal{B} = A_i = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{aligned} H(s) &= \mathcal{B}^T \left(\mathcal{A} + s\,\mathcal{E}\right)^{-1} \mathcal{B} \\ &= (1 \;\; 0)\, \frac{1}{s\,C_1} \begin{pmatrix} 1 + s\,C_1 R_1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = R_1 + \frac{1}{s\,C_1} \end{aligned}$$

as expected for the impedance of a series RC circuit.

Passivity: *Passivity* is a property of a system or a circuit. Circuits consisting of R, C and L elements and independent sources are always passive. Such circuits consume electrical energy by transforming it into thermal energy (in resistors). A passive circuit yields a passive system. And in this work, only RCL circuits and thus only passive systems are considered.

In general, systems in the form of (2.10) and (2.11) with (2.21) are passive, if and only if $\mathrm{Re}(H(s))$ is positive semidefinite for all s with $\mathrm{Re}\ s > 0$ e.g.[7]. This implies all poles of H have got a negative or zero real part, i.e. $\mathcal{A} + s\,\mathcal{E}$ is singular only for s with $\mathrm{Re}\ s \leq 0$. The meaning of poles will be explained later in Section 3.4.

3 The Idea of Model Order Reduction

This chapter explains the idea of model order reduction (MOR). Firstly, reasons for the usage of MOR are given in Section 3.1. The section explains why MOR methods are desirable or even necessary in the topic of circuit simulations. Additionally some aspects of MOR methods are discussed. Section 3.2 explains how a model is reduced, generally, with the methods, described in this work. From this it can also be seen, that passivity of the models is preserved. Next, an overview of a widely used class of MOR methods (Krylov subspace methods) is given in Section 3.3. Properties of three methods will be compared and one simple example circuit reduced. Finally in Section 3.4, the so-called *modal approximation* (MA) method is described. This section provides a basis for the following chapter, which describes a method that uses MA.

3.1 Why reduction?

Mathematical models for circuits can be very large, especially for automatically generated circuits. The simulation of very large models can take an unreasonable amount of time or might even be unfeasible at all. Model order reduction techniques aim to reduce the simulation time or make a simulation feasible. These techniques reduce the order of a model, i.e. the number of unknowns and the dimension of the matrices. The model with the reduced matrices and vectors is called *reduced model*. In many cases the reduced model can then be simulated faster than the original model.

However, one has to consider the time needed for the reduction process. The reduction process followed by the simulation of the reduced system should take less time than the simulation of the original system alone. The time needed for the reduction process can be an aspect of the principle itself or the implementation.

Another for this work important aspect of MOR methods is the so-called *realizability* of the reduced model. Some MOR methods yield e.g. non-symmetric G and C blocks or a dense $\mathcal{B}$ matrix (see (2.10) on p. 14). Generally spoken, reduced models are not always *realizable*, i.e. there cannot be found a circuit from every reduced model. Even for the case that the reduced model is realizable, it might not be desirable to synthesize (find) a circuit from the model. The realized circuits from reduced models could e.g. require controlled sources while the original circuit did not have controlled sources. However, there exist MOR methods that yield reduced models, which are realizable by using the unstamping procedure (for RC circuits with current sources), described in Section 2.3.3. An advantage of realizable models is that these do not require any special handling in simulation, since they are like usual circuits.

Sparsity preservation is another important aspect of MOR methods, especially for realizable models. Sparsity preservation means, that MOR of a sparse model yields a reduced model, that is also sparse. This is specially important for realizable models, since the MOR method then not only reduces the order of the system, but also the number of circuit elements.

3.2 Congruence transformation and preservation of passivity

A property of MOR methods of major importance is the *preservation of passivity*. That means, a passive system yields a passive reduced model. All methods considered in this work are passivity preserving, since they use *congruence transformations*.

A congruence transformation of a square matrix $\mathcal{A} \in \mathbb{R}^{(N+n_V+n_i)\times(N+n_V+n_i)}$ is defined as $\hat{\mathcal{A}} = \mathcal{V}^T \mathcal{A} \mathcal{V}$, where $\mathcal{V}$ is referred to as the congruence transform [15]. The system matrices $\mathcal{A}$ and $\mathcal{E}$ from (2.10) can be reduced to order n by a congruence transformation in the following way:

- A non-square $\mathcal{V} \in \mathbb{R}^{(N+n_V+n_i)\times n}$ with rank $\mathcal{V} = n$ has to be found.
- A change of variable $x = \mathcal{V}\hat{x}$ is performed in the two MNA equations (2.10) and (2.11).
- Additionally the first MNA equation is left multiplied by $\mathcal{V}^T$.

That results in the reduced system with matrices

$$\begin{aligned} \hat{\mathcal{A}} &= \mathcal{V}^T \mathcal{A} \mathcal{V}, & \hat{\mathcal{B}} &= \mathcal{V}^T \mathcal{B}, \\ \hat{\mathcal{E}} &= \mathcal{V}^T \mathcal{E} \mathcal{V}, & \hat{\mathcal{C}} &= \mathcal{V}^T \mathcal{C}. \end{aligned}$$

Obviously $\hat{\mathcal{A}}$ and $\hat{\mathcal{E}}$ are the congruence transformations of $\mathcal{A}$ and $\mathcal{E}$, respectively. Since congruence transformation preserves symmetry and definiteness [15] of a matrix, passivity of the system is preserved. The symmetry is certainly only preserved if no voltage sources and inductors are used, since otherwise $\mathcal{A}$ is not symmetric.

3.3 Krylov subspace methods

A widely used class of model order reduction techniques is the class of Krylov subspace methods. We will only consider three methods, that operate on models yielded by MNA.

For a Krylov subspace method that operates on a different formulation, the reader is referred to [23]. There are many different Krylov subspace methods, but the overview will be given for three methods only without going into the details of them.

PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm (PRIMA) [16] is designed for reduction of RCL interconnect circuits. We assume either current sources or voltage sources are used. PRIMA mainly consists (usually) of a Block Arnoldi algorithm [1, 4] that builds an orthogonal basis matrix $\mathcal{V}$ for a Krylov subspace. This matrix $\mathcal{V}$ is used as congruence transform (see Section 3.2) and thus the method preserves passivity, as is proved (for (2.21)) in [16].

Unfortunately, it does not preserve any structural aspects. The reduced input matrix $\hat{\mathcal{B}} = \mathcal{V}^T \mathcal{B}$ is (usually) dense, because $\mathcal{B}$ (one entry per column) is left multiplied by a usually dense matrix. $\hat{\mathcal{C}}$ is dense for the same reasons. Thus a realization cannot be found by unstamping.

The PRIMA approximation is not as accurate, as it could be for q iterations of the Arnoldi algorithm. Only the first q matrix coefficients of the Taylor expansion in Laplace domain (which are called block moments) of the transfer function of the reduced model

$$\hat{H}(s) = \hat{\mathcal{B}}^T \left(\hat{\mathcal{A}} + s\,\hat{\mathcal{E}}\right)^{-1} \hat{\mathcal{B}} = \sum_{k=0}^{\infty} \hat{M}_k\, s^k \tag{3.1}$$

are the same as the block moments of the original transfer function. One can say, q block moments are matched.

SPRIM, Structure-Preserving Reduced-Order Interconnect Macromodeling, [5, 7] is based on PRIMA but overcomes some of its disadvantages. SPRIM matches $2\,q$ block moments for a q^{th}-order approximation (q iterations) and thus SPRIM is more accurate than PRIMA. Additionally SPRIM preserves the block structure of the system matrices as well as reciprocity (symmetry of G, C and L). In contrast to PRIMA, SPRIM partitions the projection matrix according to the block structure of the system matrices (assuming no voltage sources are present, otherwise see [7]):

$$\mathcal{V} = \begin{pmatrix} \mathcal{V}_1 \\ \mathcal{V}_2 \end{pmatrix}, \qquad \text{where } \mathcal{V}_1 \in \mathbb{R}^{N\times n} \text{ and } \mathcal{V}_2 \in \mathbb{R}^{n_L\times n}.$$

Then, if $r_1 = \operatorname{rank}\mathcal{V}_1 < n$ or $r_2 = \operatorname{rank}\mathcal{V}_2 < n$ (i.e. one or both do not have full column rank) $\tilde{\mathcal{V}}_1 \in \mathbb{R}^{N\times r_1}$ and $\tilde{\mathcal{V}}_2 \in \mathbb{R}^{n_L\times r_2}$ can be defined with $\operatorname{colspan}\tilde{\mathcal{V}}_1 = \operatorname{colspan}\mathcal{V}_1$ and colspan $\tilde{\mathcal{V}}_2 =$ colspan $\mathcal{V}_2$ to build the projection matrix

$$\tilde{\mathcal{V}} = \begin{pmatrix} \tilde{\mathcal{V}}_1 & 0 \\ 0 & \tilde{\mathcal{V}}_2 \end{pmatrix}.$$

Then a congruence transformation of the MNA system with congruence transform $\tilde{\mathcal{V}}$ yields for the respective blocks (compare to (2.10))

$$\tilde{G} = \tilde{\mathcal{V}}_1^T \, G \, \tilde{\mathcal{V}}_1, \qquad \tilde{L} = \tilde{\mathcal{V}}_2^T \, L \, \tilde{\mathcal{V}}_2,$$

$$\tilde{C} = \tilde{\mathcal{V}}_1^T \, C \, \tilde{\mathcal{V}}_1, \qquad \tilde{A}_L = \tilde{\mathcal{V}}_1^T \, A_L \, \tilde{\mathcal{V}}_2, \qquad \tilde{A}_i = \tilde{\mathcal{V}}_1^T \, A_i.$$

The reduced system with above described properties is achieved. However, it does not preserve the input-output structure of the system ($\mathcal{B}$ has changed its structure similarly, as in PRIMA) and thus the reduced system is not synthesizable by unstamping.

IOPOR, which stands for Input Output structure Preserving Order Reduction [22] is a technique, which can use the projection matrix of SPRIM (then the method can be called SPRIM/IOPOR). For IOPOR with a different projection matrix, we refer the reader to [21, 22]. We restrict our analysis to SPRIM/IOPOR and then this method is as accurate as SPRIM. The analysis is furthermore restricted to current sources only, again. In the node numbering scheme used for MNA, in IOPOR the current sources are assumed to be connected to the nodes with the first numbers (see also Section 2.3.5). This allows further partitioning of the current source incidence matrix, since the only non-zero entries are in the first rows:

$$A_i = \begin{pmatrix} A_{i_1} \\ 0 \end{pmatrix}, \qquad \text{where } A_{i_1} \in \mathbb{R}^{n_i \times n_i} \tag{3.2}$$

Now the projection matrix of SPRIM is further partitioned, according to the partitioning of A_i:

$$\tilde{\mathcal{V}}_1 = \begin{pmatrix} \mathcal{V}_{1,1} \\ \mathcal{V}_{1,2} \end{pmatrix}, \qquad \text{where } \mathcal{V}_{1,1} \in \mathbb{R}^{n_i \times r_1} \text{ and } \mathcal{V}_{1,2} \in \mathbb{R}^{(N-n_i) \times r_1}.$$

Then the SPRIM/IOPOR projection matrix is formed by

$$\mathring{\mathcal{V}} = \begin{pmatrix} \mathcal{W} & 0 \\ 0 & \tilde{\mathcal{V}}_2 \end{pmatrix} = \begin{pmatrix} I & 0 & 0 \\ 0 & \tilde{\mathcal{V}}_{1,2} & 0 \\ 0 & 0 & \tilde{\mathcal{V}}_2 \end{pmatrix}, \tag{3.3}$$

where I is the $n_i \times n_i$ identity matrix and $\tilde{\mathcal{V}}_{1,2}$ is the orthogonalized $\mathcal{V}_{1,2}$.

Then, by performing a congruence transformation with $\mathring{\mathcal{V}}$ the input-output structure is preserved, as can be seen by considering the block structure.

$$\tilde{G} = \mathcal{W}^T G \mathcal{W}, \qquad \tilde{L} = \tilde{\mathcal{V}}_2^T L \tilde{\mathcal{V}}_2,$$

$$\tilde{C} = \mathcal{W}^T C \mathcal{W}, \qquad \tilde{A}_L = \mathcal{W}^T A_L \tilde{\mathcal{V}}_2, \qquad \tilde{A}_{i_1} = A_{i_1}.$$

The input-output structure is preserved in the reduced model, since $\tilde{A}_i = (A_{i_1} \quad 0)^T$ has the same structure as A_i. Thus the reduced model can easily be synthesized (by unstamping). However, negative values can occur, but that is usually not a problem for simulation tools. One drawback of this method is, that the blocks will be dense. An example will illustrate the density of the blocks in the reduced model.

Example: An RC circuit, shown in Figure 3.1, serves as example to illustrate a reduced model of the SPRIM/IOPOR method.

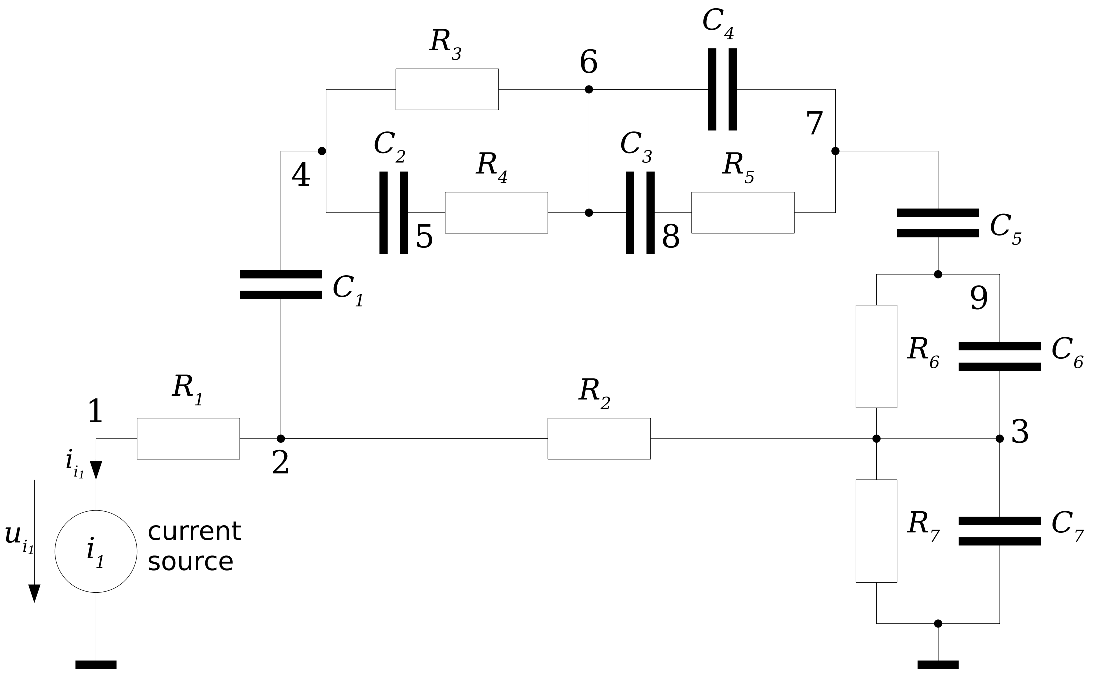


Figure 3.1: RC circuit with nine numbered nodes, seven resistors and capacitors, and one current source, where $G_k = R_k^{-1} = k\,\mathrm{S}$ ($\mathrm{S} = \Omega^{-1}$) and $C_k = k \cdot 10^{-3}\,\mathrm{F}$ for $k = 1, \ldots, 7$.

The MNA procedure (e.g. by stamping) applied to the circuit in Figure 3.1 yields the corresponding model. Its system matrices $\mathcal{A} = G$ and $\mathcal{E} = C$ are

$$G = \left(\begin{array}{c|cccccccc} G_1 & -G_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -G_1 & G_1 + G_2 & -G_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -G_2 & G_2 + G_6 + G_7 & 0 & 0 & 0 & 0 & 0 & -G_6 \\ 0 & 0 & 0 & G_3 & 0 & -G_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & G_4 & -G_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -G_3 & -G_4 & G_3 + G_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & G_5 & -G_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -G_5 & G_5 & 0 \\ 0 & 0 & -G_6 & 0 & 0 & 0 & 0 & 0 & G_6 \end{array}\right), \tag{3.4}$$

$$C = \left(\begin{array}{c|cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & C_1 & 0 & -C_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_6 + C_7 & 0 & 0 & 0 & 0 & 0 & -C_6 \\ 0 & -C_1 & 0 & C_1 + C_2 & -C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -C_2 & C_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_3 + C_4 & -C_4 & -C_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & -C_4 & C_4 + C_5 & 0 & -C_5 \\ 0 & 0 & 0 & 0 & 0 & -C_3 & 0 & C_3 & 0 \\ 0 & 0 & -C_6 & 0 & 0 & 0 & -C_5 & 0 & C_5 + C_6 \end{array}\right) \tag{3.5}$$

and $\mathcal{B} = A_i = (1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)^T$. By the partitioning (3.2), $A_{i_1} = 1 \in \mathbb{R}$ and so the identity matrix I in (3.3) is also 1×1 (i.e. $I = 1$).

For the computation the SPRIM/IOPOR technique was implemented (see Appendix on CD). Here we used the description of the algorithm as given in [10]. The general IOPOR technique is presented in [22]. The description proposes to use the projection matrix of PRIMA [16]. To generate this projection matrix, PRIMA is not bound to a particular Arnoldi process as stated in [16]. We decided to use the block-Arnoldi algorithm described in [4, Algorithm 1]. However, other implementations are possible. An expansion point $s_0 \in \mathbb{R}$ can also be used, as stated in [5]. For details on this method we refer the reader to the references, since this work focuses on sparsity preserving methods (see Chapter 4).

Now the reduction to three nodes (including one node for the terminal) with expansion point chosen[1] $s_0 = 2\,\pi \cdot 200\,\mathrm{Hz}$ yields the following system matrices (here shown in fixed

[1] the expansion point was chosen by trial and error by considering the transfer function

notation with 4 decimals):

$$\tilde{G} = \left(\begin{array}{c|cc} 1.0000 & -0.7382 & 0.0009 \\ \hline -0.7382 & 1.4668 & -0.6656 \\ 0.0009 & -0.6656 & 6.6079 \end{array}\right), \quad \tilde{C} = \left(\begin{array}{c|cc} 0 & 0 & 0 \\ \hline 0 & 0.0003 & 0.0005 \\ 0 & 0.0005 & 0.0021 \end{array}\right),$$

where $\tilde{G}$ is in S (Siemens, also known as mho = $\mho$) and $\tilde{C}$ in F. Furthermore $\tilde{A}_i = (1\ 0\ 0)^T$.

It can be seen, that the upper left entry in $\tilde{G}$, $\tilde{C}$ and $\tilde{A}_i$ has been preserved ($G_1 = 1\,\mathrm{S}$). Thus the structure of the input-output matrix has been preserved in the reduced model. However, the remaining parts in $\tilde{G}$ and $\tilde{C}$ are dense, i.e. there are more non-zero entries than zero entries .

By unstamping of this reduced model, a reduced circuit with six resistors and three capacitors can be found. The values of resistors and capacitors are:

$$\begin{array}{lclclclclcl} R_{1,0} & = & 3.8070\,\Omega & & R_{1,2} & = & 1.3546\,\Omega & & C_{2,0} & = & 800.83\,\mu\mathrm{F} \\ R_{2,0} & = & 15.8976\,\Omega & & R_{2,3} & = & 1.5023\,\Omega & & C_{3,0} & = & 2.597\,\mathrm{mF} \\ R_{3,0} & = & 0.1683\,\Omega & & R_{1,3} & = & -1128.4\,\Omega & & C_{2,3} & = & -527.98\,\mu\mathrm{F} \end{array}$$

The current source is still connected to node one, since the structure of the input-output matrix did not change.

As described and illustrated, this method is not sparsity preserving. Therefore, this method might not always be the best choice (e.g. for large, sparse models). However, there are applications, where densly connected circuits yield dense models. A technique which extracts dense models from three-dimensional geometries (e.g. integrated circuit packages) is called *partial element equivalent circuit* (PEEC) method [20]. For these models this method does not have the disadvantage that sparsity is lost during reduction. So electrical circuits (or models in general) suitable for reduction with IOPOR could be identified by the sparsity of the model.

3.4 Modal approximation

Modal approximation (MA), also known as modal model reduction, is another MOR method. This section explains the principle and serves as a basis for the next chapter. Because in the next chapter only RC circuits with current sources will be analyzed, the analysis is restricted in this section to RC circuits with current sources, too. Additionally, we assume (2.21) holds. That means, the circuit is in Z-parameter form, i.e. has got as output matrix the input matrix. So,

$$\mathcal{A} = G, \qquad \mathcal{E} = C \qquad \text{and} \qquad \mathcal{B} = \mathcal{C} = A_i. \tag{3.6}$$

Note that G and C are real symmetric $N \times N$ matrices, $A_i \in \{-1, 0, 1\}^{N \times n_i}$ yielded by MNA of a passive circuit and thus the system is passive. Then the transfer function H from (2.20) can be rewritten as

$$H(s) = A_i^T \, (G + s\, C)^{-1} \, A_i \tag{3.7}$$

Now a pole of the transfer function H is defined as a $\lambda \in \mathbb{R}_{\leq 0}$, for which

$$\lim_{s \to \lambda} \|H(s)\|_2 = \lim_{s \to \lambda} \sigma_{\max}(H(s)) = \infty \tag{3.8}$$

(where $\sigma_{\max}(H(s))$ is the largest singular value of $H(s)$) holds. By looking at (3.7) it can be seen, that this happens for s, where $G + s\, C$ becomes singular. $G + \lambda_k\, C$ is singular if there exists a non-trivial solution for v_k in

$$0 = (G + \lambda_k\, C)\, v_k \iff -G\, v_k = C\, v_k\, \lambda_k. \tag{3.9}$$

This is known as the *generalized eigenvalue problem* of $(-G, C)$.

The generalized eigenvalue problem of $(-G, C)$ of the square matrices G and C is the generalization of the eigenvalue problem of $-G$. If $C = I$ the problems are the same. The matrix pencil $G + \lambda\, C$ is again assumed to be regular, i.e. $G + \lambda\, C$ is assumed to be singular only for a finite number of λ. Then the generalized eigenvalues and generalized eigenvectors are the solutions of (3.9). This problem can be rewritten like

$$-\, G\, V = C\, V\, \Lambda \tag{3.10}$$

where the k^{th} column of V is the generalized eigenvector v_k corresponding to the generalized eigenvalue $\lambda_k = \Lambda_{k,k}$, which is the k^{th} diagonal entry in the diagonal matrix Λ. Note that the (generalized) eigenvalues are the poles of the system and the corresponding eigenvectors are called *modes*.

Let us assume there are $n_e \leq N$ distinct eigenvalues. V can be scaled, such that $V^T\, C\, V = I$, which induces by (3.10) that $V^T\, G\, V = -\Lambda$. Alternatively the eigenvectors can be normalized to unit length, i.e. $\|v_k\|_2 = 1$, which will be later referred to. Here, the first scaling is assumed. Then, by using V as congruence transform (see Section 3.2),

$$V^T\, G\, V = -\Lambda, \qquad V^T\, C\, V = I \quad \text{and} \quad \hat{A}_i = V^T\, A_i.$$

Hence,

$$H(s) = \hat{A}_i^T \, (I\, s - \Lambda)^{-1} \, \hat{A}_i, \tag{3.11}$$

which is no approximation, since all eigenvectors were used. Now, the transfer function can be expressed differently, to define the term *residue*. $I\,s - \Lambda$ is diagonal, so

$$H(s) = \hat{A}_i^T \begin{pmatrix} (s-\lambda_1)^{-1} & & \\ & \ddots & \\ & & (s-\lambda_{n_e})^{-1} \end{pmatrix} \hat{A}_i \tag{3.12}$$

$$\begin{aligned} &= A_i^T\,V \sum_{k=1}^{n_e} \frac{e_k\,e_k^T}{s-\lambda_k}\,V^T\,A_i = \sum_{k=1}^{n_e} \frac{\left(A_i^T\,V\,e_k\right)\left(A_i^T\,V\,e_k\right)^T}{s-\lambda_k} \\ &= \sum_{k=1}^{n_e} \frac{q_k\,q_k^T}{s-\lambda_k} = \sum_{k=1}^{n_e} \frac{R_k}{s-\lambda_k} \end{aligned} \tag{3.13}$$

where e_k is the k^{th} unit vector, the residues $R_k = q_k\,q_k^T$ and $q_k = A_i^T\,v_k$ with v_k as the k^{th} column vector of V. To approximate H some of the summands are simply discarded.

Certainly the question arises, which summands should be used and which not. This can be decided by defining a criterion for selecting the summands, or alternatively for selecting the poles since each pole is related to one summand. This is called *dominance criterion* and the poles, that satisfy the criterion are called *dominant poles*. However, there are several different definitions for dominant poles, depending on the application (see [18] for references). Here, we will use a similar definition as in [17, Chapter 4], unless otherwise stated: Poles at zero ($\lambda_k = 0$) are defined to be dominant. Furthermore for $\lambda_k \neq 0$, we define its *dominance value* as the upper bound for the gain of a summand by

$$\hat{R}_k := \frac{\|R_k\|_2}{|\lambda_k|} = \frac{\sigma_{\max}(R_k)}{|\lambda_k|} \tag{3.14}$$

(λ_k is real). Now, as the summands corresponding to the largest dominance value can give the largest contribution to the sum, it is a natural choice to leave them in the sum. Hence, one can define an $R_{\min}$ and all poles λ_k that satisfy the dominance criterion $\hat{R}_k \geq R_{\min}$ are called dominant poles (in addition to the pole at zero).

To write the approximated transfer function, we assume, there are n_d dominant poles and these are $\lambda_1, \ldots, \lambda_{n_d}$. Then the approximated transfer function is the sum of summands corresponding to the dominant poles:

$$\tilde{H}(s) = \sum_{k=1}^{n_d} \frac{R_k}{s-\lambda_k}. \tag{3.15}$$

Furthermore,

$$\tilde{V} := (v_1, \ldots, v_{n_d}) \quad \text{and} \quad \tilde{\Lambda} := \operatorname{diag}(\lambda_1, \ldots, \lambda_{n_d}).$$

So, by comparing (3.15) to the result (3.13) and prior equations, it can be seen, that the approximated (or reduced) transfer function $\tilde{H}(s)$ can be written as

$$\tilde{H}(s) = \tilde{A}_i^T \, (I\,s - \tilde{\Lambda})^{-1} \, \tilde{A}_i,$$

where

$$-\tilde{\Lambda} = \tilde{G} = \tilde{V}^T \, G \, \tilde{V}, \qquad \tilde{C} = \tilde{V}^T \, C \, \tilde{V} \quad \text{and} \quad \tilde{A}_i = \tilde{V}^T \, A_i. \tag{3.16}$$

The matrices (3.16) are the reduced system matrices and given by the congruence transformation of the original matrices with congruence transform $\tilde{V}$.

Obviously, the structure of A_i is not preserved, since $\tilde{V}$ is generally dense. So the modal equivalent is not realizable by unstamping. The circuit would require controlled sources [11, 12]. Additionally, the *DC offset* is not preserved, generally. With DC offset we mean $H(0)$, if it exists. An MOR technique, which uses modal approximation and yields realizable models is described and discussed in the next chapter.

Remark: It is important to say, that a full eigendecomposition (calculation of the matrices V and Λ in (3.10)) is in practical systems not feasible. Usually the applications are very large models of circuits (or systems generally). Therefore, numerical eigensolvers are used, to compute the dominant poles (and modes). In [17] new algorithms to find dominant poles are presented and extensively discussed. One of them, the Subspace Accelerated MIMO Dominant Pole Algorithm[2] (SAMDP) [17, 19] is capable to efficiently compute a set of dominant poles of a MIMO system. For more detailed information we want to refer to the referenced papers, but a closer look will be taken on the conditions when the algorithm can be applied. This will be of importance for the next Chapter, too.

The definition of a pole (3.8) can be used to form an equivalent statement for square transfer functions [17, Section 4.3.1]:

$$\lim_{s \to \lambda} \underbrace{\lambda_{\min}(H(s)^{-1})}_{=:f(s)} = 0. \tag{3.17}$$

This definition obviously assumes a regular $H(s)$ (except for a finite number of s). Then the Newton scheme for the function f can be used to find its roots, which are the poles of H. For this Newton scheme, the derivative of H is used. SAMDP is based on this scheme. For more details, see [17, Section 4.3].

It is also stated in [17] that it is numerically more stable for SAMDP to scale the eigenvectors v_k to unit length, i.e. $\|v_k\| = 1$, than by $\tilde{V}^T \, C \, \tilde{V} = I$. However, throughout this work the scaling $\tilde{V}^T \, C \, \tilde{V} = I$ is used for the sake of simplicity.

[2] Matlab® implementation available online at http://sites.google.com/site/rommes/software

4 Model Order Reduction with SparseMA

Sparse modal approximation (SparseMA) is a model order reduction method and presented in [12], by Joost Rommes and Roxana Ionutiu. This chapter describes the method and how it is applied. Additionally an example as well as some practical aspects are given.

Some assumptions have to be made, similar to these made in Section 3.4. SparseMA is capable to reduce RC circuits with current sources (without inductors and and voltage sources), so (3.6) is assumed. Furthermore, the circuit has to be in Z-parameter form, i.e. the output matrix and input matrix must be equal ((2.21) holds).

Roughly summarized, SparseMA partitions the circuit into two parts. The nodes in one part will be preserved (not reduced) and the other part reduced by modal approximation.

This chapter describes in the first sections the principle. Section 4.1 describes which nodes should be selected as the nodes that will be preserved. The initial reasons for preservation are explained, too. These reasons change afterwards, when the kind of circuits is considered, that the reduced models yield. According to the selected nodes, the system matrices will be partitioned, as described in Section 4.2. Then one part is reduced by modal approximation. In Section 4.3 the differences to conventional modal approximation that arise by the partitioning are discussed. Section 4.4 illustrates the kind of circuits realized from the reduced models. This brings insights for possible selection strategies, which preserve the DC offset. Finally, in Section 4.5 some practical aspects are highlighted. A practical method to find the dominant poles is described, which is slightly different than in conventional MA. Additionally a negative example should give the reader a general idea, that this method is not applicable for every circuit.

4.1 Selection of nodes to preserve

The first step of SparseMA is to select some nodes, that will not be reduced. These nodes are referred to as *selected nodes*. Which nodes will be the selected nodes, depends on the selection strategy. The idea is that the *remaining nodes* (i.e. unselected nodes) should be “not so important” in comparison to the selected nodes. However, this must be regarded with respect to the effects of the reduction. The reduction is done with modal approximation and a property of this method is, that the reduction does not preserve the DC offset. So, one could think, if the nodes that are responsible for the DC offset are selected to be preserved during reduction, the DC offset will be preserved with them. We present now two selection strategies, based on this thought. It appears [13] that a strategy similar to the following strategy 2 was used in [12]. Later in Section 4.4 we can explain, why their strategy must be different from strategy 2. Additionally we show in Section 4.4, why one of the following strategies fails to preserve the DC offset and what other strategies could be possible.

Strategy 1: Only the nodes are selected to be preserved, that are directly on the DC path (connected by resistors) from terminals to ground.

Strategy 2: All nodes connected to the terminals by resistors are selected to be preserved.

A simple example should illustrate these strategies. Figure 4.1 is basically the circuit of Figure 3.1, but with a different node numbering. This reordering of the numbers is done already for the next step (Section 4.2). The selected nodes have to get assigned the last numbers. Alternatively a permutation (Section 2.3.5) can be done to get the same effect in the MNA equations.

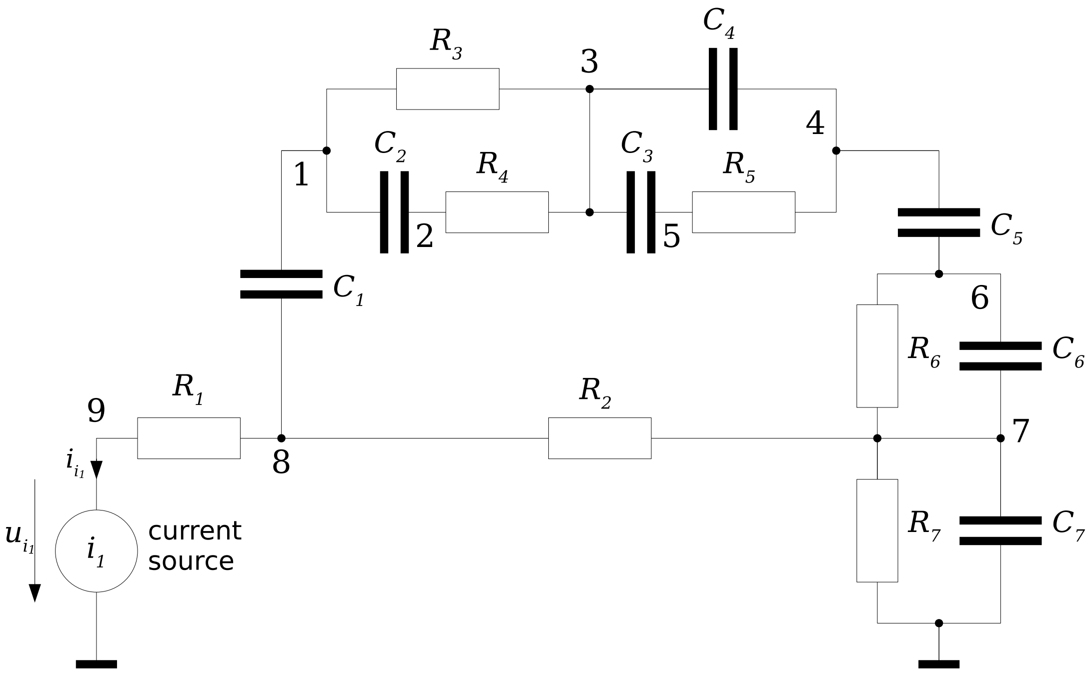


Figure 4.1: Circuit of Figure 3.1 with different ordering of the nodes.

Anyway, strategy 1 would declare nodes 9, 8 and 7 as selected nodes, because these nodes are on the DC path to ground. DC path to ground means: start from the terminal, go directly all possible ways to ground node only by resistors. The term “all possible ways” is used, because there can be more than one node connected to ground by a resistor. The ones that are also connected to (at least) one terminal by resistors would belong to the set of selected nodes (including the nodes on the way from terminal to ground). Anyway, consequently 1, 2, 3, 4, 5 and 6 would be declared as *remaining nodes* (which will be reduced) in strategy 1. One can see that the coupling of the selected nodes to the remaining nodes is done by C_1, C_6 and R_6. Later, we will see, that this strategy will fail to preserve the DC offset, but this can only be understood by taking a look on the reduced circuit.

Strategy 2, on the other hand, will preserve the DC offset. The difference is, that strategy 2 selects *all* nodes connected by resistors to the terminals. So, the number of the selected nodes of strategy 2 is greater or equal to the number of selected nodes of strategy 1. In the example circuit in Figure 4.1 nodes 9, 8, 7 and 6 would be selected by strategy 2. In general the coupling of the selected nodes to the remaining nodes is only done by capacitors in this strategy. This is, because if a node connected by a resistor to a selected node it would also belong to the set of selected nodes.

In the following we will *not* assume a specific strategy. So all theory holds also for other strategies, that could be thought of. Later (in Section 4.5) we discuss the effect of the selection strategy on the implementation. In practical applications, the number of selected nodes is much smaller than the number of remaining nodes.

4.2 Partitioning

Now, that the selected nodes are determined, the partitioning of the system (MNA equations) can be performed. It serves for two purposes. On the one hand it is for the preservation of the input-output structure. This is realized, by the fact that the terminal nodes are always included in the set of selected nodes (in every strategy). On the other hand the system is partitioned to preserve the DC offset.

As a basic requirement for partitioning the MNA equations, the nodes have to be numbered in a special order. The numbering of the nodes can be changed by a permutation of the system matrices, see Section 2.3.5. The selected nodes must be the nodes with the last numbers. The terminal nodes, which are included in the selected nodes, can get assigned the very last numbers. Note, that it is not required for SparseMA that the terminal nodes get assigned the *very* last numbers, but it gives a more consistent ordering. This can be expressed by the following partitions. Firstly A_i must be partitionable in the following way

$$A_i = \begin{pmatrix} 0 \\ A_{i_2} \end{pmatrix},$$

where $A_{i_2} \in S^{n_i \times n_i}$ are the non-zero rows of A_i. That means, the terminals are the last nodes in the numbering scheme. This is similar to the required partitioning of SPRIM/IOPOR (see Section 3.3 and [22]) – but terminal nodes are the first nodes in SPRIM/IOPOR instead of the last nodes as here, in SparseMA.

Secondly, the other selected nodes have the numbers directly prior to the numbers of the terminal nodes. Together, let the selected nodes yield a total number of $n_S > n_i$ nodes. Then, with the number of remaining nodes $n_R := N - n_S$, the system matrices G, C and A_i can be partitioned [12]:

$$G = \begin{pmatrix} G_R & G_K \\ G_K^T & G_S \end{pmatrix}, \qquad C = \begin{pmatrix} C_R & C_K \\ C_K^T & C_S \end{pmatrix} \quad \text{and} \quad A_i = \begin{pmatrix} 0 \\ A_{i_S} \end{pmatrix}, \tag{4.1}$$

where the matrices of the mutual connections of the remaining nodes $G_R, C_R \in \mathbb{R}^{n_R \times n_R}$, the matrices of the mutual connections of the selected nodes $G_S, C_S \in \mathbb{R}^{n_S \times n_S}$, the matrices for the coupling of selected nodes to the remaining nodes $G_K, C_K \in \mathbb{R}^{n_R \times n_S}$ and the incident matrix of the current sources $A_{i_S} \in S^{n_S \times n_i}$.

Then, by partitioning $x = (x_R \quad x_S)^T$, where $x_R \in \mathbb{R}^{n_R}$ and $x_S \in \mathbb{R}^{n_S}$ and using the partitioning of (4.1), the first MNA equation (2.10) becomes

$$\begin{pmatrix} C_R & C_K \\ C_K^T & C_S \end{pmatrix} \begin{pmatrix} \dot{x}_R \\ \dot{x}_S \end{pmatrix} + \begin{pmatrix} G_R & G_K \\ G_K^T & G_S \end{pmatrix} \begin{pmatrix} x_R \\ x_S \end{pmatrix} = \begin{pmatrix} 0 \\ A_{i_S} \end{pmatrix} u(t) \tag{4.2}$$

and in Laplace domain with Laplace transforms[1] of x_R, x_S and $u(t)$ being X_R, X_S and $U(s)$, respectively

$$\left(\begin{pmatrix} G_R & G_K \\ G_K^T & G_S \end{pmatrix} + s \begin{pmatrix} C_R & C_K \\ C_K^T & C_S \end{pmatrix} \right) \begin{pmatrix} X_R \\ X_S \end{pmatrix} = \begin{pmatrix} 0 \\ A_{i_S} \end{pmatrix} U(s). \tag{4.3}$$

Assuming a regular matrix pencil $G_R + s\,C_R$, the first (upper) part of the system (4.3) can be solved for X_R:

$$\begin{aligned} &(G_R + s\,C_R)\,X_R + (G_K + s\,C_K)\,X_S = 0 \\ \Longleftrightarrow &X_R = -(G_R + s\,C_R)^{-1}\,(G_K + s\,C_K)\,X_S \end{aligned}$$

[1] assuming $x_R(0) = 0$ and $x_S(0) = 0$

By inserting X_R into the second (lower) part of (4.3) and then solving for X_S:

$$\begin{aligned} & (G_S + s\,C_S)\,X_S + (G_K^T + s\,C_K^T)\,X_R && = A_{i_S}\,U(s) \\ \Longleftrightarrow \quad & (\,\underbrace{(G_S + s\,C_S)}_{=:Y_S(s)} - \underbrace{(G_K + s\,C_K)^T\,(G_R + s\,C_R)^{-1}\,(G_K + s\,C_K)}_{=:Y_R(s)}\,)\,X_S && = A_{i_S}\,U(s) \\ \Longleftrightarrow \quad & X_S = (Y_S(s) - Y_R(s))^{-1} A_{i_S}\,U(s). \end{aligned} \tag{4.4}$$

Let $Y_H(s) = Y_S(s) - Y_R(s)$. Then the transfer function is given in the output equation by inserting X_S into the second MNA equation (2.18):

$$Y(s) = A_i^T\,X = (0 \quad A_{i_S}^T) \begin{pmatrix} X_R \\ X_S \end{pmatrix} = A_{i_S}^T\,X_S = \underbrace{A_{i_S}^T\,(Y_H(s))^{-1}\,A_{i_S}}_{=H(s)}\,U(s).$$

So, the partitioning allows for a different form of the transfer function. The next section explains how MA can be applied, such that the selected nodes will be preserved.

4.3 Modal approximation in SparseMA

The different form of $H(s)$ enables us a way to reduce only a part of the system. Firstly, instead of computing the poles of H as in conventional modal approximation, the poles of Y_R are computed. By comparing the expression for $Y_R(s)$ (4.4) and $H(s)$ in (3.7) (p. 30) one can see, that the poles of $Y_R(s)$ can be found, by considering $G_R + s\,C_R$. The poles are now the s such that $G_R + s\,C_R$ is singular and can be computed from the generalized eigenvalue problem of $(-G_R, C_R)$ (compare to Section 3.4).

These poles can be used to write $Y_R(s)$ as sum, similarly as done for $H(s)$ in Section 3.4:

$$Y_R(s) = \sum_{k=1}^{n_e} \frac{R_k(s)}{s - \lambda_k} \tag{4.5}$$

where the residues $R_k(s) = q_k(s)\,q_k(s)^T$ and $q_k(s) = (G_K + s\,C_K)^T\,v_k$ with v_k as the eigenvector corresponding to the eigenvalue λ_k. Note, that where A_i was in conventional MA in the vector q_k (see (3.13), p. 31) is now $G_K + s\,C_K$ in the vector $q_k(s)$, which is dependent on s. Therefore the residues $R_k(s)$ are dependent on s, too. So the dominance value from (3.14) (p. 31) changes to a frequency dependent version. There might be other plausible definitions, but we will use the following.

$$\hat{R}_k := \frac{\|R_k(s_0)\|_2}{|\lambda_k|}, \quad \text{where } s_0 = i\,\omega, \quad \omega \in \mathbb{R} \tag{4.6}$$

ω can be chosen appropriately as the frequency, for which the model is used for. Later in Section 4.5 we will consider a special case, where no frequency has to be chosen.

With the definition of the dominance values (4.6) we can now approximate Y_R with n_d dominant poles:

$$\tilde{Y}_R(s) = \sum_{k=1}^{n_d} \frac{R_k(s)}{s - \lambda_k}. \tag{4.7}$$

Let

$$\tilde{V} := (v_1, \ldots, v_{n_d}) \quad \text{and} \quad \tilde{\Lambda} := \operatorname{diag}(\lambda_1, \ldots, \lambda_{n_d}).$$

By comparing the approximation $\tilde{Y}_R(s)$ to $\tilde{H}(s)$ in (3.15) one can rewrite $\tilde{Y}_R(s)$ for the same reasons $\tilde{H}(s)$ was rewritten in a similar way:

$$\tilde{Y}_R(s) = (\tilde{G}_K + s\,\tilde{C}_K)^T\,(I\,s - \tilde{\Lambda})^{-1}(\tilde{G}_K + s\,\tilde{C}_K) \tag{4.8}$$

where

$$\begin{aligned} \tilde{G}_R &:= \tilde{V}^T\,G_R\,\tilde{V} = -\tilde{\Lambda}, & \qquad \tilde{G}_K &= \tilde{V}^T\,G_K, \\ \tilde{C}_R &:= \tilde{V}^T\,C_R\,\tilde{V} = I, & \qquad \tilde{C}_K &= \tilde{V}^T\,C_K. \end{aligned}$$

These matrices are the reduced system blocks and give the reduced partitioned model

$$\underbrace{\begin{pmatrix} \tilde{G}_R & \tilde{G}_K \\ \tilde{G}_K^T & G_S \end{pmatrix}}_{=:\tilde{G}} \begin{pmatrix} \tilde{X}_R \\ X_S \end{pmatrix} + s \underbrace{\begin{pmatrix} \tilde{C}_R & \tilde{C}_K \\ \tilde{C}_K^T & C_S \end{pmatrix}}_{=:\tilde{C}} \begin{pmatrix} \tilde{X}_R \\ X_S \end{pmatrix} = \begin{pmatrix} 0 \\ A_{i_S} \end{pmatrix} U(s) \tag{4.9}$$

and

$$Y(s) = (0 \quad A_{i_S}^T) \begin{pmatrix} \tilde{X}_R \\ X_S \end{pmatrix} = A_{i_S}^T\,X_S. \tag{4.10}$$

This reduction can be expressed as a congruence transformation with congruence transform

$$W = \begin{pmatrix} \tilde{V} & 0 \\ 0 & I \end{pmatrix}.$$

One can see, that the reduced model has preserved the input-output structure of the original system, since A_{i_s} has not changed. Only some zeros in the input-output matrix are discarded by the reduction. $\tilde{G}$ and $\tilde{C}$ are symmetric. So this model is realizable by unstamping.

Additionally the sparsity of the original system is preserved in the reduced model. $\tilde{G}_R$ and $\tilde{C}_R$ are diagonal, $\tilde{G}_S$ and $\tilde{C}_S$ are not changed at all and thus remain sparse and $\tilde{G}_K$ and $\tilde{C}_K$ preserve the zero columns of G_K and C_K.

Since G_S and C_S are not changed by the reduction the elements that connect selected nodes to other selected nodes or the ground node are preserved. Thus, if the right selection strategy was chosen, the DC offset is preserved.

4.4 Realization of reduced model

In this section we want to take a closer look on the reduced system matrices. From the reduced system matrices the reduced circuit can be directly synthesized by unstamping. The reduced circuit will give insights about selection strategies. So, by considering the reduced system matrices one can understand, which selection strategies generally preserve the DC offset and which do not.

First of all the circuit elements, that connect the selected nodes with each other or to ground are not changed by the reduction, since the corresponding blocks G_S and C_S remain unchanged by the reduction.

The circuit elements that connected the remaining nodes to each other or ground are gone and replaced by other elements with a different circuit structure. $\tilde{G}_R$ and $\tilde{C}_R$ are diagonal matrices, so the reduced remaining nodes are not connected to each other. Generally, these nodes are connected by one resistor and one capacitor to ground.

However, the reduced remaining nodes have a connection to the selected nodes by the reduced coupling matrices $\tilde{G}_K$ and $\tilde{C}_K$. To understand which nodes are connected by which kinds of elements, the meaning of the entries in G_K and C_K has to be explained. Then the effect of their reduction can be described, which leads to the entries in the reduced coupling matrices.

Assume a coupling capacitor C_c. By considering the partitioning (4.1), let one end of C_c be connected to the selected node $n_R + k$, that the k^{th} column (and the k^{th} row) in C_S corresponds to. Then let the other end of C_c be connected to the remaining node j, that the j^{th} row (and the j^{th} column) in C_R corresponds to. This coupling of C_c would lead to an entry $-C_c$ in the k^{th} column and the j^{th} row in the matrix C_K. So, if a selected node $n_R + k$ would not be coupled to any of the remaining nodes by a capacitor, the k^{th} column in C_K would be a zero column. This holds in an analogous way for resistors in G_K. Thus there can be a lot of zero columns in G_K and C_K. As example, the partitioned matrices G and C for the circuit in Figure 4.1 are given in the following. The selection strategy 2 of Section 4.1 is assumed, so the selected nodes are 9, 8, 7 and 6 and $n_S = 4$. Thus the remaining nodes are 1, 2, 3, 4 and 5 and $n_R = 5$.

First the matrix G:

$$G = \left(\begin{array}{ccccc|cccc}
G_3 & 0 & -G_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & G_4 & -G_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
-G_3 & -G_4 & G_3+G_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & G_5 & -G_5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -G_5 & G_5 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & G_6 & -G_6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -G_6 & G_2+G_6+G_7 & -G_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -G_2 & G_1+G_2 & -G_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -G_1 & G_1
\end{array}\right)$$

It can be seen that in this example $G_K = 0$. G_K is always zero for selection strategy 2, since coupling is done by capacitors only. Here, the selected nodes are connected to the remaining nodes via capacitors C_1 and C_5. Now the matrix C is given:

$$C = \left(\begin{array}{ccccc|cccc}
C_1+C_2 & -C_2 & 0 & 0 & 0 & 0 & 0 & -C_1 & 0 \\
-C_2 & C_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & C_3+C_4 & -C_4 & -C_3 & 0 & 0 & 0 & 0 \\
0 & 0 & -C_4 & C_4+C_5 & 0 & -C_5 & 0 & 0 & 0 \\
0 & 0 & -C_3 & 0 & C_3 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & -C_5 & 0 & C_5+C_6 & -C_6 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -C_6 & C_6+C_7 & 0 & 0 \\
-C_1 & 0 & 0 & 0 & 0 & 0 & 0 & C_1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)$$

From C_K it can be seen the selected node 6 $(= n_R+1)$ is connected to remaining node 4 via C_5. That is why the entry $-C_5$ is in the 1^{st} column and 4^{th} row in C_K. Now, because selected nodes 7 $(= n_R + 2)$ and 9 $(= n_R + 4)$ are not at all connected to remaining nodes, the 2^{nd} and 4^{th} columns in C_K consist of zero entries only. These zero columns are preserved in $\tilde{C}_K$. However, the columns in C_K that have at least one non-zero entry, result in a full column in $\tilde{C}_K$, since C_K is left multiplied by a (in general) full matrix. The same holds for G_K.

Now, the connections in the reduced circuit can be specified. Assume in the original circuit was one connection by a resistor or a capacitor of selected node n to remaining node m. Then in the reduced circuit all reduced remaining nodes will be connected to selected node n by resistors or capacitors, respectively.

Therefore the reduced circuit for the circuit in Figure 4.1 would look like in Figure 4.2. It can be seen, that selected nodes $(n_d + 1)$ and $(n_d + 3)$ are coupled to every reduced

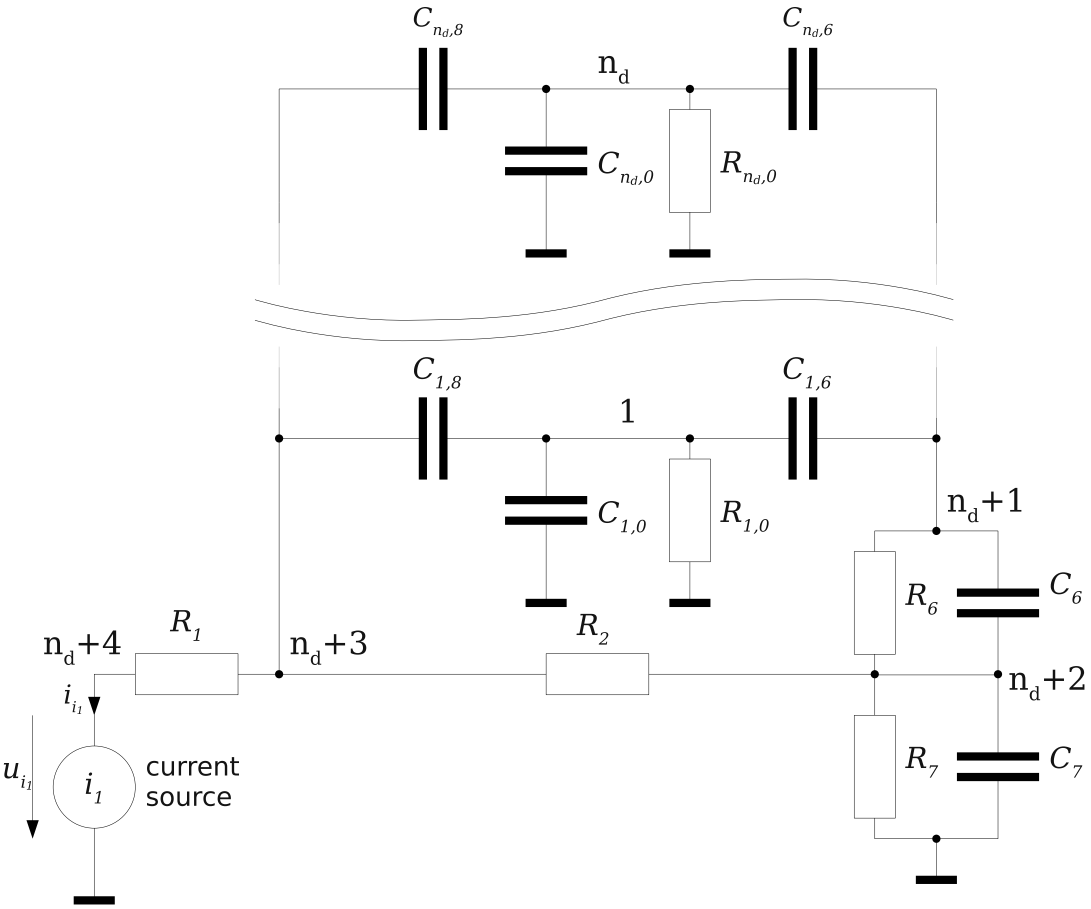


Figure 4.2: Realized reduced model yielded by reduction of the circuit in Figure 4.1 with SparseMA. n_d dominant poles are used and the selected nodes in the original circuit were 9, 8, 7 and 6.

remaining node. This is because in the original circuit each of the two selected nodes (formerly numbered 6 and 8, respectively) were coupled by a capacitor to one of the remaining nodes. Here, the DC offset is preserved, since in the reduced circuit the DC path to ground is the same as before.

However, if we had used selection strategy 1 from Section 4.1 node 6 would have been a remaining node. Thus the coupling from selected node 7 to remaining node 6 would have been done by capacitor C_6 and resistor R_6. Therefore the reduced circuit would have couplings by resistors, too, from selected node 7 (as numbered in the original circuit) to the reduced remaining nodes. As the reduced nodes consist of a capacitor and a resistor connected to ground, there would be a different DC path to ground in the reduced circuit. The resistors connected in the reduced remaining nodes to ground would be reachable by the resistive coupling from the terminal. Therefore this selection strategy fails to preserve the DC offset.

Hence, it is not sufficient to preserve the nodes on the DC path to ground only from the original circuit. There must be preserved at least one series capacitor (without a resistor in parallel) or the coupling can be only allowed by capacitors, if the DC offset should be preserved. So alternative successful strategies could select additional nodes to be preserved. E.g. in the circuit in Figure 4.1 node 1 could be selected to be preserved in addition to nodes 9, 8, 7 and 6. Because the connection from node 8 to node 1 is done by a series capacitor (C_1) it does not matter for the DC offset how the circuit behind this circuit looks like.

Remark: It appears, that J. Rommes and R. Ionutiu use in [12] a similar strategy to strategy 2 from Section 4.1 in their example[13]. However, the coupling caused by their strategy is not purely capacitive. This can be seen from the example in [12, Section 4.2], because $\hat{G}_K$ has got one non-zero column. That means neither G_K nor C_K was zero. Later, in Section 4.5.1 will be explained that $G_K \neq 0$ and $C_K \neq 0$ induces that the frequency dependent SAMDP was used to reduce that example.

4.5 Practical aspects

We would like to highlight some practical aspects in this section that are important for implementation of SparseMA. In Section 4.5.1 is explained how the dominant poles can be found. In practice this is not done by the computation of all eigenvalues and eigenvectors. Section 4.5.2 shows a circuit where SparseMA is not applicable. This leads to a discussion of the application of MOR methods in general.

4.5.1 Numerical eigensolver

Firstly, an example in [12] is stated in the following table to give an impression of the dimensions in practical applications. The order of the original circuit in that example is 3253 with $n_S = 62$ selected nodes and thus $n_R = 3191$ remaining nodes.

	original	reduced
Order	$3191 + 62$	$7 + 62$
♯ Resistors	7944	78
♯ Capacitors	4366	383

For matrices of this size (3191) a complete solution of the generalized eigenvalue problem can be too expensive. So other ways to find the (dominant) poles of the part of the circuit consisting of remaining nodes have to be used. In [12] the Subspace Accelerated MIMO

Dominant Pole Algorithm (SAMDP) [17, 19] or a variant, called frequency dependent SAMDP is used. We will now consider for which cases what kind has to be used.

If selection strategy 2 of Section 4.1 is used, the coupling is purely capacitive. This results in a zero G_K, as described in Section 4.4. Then, we can simplify the expression for $Y_R(s)$ from (4.4) or (4.5):

$$\begin{aligned} Y_R(s) &= (G_K + s\,C_K)^T\,(G_R + s\,C_R)^{-1}\,(G_K + s\,C_K) \\ &= s^2\;C_K^T\,(G_R + s\,C_R)^{-1}\,C_K \\ &= s^2 \sum_{k=1}^{n_e} \frac{C_K^T\,v_k\,v_k^T\,C_K}{s - \lambda_k} \\ &= s^2 \sum_{k=1}^{n_e} \frac{\check{R}_k}{s - \lambda_k} \end{aligned}$$

where the frequency independent residues $\check{R}_k = \check{q}_k\,\check{q}_k^T$ and $\check{q}_k = C_K^T\,v_k$ with v_k as the eigenvector corresponding to the eigenvalue λ_k.

With these residues a frequency independent dominance value can be defined as

$$\hat{\check{R}}_k := \frac{\|\check{R}_k\|_2}{|\lambda_k|}. \tag{4.11}$$

The dominant poles determined by this definition of dominance values are the same as for the definition of (4.6). The different definitions do not influence the selection of the dominant poles, because the frequency in (4.6) just scales the dominance values (all in the same way). However, this only holds for $G_K = 0$. In case of $C_K = 0$ the redefinition would not be even necessary, since (4.6) would not depend on the frequency anymore. Anyway, in case that $G_K = 0$ (or $C_K = 0$) there is no need for a frequency dependent version of the SAMDP.

Although the conventional SAMDP can be used, it is not obvious how. As stated in the end of Section 3.4, the SAMDP needs to calculate the inverse of the transfer function. Since in SparseMA not the poles for the transfer function H have to be found, but the ones of Y_R, we will consider Y_R in the following. This will not be limited to $G_K = 0$ or $C_K = 0$.

The zero columns in G_K and C_K cause zero columns in $G_K + s\,C_K$. These lead to zero columns and rows in $Y_R(s)$. More exactly, if the k^{th} column in $G_K + s\,C_K$ is a zero column, the k^{th} column in $Y_R(s)$ as well as the k^{th} row (by symmetry) will consist only of zeros. This can be seen by taking a closer look on the nominator of the sum in $Y_R(s)$

$$Y_R(s) = \sum_{k=1}^{n_e} \frac{R_k}{s - \lambda_k}, \quad \text{where } R_k = (G_K + s\,C_K)^T\,(v_k\,v_k^T)\,(G_K + s\,C_K)$$

So a regularized version of Y_R has to be generated. Here, regularization means that non-zero rows and columns will be discarded, such that the yielded matrix is regular. Let n_{nz} denote the number of non-zero columns in $G_K + s\,C_K$. Assuming there are at least n_{nz} distinct eigenvalues, $Y_R(s)$ can be regularized. Let regularized version be denoted by $\hat{Y}_R(s)$. By defining a matrix $\hat{A}(s) \in \mathbb{C}^{n_R \times n_{nz}}$ which consists of the non-zero columns of $G_K + s\,C_K$ one can define

$$\hat{Y}_R(s) := \hat{A}(s)^T \left(G_R + s\,C_R\right)^{-1} \hat{A}(s) = \sum_{k=1}^{n_e} \frac{\hat{A}(s)^T\, v_k\, v_k^T\, \hat{A}(s)}{s - \lambda_k}. \tag{4.12}$$

Note, that the poles and modes of Y_R are the same as of $\hat{Y}_R$. Hence, one can use $\hat{Y}_R$ to find the poles of Y_R.

In case of constant dominance values ($C_K = 0$ or $G_K = 0$) the poles can be found in this way with SAMDP. For $C_K = 0$ the described $\hat{A}(s)$ is constant. For $G_K = 0$ a constant $\hat{A}$ can be defined, by taking the non-zero columns of C_K (instead of $s\,C_K$). For this purpose, $\hat{A}(s)$ can be taken as input-output matrix.

However, for the case $G_K \neq 0$ and $C_K \neq 0$ a frequency dependent SAMDP [12] version has to be used. This goes beyond the scope of this work. Basically the Newton scheme has to be derived for $\hat{Y}_R$ and the algorithm needs to be modified accordingly.

4.5.2 A negative example

There can still be found applications where both of the selection strategies from Section 4.1 fail. An obvious, practicable example can be a simple model, extracted by a two-terminal (or two port) RC interconnect line, as shown in Figure 4.3. In this model basically all nodes are connected by resistors from the terminals, so selection strategy 2 from Section 4.1 would consider all nodes as selected nodes and thus no reduction could

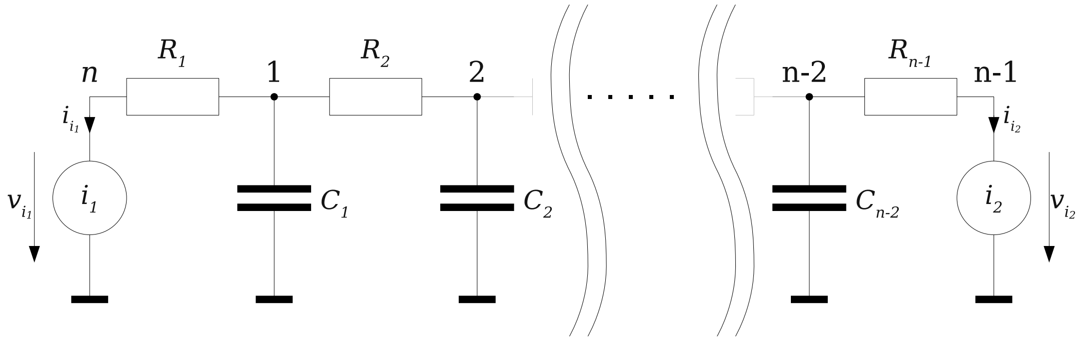


Figure 4.3: Scalable model of a two-port RC interconnect line with n nodes, $n-1$ resistors and $n-2$ capacitors.

be applied; selection strategy 1 would not select just a single node (except the terminal nodes), since there is no DC path to ground.

So, there exist electrical circuits identifiable by topology that cannot be reduced with SparseMA with the proposed selection strategies. Probably this is a circuit which is generally not suitable for modal approximation since when plotting the logarithm of the sorted dominance values (3.14) (see Figure 4.4), there is no rapid fall off observable. So it cannot be decided in a natural, clear way, which of the poles are dominant. The very specialized model order reduction method described in [3] is best suited, but maybe Krylov space methods like SPRIM/IOPOR [22] (see Section 3.3) could yield good results here, too. However, this is left for further research.

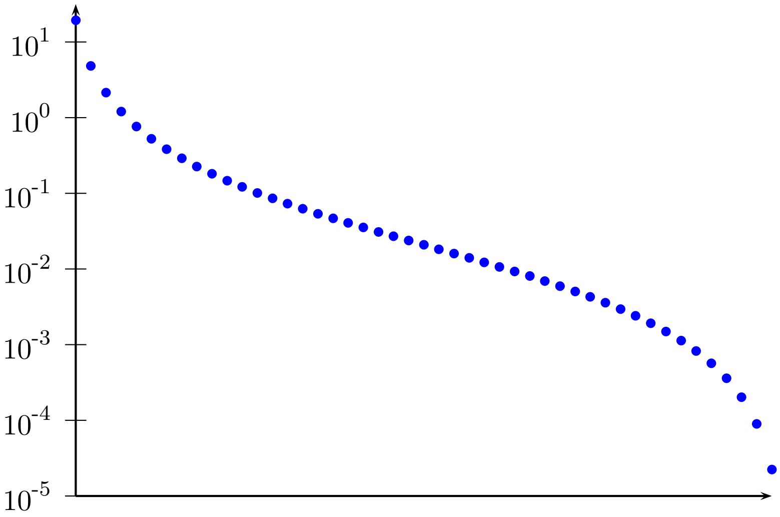


Figure 4.4: All dominance values of non-zero poles $\hat{R}_1, \ldots, \hat{R}_{47}$ (logarithmically scaled) plotted in descending order. The circuit has got 50 numbered nodes.

From the last example it should be clear, that there are RC circuits not suitable for reduction with MA and so with SparseMA. It seems to be likely, that there is no single MOR method that is the best suited method for all applications or circuits. This is an important insight for future research.

5 Conclusion and further work

An overview of modelling of circuits and model order reduction with focus on the method SparseMA [12] was presented in this work. Some questions, e.g. how to apply the SAMDP [17] to find the dominant poles, could be answered. However, there are a number of open questions that motivate further research.

Selection strategies: The question about the right selection strategy (for the selected nodes) arises in Section 4.1. From insights gained in Section 4.4 possible strategies that preserve the DC offset are described in the end of that section. However, for different applications one could think of different selection strategies. For example there could be found selection strategies for high frequency applications, where the nodes are not selected by the graph of G but of C. This could be especially useful for circuits, where the selection strategy 2 from Section 4.1 cannot be applied. There can be circuits e.g. in which only capacitors are connected to the terminals. These circuits might operate in a certain frequency interval, in which a small error would be desirable. Another possibility for selection strategies could be that a *weighted graph* would be used, maybe in addition to some necessary purely topological aspects. Previous ideas for selection strategies considered only topological aspects (graphs), without regarding the element values. A mixture might yield other plausible selection strategies.

Different dominance criterion: In Section 4.3 a frequency dependent dominance criterion is defined for SparseMA. Here are different criteria possible and the theoretical background could improve the decision which criterion is a good choice. This leads to the question: What is the best dominance criterion? Maybe the denominator should also depend on S_0, since that reflects the contribution of the summands more exactly.

Synthesis method: The structure of the coupling matrices G_K and C_K was described in Section 4.4. The structure of $G_K + s\,C_K$ can be compared to (conventional) input-output matrices like A_i. Let now be the situation differently. G_R and C_R as reduced system matrices and (maybe only non-zero columns of) $G_K + s\,C_K$ as reduced input-output matrix are given. Now let the task be to synthesize this circuit. Here, in Chapter 4, a surrounding circuit (described by $G_S + s\,C_S$) with independent current sources and their placement (described by A_i) is given. However, assume G_S, C_S and A_i are not given and the question is now posed from the other end. Can a surrounding circuit with independent current sources be found to avoid the use of controlled sources? The answer could yield a new synthesis method (I am not aware of the existence of such a method).

General applicability of SparseMA: Although there is optimization potential with selection strategies, not every circuit is reducible with a reasonable result (small order with good accuracy) by SparseMA. The dominance value plot in Figure 4.4 did indicate that modal approximation in general is probably not the preferred method for the example circuit in Figure 4.3. So, independent of the selection strategy, there are circuits that would just require too much poles to accurately match the behaviour of the original circuit. This has to be generalized in further work to define a criterion that *predicts* (maybe from topological conditions) whether MA is suitable for reduction or not. With the term predict is meant, that it should be decidable without doing the majority of the computational effort of the reduction itself (finding the dominant poles). Such a criterion could identify unsuited circuits for reduction with MA. If a circuit is unsuited for MA, it is probably also unsuited for SparseMA. Similar criteria would be possible and desirable for other reduction methods.

However, even if a circuit is well reducible with (i.e. suited for) MA, it might still be unsuited for reduction with SparseMA. There can be circuits that are not reducible with SparseMA due to the selection strategy. E.g. if in a circuit all nodes are connected to terminals by resistors, the proposed strategy 2 from Section 4.1 will clearly fail. Even if only most of nodes are connected to terminals by resistors, a reduction does probably not make sense when looking at the speed-up of the simulation. So by comparing the number of selected nodes to the number of remaining nodes one might be able to estimate if a reduction with SparseMA can be worthwile. From this thought a criterion could be employed again to identify circuits unsuitable for reduction with SparseMA.

If all criterions are known to identify circuits unsuited for reduction with SparseMA it can be decided whether SparseMA is the method of choice for a circuit.

Bibliography

[1] D. L. Boley. Krylov Space Methods on State-Space Control Models. *Circuits, Systems, and Signal Processing*, 13:733–758, 1994.

[2] G. B. Dantzig and M. N. Thapa. *Linear Programming: Introduction.* Springer, Berlin, Mar 1997.

[3] A. Devgan and P. R. O'Brien. Realizable reduction for RC interconnect circuits. In *ICCAD '99: Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, pages 204–207, Piscataway, NJ, USA, 1999. IEEE Press.

[4] R. W. Freund. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J. Comput. Appl. Math.*, 123(1-2):395–421, 2000.

[5] R. W. Freund. SPRIM: structure-preserving reduced-order interconnect macromodeling. In *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 80–87. IEEE Computer Society, 2004.

[6] R. W. Freund. RCL circuit equations. In P. Benner, V. Mehrmann, and D. C. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 367–371. Springer Berlin, 2005.

[7] R. W. Freund. The SPRIM Algorithm for Structure-Preserving Order Reduction of General RCL Circuits. Technical report, Department of Mathematics, UC Davis, Sep 2009.

[8] M. Günther, U. Feldmann, and J. ter Maten. *Modelling and Discretization of Circuit Problems*, chapter 1, page 532. Elsevier B.V., 2005.

[9] M. Günther, M. Hoschek, and P. Rentrop. Differential-Algebraic Equations in Electric Circuit Simulation. *Int. J. Electron. Commun. (AEÜ)*, 57:101–107, 2000.

[10] R. Ionutiu and J. Rommes. A framework for synthesis of reduced order models. Technical report, Department of Mathematics and Computer Science, TU Eindhoven, Sep 2009.

[11] R. Ionutiu and J. Rommes. Circuit synthesis of reduced order models. Technical Report 2008/00316, NXP Semiconductors, 2009.

[12] R. Ionutiu and J. Rommes. Model order reduction for multi-terminal circuits. Technical report, Department of Mathematics and Computer Science, TU Eindhoven, Sep 2009.

[13] R. Ionutiu and J. Rommes. personal communication, Feb 2010.

[14] C. Jelenowski. Differential-Algebraic Equations and Mathematical Modelling of Electrical Circuits for Radio Frequency Applications. Diploma thesis, University of Applied Sciences Bochum, Feb 2010.

[15] K. J. Kerns and A. T. Yang. Stable and efficient reduction of large, multiport rc networks by pole analysis via congruence transformations. *Design Automation Conference*, 0:280–285, 1996.

[16] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. In *ICCAD '97: Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, pages 58–65, Washington, DC, USA, 1997. IEEE Computer Society.

[17] J. Rommes. *Methods for eigenvalue problems with applications in model order reduction.* PhD thesis, Utrecht University, 2007.

[18] J. Rommes. *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*, chapter Modal Approximation and Computation of Dominant Poles. Springer, 2008. Available at http://sites.google.com/site/rommes/publications.

[19] J. Rommes and N. Martins. Efficient computation of multivariable transfer function dominant poles using subspace acceleration. *IEEE Trans. on Power Systems*, 21(4):1471–1483, Nov 2006.

[20] A. E. Ruehli. Equivalent Circuit Models for Three-Dimensional Multiconductor Systems. *IEEE Transactions on Microwave Theory and Techniques*, 22(3):216–221, 1974.

[21] Y. Su, J. Wang, X. Zeng, Z. Bai, C. Chiang, and D. Zhou. Sapor: second-order arnoldi method for passive order reduction of rcs circuits. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 74–79, Washington, DC, USA, 2004. IEEE Computer Society.

[22] F. Yang, X. 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(2):376–396, Feb 2008.

[23] H. Yu, Y. Shi, L. He, and D. Smart. A fast block structure preserving model order reduction for inverse inductance circuits. In *ICCAD '06: Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, pages 7–12, New York, NY, USA, 2006. ACM.

Index