

Model order reduction for multi-terminal circuits

Citation for published version (APA):

Ionutiu, R., & Rommes, J. (2009). *Model order reduction for multi-terminal circuits*. (CASA-report; Vol. 0929). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/2009

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

CASA-Report 09-29
September 2009

Model order reduction for
multi-terminal circuits

by

R. Ionutiu, J. Rommes



Centre for Analysis, Scientific computing and Applications
Department of Mathematics and Computer Science
Eindhoven University of Technology
P.O. Box 513
5600 MB Eindhoven, The Netherlands
ISSN: 0926-4507

Model order reduction for multi-terminal circuits

Roxana Ionutiu¹ and Joost Rommes²

Abstract Analysis of effects due to parasitics is of vital importance during the design of large-scale integrated circuits, since it gives insight into how circuit performance is affected by undesired parasitic effects. Due to the increasing amount of interconnect and metal layers, parasitic extraction and simulation may become very time consuming or even unfeasible. Developments are presented, for reducing systems describing R and RC netlists resulting from parasitic extraction. The methods exploit tools from graph theory to improve sparsity preservation especially for circuits with multi-terminals. Circuit synthesis is applied after model reduction, and the resulting reduced netlists are tested with industrial circuit simulators. With the novel RC reduction method SparseMA, experiments show reduction of 95% in the number of elements and 68x speed-up in simulation time.

1 Introduction

Analysis of effects due to parasitics is of vital importance during the design of large-scale integrated circuits and derived products. One way to model parasitics is by means of parasitic extraction, which results in large linear $RCL(k)$ networks. In ESD analysis, for instance, the interconnect network is modeled by resistors with resistances that are based on the metal properties. In other (RF) applications one needs RC or even $RCLk$ extractions to deal accurately with higher frequencies as well.

Roxana Ionutiu

Eindhoven University of Technology, Department of Mathematics and Computer Science, P.O. Box 513 5600 MB Eindhoven jointly with the School of Engineering and Science, Jacobs University Bremen, 28725 Bremen, Germany, e-mail: r.ionutiu@tue.nl, r.ionutiu@jacobs-university.de

Joost Rommes

NXP Semiconductors, HTC46-2.206, PostBox WDA-2, NL-5656 AE Eindhoven, e-mail: joost.rommes@nxp.com

The resulting parasitic networks may contain up to millions of resistors, capacitors, and inductors, and hundreds of thousands of internal nodes, and thousands of external nodes (nodes with connections to active elements such as transistors). Simulation of such large networks within reasonable time is often not possible, and including such networks in full system simulations may be even unfeasible. Hence, there is need for much smaller networks that accurately or even exactly describe the behavior of the original network, but allow for fast analysis.

In this chapter we describe recently developed methods for the reduction of large R networks, and present a new approach for the reduction of large RC networks. We show how insights from graph theory, numerical linear algebra, and matrix reordering algorithms can be used to construct a reduced network with the same number of external nodes, but much fewer internal nodes and circuit elements (resistors and capacitors). The approach is illustrated by numerical results.

The chapter is organized as follows. Sect. 2 revisits recent work on reduction of R networks [8, 9]. It provides the basis for understanding how graph theoretical tools can be used to significantly improve the sparsity of the reduced models, which are later synthesized [29] into reduced netlists. Sect. 3 deals with the reduction of RC networks. Sect. 3.1 first reviews an existing method which employs *pole analysis* and *congruence transformations (PACT)* [1] to reduce RC netlists with multi-terminals. In Sect. 3.2 the new method *Sparse Modal Approximation (SparseMA)* is presented, where graph-theoretical tools are brought in to enhance sparsity preservation for the reduced models. The numerical results for both R and RC netlist reduction are presented in Sect. 4. Sect. 5 concludes.

2 Reduction of R networks

In this section we review the approach for reducing R networks, as developed in [8, 9]. Reduction of R networks, i.e., networks that consist of resistors only, is needed in electro-static discharge analysis (ESD) analysis, where large extracted R networks are used to model the interconnect. Accurate modeling of interconnect is required here, since the costs involved may vary from a few cents to millions if, due to interconnect failures, a respin of the chip is needed. An example of a damaged piece of interconnect that was too small to conduct the amount of current is shown in Figure 1.

2.1 Circuit equations and matrices

Kirchhoff's Current Law and Ohm's Law for resistors lead to the following system of equations for a resistor network with N resistors (resistor i having resistance r_i) and n nodes ($n < N$):

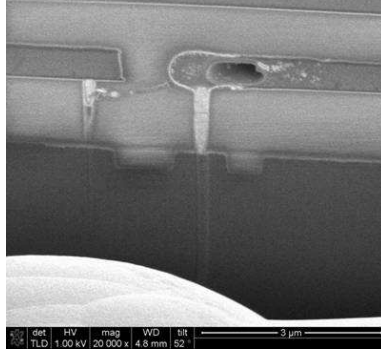


Fig. 1 Example of a piece of interconnect that was damaged because it was too small to conduct the amount of current caused by a peak charge.

$$\begin{bmatrix} R & P \\ -P^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{i}_b \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{i}_n \end{bmatrix}, \quad (1)$$

where $R = \text{diag}(r_1, \dots, r_N) \in \mathbb{R}^{N \times N}$ is the resistor matrix, $P \in \{-1, 0, 1\}^{N \times n}$ is the incidence matrix, $\mathbf{i}_b \in \mathbb{R}^N$ are the resistor currents, $\mathbf{i}_n \in \mathbb{R}^n$ are the injected node currents, and $\mathbf{v} \in \mathbb{R}^n$ are the node voltages.

The MNA (modified nodal analysis) formulation [11] can be derived from (1) by eliminating the resistor currents $\mathbf{i}_b = -R^{-1}P\mathbf{v}$:

$$G\mathbf{v} = \mathbf{i}_n, \quad (2)$$

where $G = P^T R^{-1} P \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite. Since currents can only be injected in external nodes, and not in internal nodes of the network, system (2) has the following structure:

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_e \\ \mathbf{v}_i \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{i}_e, \quad (3)$$

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

A current source (with index s) between terminals a and b with current j results in contributions $B_{a,s} = 1$, $B_{b,s} = -1$, and $\mathbf{i}_e(s) = j$. If current is only injected in a terminal a (for instance if a connects the network to the top-level circuit), the contributions are $B_{a,s} = 1$ and $\mathbf{i}_e(s) = j$.

Finally, systems (1)–(3) must be made consistent by grounding a node *gnd*, i.e., setting $\mathbf{v}(\text{gnd}) = 0$ and removing the corresponding equations. In the following we will still use the notation G for the grounded system matrix, if this does not lead to confusion.

2.2 Problem formulation

The problem is: given a very large resistor network described by (1), find an equivalent network with (a) the same external nodes, (b) exactly the same path resistances between external nodes, (c) $\hat{n} \ll n$ internal nodes, and (d) $\hat{r} \ll r$ resistors. Additionally, (e) the reduced network must be realizable as a netlist so that it can be (re)used in the design flow as subcircuit of large systems.

Simply eliminating all internal nodes will lead to an equivalent network that satisfies conditions (a)–(c), but violates (d) and (e): for large numbers m of external nodes, the number of resistors $\hat{r} = (m^2 - m)/2$ in the dense reduced network is in general much larger than the number of resistors in the sparse original network (r of $O(n)$), leading to increased memory and CPU requirements.

2.3 Existing approaches

There are several approaches to deal with large resistor networks. In some cases the need for an equivalent reduced network can be circumvented in some way: due to sparsity of the original network, memory usage and computational complexity are *in principle* not an issue, since solving linear systems with the related conductance matrices is typically of complexity $O(n^\alpha)$, where $1 < \alpha \leq 2$, instead of the traditional $O(n^3)$ [17]. Of course, α depends on the sparsity and will rapidly increase as sparsity decreases. This also explains why eliminating all internal nodes does not work in practice: the large reduction in unknowns is easily undone by the enormous increase in number of resistors, mutually connecting all external nodes.

However, if we want to (re)use the network in full system simulations, a reduced equivalent network is needed to limit simulation times or make simulation possible at all. In [20] approaches based on large-scale graph partitioning packages such as (h)METIS [21] are described, but only applied to small networks. Structure preserving projection methods for model reduction [22, 23], finally, have the disadvantage that they lead to dense reduced-order models if the number of terminals is large. There is commercial software [18, 19] available for the reduction of parasitic reduction networks.

2.4 Improved approach

Knowing that eliminating all internal nodes is not an option and that projection methods lead to dense reduced-order models, we use concepts from matrix reordering algorithms such as AMD [24] and BBBB [25], usually used as preprocessing step for (parallel) LU- or Cholesky-factorization, to determine which nodes to eliminate. The fill-in reducing properties of these methods also guarantee sparsity of the reduced network. Similar ideas have also been used in [20, 26].

Our main motivation for this approach is that large resistor networks in ESD typically are extracted networks with a structure that is related to the underlying (interconnect) layout. Unfortunately, the extracted networks are usually produced by extraction software of which the algorithms are unknown, and hence the structure of the extracted network is difficult to recover. Standard tools from graph theory, however, can be used to recover at least part of the structure.

Our approach can be summarized as follows:

1. The first step is to compute the strongly connected components [12] of the network. The presence of strongly connected components is very natural in extracted networks: a piece of interconnect connecting two other elements such as diodes or transistors, for instance, results in an extracted network with two terminals, disconnected from the rest of the extracted circuit. By splitting the network in to connected components, we have simplified the problem of reduction because we can deal with the connected components one by one.
2. The second step is to selectively eliminate internal nodes in the individual connected components. For resistor networks, this can be done using the Schur complement [28], and no approximation error is made. The key here is that those internal nodes are eliminated that give the least fill-in. First, (Constrained) AMD [13] is used to reorder the unknowns such that the terminal nodes will be among the last to eliminate. To find the optimal reduction, internal nodes are eliminated one-by-one in the order computed by AMD, while keeping track of the reduced system with fewest resistors.
Since the ordering is chosen to minimize fill-in, the resulting reduced matrix is sparse. Note that all operations are exact, i.e., we do not make any approximations. As a result, the path resistances between external nodes remain equal to the path resistances in the original network.
3. Finally, the reduced conductance matrix can be realized as a reduced resistor network that is equivalent to the original network. This is done easily by unstampig the values in the G matrix into the corresponding resistor values and their node connections in the netlist [5]. Since the number of resistors (and number of nodes) is smaller than in the original network, also the resulting netlist is smaller in size.

An additional reduction could be obtained by removing relatively large resistors from the resulting reduced network. However, this will introduce an approximation error that might be hard to control a priori, since no sharp upper bounds on the error are available [7]. Another issue that is subject to further research is that the optimal ratio of number of (internal) nodes to resistors (sparsity) may also depend on the ratio of number of external to internal nodes, and on the type of simulation that will be done with the network.

In the following sections we will describe how strongly connected components and fill-in minimizing reorderings can be used for the reduction of RC networks as well.

3 Reduction of RC networks

This section presents the developments for RC netlist reduction, first by reviewing an existing approach called PACT (Pole Analysis via Congruence Transformations). Then, graph-based tools are brought in to enhance sparsity preservation with the novel reduction method, SparseMA (Sparse Modal Approximation).

Following the problem description in [1], consider the modified nodal analysis (MNA) description of an input *impedance* type RC circuit, driven by input currents:

$$(\mathbf{G} + s\mathbf{C})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s), \quad (4)$$

where \mathbf{x} denote the node voltages, and \mathbf{u} represent the currents injected into the terminals (also called ports or external nodes). The number of internal nodes is n , and the number of terminals is p , thus $\mathbf{G} \in \mathbb{R}^{(p+n) \times (p+n)}$, $\mathbf{C} \in \mathbb{R}^{(p+n) \times (p+n)}$ and $\mathbf{B} \in \mathbb{R}^{(p+n) \times p}$. A natural choice for the system outputs are the voltage drops at the terminal nodes, i. e., $\mathbf{y}(s) = \mathbf{B}^T \mathbf{x}(s)$. Thus the transfer function of (4) is the input impedance:

$$\mathbf{Z}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{B}^T (\mathbf{G} + s\mathbf{C})^{-1} \mathbf{B}. \quad (5)$$

Modal approximation is a method to reduce (4), by preserving its most dominant eigenmodes. The dominant eigenmodes are a subset of the poles of $\mathbf{Z}(s)$ [i. e. of the generalized eigenvalues $\Lambda(-\mathbf{G}, \mathbf{C})$] and can be computed using specialized eigenvalue solvers (SADPA [4] or SAMDP [2, 10]). For the complete discussion on modal approximation and its implementation we refer to [3, 4, 10]. Here, we emphasize that applying modal approximation to reduce (4) directly is unsuitable especially if the underlying RC circuit has many terminals (inputs). This is because modal approximation does not preserve the structure of \mathbf{B} and \mathbf{B}^T during reduction (for ease of understanding we denote the input-output structure loss as *non-preservation of terminals*) [5]. Modeling the input-output connectivity of the reduced model would require synthesis via controlled sources at the circuit terminals, and furthermore would connect all terminals with one-another [5]. In this chapter we present several alternatives for reducing RC netlists where not only the terminals are preserved, but also the sparsity of the reduced models.

Grouping the node voltages so that $\mathbf{x}_P \in \mathbb{R}^p$ are the voltages measured at the terminal nodes, and $\mathbf{x}_I \in \mathbb{R}^n$ are the voltages at the internal nodes, we can partition (4) as follows:

$$\left(\begin{bmatrix} \mathbf{G}_P & \mathbf{G}_C^T \\ \mathbf{G}_C & \mathbf{G}_I \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_P & \mathbf{C}_C^T \\ \mathbf{C}_C & \mathbf{C}_I \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_P \\ \mathbf{x}_I \end{bmatrix} = \begin{bmatrix} \mathbf{B}_P \\ \mathbf{0} \end{bmatrix} \mathbf{u}. \quad (6)$$

Since no current is injected into internal nodes, the non-zero contribution from the input is $\mathbf{B}_P \in \mathbb{R}^{(p \times p)}$. Eliminating \mathbf{x}_I , system (6) is equivalent to:

$$\underbrace{[(\mathbf{G}_P + s\mathbf{C}_P)]}_{\mathbf{Y}_P(s)} - \underbrace{(\mathbf{G}_C + s\mathbf{C}_C)^T (\mathbf{G}_I + s\mathbf{C}_I)^{-1} (\mathbf{G}_C + s\mathbf{C}_C)}_{\mathbf{Y}_I(s)} \mathbf{x}_P = \mathbf{B}_P \mathbf{u} \quad (7)$$

$$\mathbf{Y}(s) = \mathbf{Y}_P(s) - \mathbf{Y}_I(s) \quad (8)$$

In (7) the matrix blocks $(\mathbf{G}_P + s\mathbf{C}_P)$ corresponding to the circuit terminals are isolated. Applying modal approximation on $\mathbf{Y}_I(s)$ would reduce the system and preserve the location of the terminals. This would involve for instance computing the dominant eigenmodes of $(-\mathbf{G}_I, \mathbf{C}_I)$ via a variant of SAMDP [called here *frequency dependent SAMDP*, because the input-output matrices $(\mathbf{G}_C + s\mathbf{C}_C)$ depend on the frequency s]. We have implemented this approach, but it turns out that a large number of dominant eigenmodes of $(-\mathbf{G}_I, \mathbf{C}_I)$ would be needed to capture the DC and offset of the full system $\mathbf{Y}(s)$. Instead, two alternatives are presented that improve the quality of the approximation: an existing method called *PACT (Pole Analysis via Congruence Transformations)* [1] and a novel graph-based reduction called *SparseMA (Sparse Modal Approximation)*.

3.1 Existing method: PACT

In [1] the authors propose to capture the DC and offset of $\mathbf{Y}(s)$ via a congruence transformation which reveals the first two moments of $\mathbf{Y}(s)$ as follows. Since \mathbf{G}_I is symmetric positive definite, the Cholesky factorization $\mathbf{L}\mathbf{L}^T = \mathbf{G}_I$ exists. Using the following congruence transformation:

$$\mathbf{X} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{G}_I^{-1}\mathbf{G}_C & \mathbf{L}^{-T} \end{bmatrix}, \quad \mathbf{G}' = \mathbf{X}^T \mathbf{G} \mathbf{X} = \begin{bmatrix} \mathbf{G}'_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{C}' = \mathbf{X}^T \mathbf{C} \mathbf{X} = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}'_C \\ \mathbf{C}'_C & \mathbf{C}'_I \end{bmatrix} \quad (9)$$

equations (7), (8) are rewritten as:

$$\underbrace{[(\mathbf{G}'_P + s\mathbf{C}'_P)]}_{\mathbf{Y}'_P(s)} - \underbrace{s^2 \mathbf{C}'_C{}^T (\mathbf{I} + s\mathbf{C}'_I)^{-1} \mathbf{C}'_C}_{\mathbf{Y}'_I(s)} \mathbf{x}'_P = \mathbf{B}_P \mathbf{u} \quad (10)$$

$$\mathbf{Y}'(s) = \mathbf{Y}'_P(s) - \mathbf{Y}'_I(s), \quad (11)$$

where:

$$\mathbf{G}'_P = \mathbf{G}_P - \mathbf{G}_C^T \mathbf{M}, \quad \mathbf{M} = \mathbf{G}_I^{-1} \mathbf{G}_C \quad (12)$$

$$\mathbf{C}'_P = \mathbf{C}_P - \mathbf{N}^T \mathbf{M} - \mathbf{M}^T \mathbf{C}_C, \quad \mathbf{N} = \mathbf{C}_C - \mathbf{C}_I \mathbf{M} \quad (13)$$

$$\mathbf{C}'_C = \mathbf{L}^{-1} \mathbf{N}, \quad \mathbf{C}'_I = \mathbf{L}^{-1} \mathbf{C}_I \mathbf{L}^{-T}. \quad (14)$$

In (10), the term $\mathbf{Y}'_P(s)$ captures the first two moments of $\mathbf{Y}'(s)$ and is preserved in the reduced model. The reduction is performed on $\mathbf{Y}'_I(s)$ only. In [1] this is done via modal approximation as described next. Using the symmetric eigendecomposition

$\mathbf{C}'_I = \mathbf{U}\Lambda'_I\mathbf{U}^T$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, the system matrices (9) are block diagonalized as follows:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{bmatrix}, \quad \mathbf{G}'' = \mathbf{X}'^T \mathbf{G}' \mathbf{X}' = \begin{bmatrix} \mathbf{G}'_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \mathbf{G}' \quad (15)$$

$$\mathbf{C}'' = \mathbf{X}'^T \mathbf{C}' \mathbf{X}' = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}'_C{}^T \mathbf{U} \\ \mathbf{U}^T \mathbf{C}'_C & \mathbf{U}^T \mathbf{C}'_I \mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}''^T_C \\ \mathbf{C}''_C & \Lambda'_I \end{bmatrix} \quad (16)$$

$$\mathbf{Y}''(s) = \mathbf{Y}'_P(s) - s^2 [\mathbf{C}''^T_C (\mathbf{I} + s\Lambda'_I)^{-1} \mathbf{C}''_C] \quad (17)$$

The reduced model is obtained by selecting only k of the n eigenvalues from Λ'_I :

$$\mathbf{Y}''_k(s) = \mathbf{Y}'_P(s) - s^2 \sum_{i=1}^k \frac{\mathbf{r}_i^T \mathbf{r}_i}{1 + s\lambda'_i}, \quad \mathbf{r}_i^T = \mathbf{C}'_C{}^T \mathbf{U}_{[:,1:k]}, \quad \lambda'_i = \Lambda'_{I[i,i]}. \quad (18)$$

In [1], a selection criterion for λ'_i , $i = 1 \dots k$ is proposed, based on a user-specified error and a maximum frequency. These eigenmodes are computed in [1] via the Lanczos algorithm. The criterion proposed in [3, 10] can also be used to compute the dominant eigenmodes λ'_i via SAMDP.

The advantage of the PACT reduction method is the preservation of the first two moments of $\mathbf{Y}(s)$ in $\mathbf{Y}'_P(s)$. This ensures that the DC and offset of the response is approximated well in the reduced model. The main costs of such an approach are: (1) performing a Cholesky factorization of \mathbf{C}'_I (which becomes expensive when n is very large), (2) solving an eigenvalue problem from a dense \mathbf{C}'_I matrix and, most importantly, (3) the fill-in in the port block matrices \mathbf{G}'_P , \mathbf{C}'_P and in \mathbf{C}'_C . It turns out that (2) can be solved more efficiently by keeping \mathbf{C}'_I as a product of sparse matrices during computation, and will be addressed elsewhere. Avoiding problems (1) and (3) however require new strategies to improve sparsity, and are presented in Sect. 3.2. The fill-in introduced in \mathbf{G}'_P , \mathbf{C}'_P becomes especially important for RC netlists with many terminals [$p \sim O(10^3)$]. Compared to the original model where the port blocks \mathbf{G}_P and \mathbf{C}_P were sparse, the dense \mathbf{G}'_P , \mathbf{C}'_P will yield many R and C components during synthesis, resulting in a reduced netlist where almost all the nodes are interconnected. Simulating such netlists might require longer time measures than the original circuit simulation, hence sparser reduced models (and netlists) are desired. Next, we present several ideas for improving the sparsity of RC reduced models via a combination of tools including: netlist partitioning, graph-based node reordering strategies, and efficient algorithms for modal approximation.

3.2 Improved graph-based method: SparseMA

In this section we present an improved model reduction method for RC circuits, which overcomes the disadvantages of PACT: it requires no matrix factorizations prior to reduction, performs all numerical computations on sparse matrices, and

most importantly, preserves the sparsity of the matrix blocks corresponding to the external nodes. The method is called *sparse modal approximation (SparseMA)* and uses tools from graph theory to identify a partitioning and reordering of nodes that, when applied prior to the model reduction step, can significantly improve the sparsity of the reduced model.

The idea is to reorder the nodes in the *RC* netlist so that some of the internal nodes (m) are promoted as external nodes, together with the circuit terminals (p). We will denote as *selected nodes* the collection of $p + m$ terminals and promoted internal nodes. The $n - m$ internal nodes are the *remaining nodes*. Supposing one has already identified such a partitioning of nodes, the following structure is revealed, where without loss of generality we assume the selected nodes appear in the border of the \mathbf{G} and \mathbf{C} matrices:

$$\left(\begin{bmatrix} \mathbf{G}_R & \mathbf{G}_K \\ \mathbf{G}_K^T & \mathbf{G}_S \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_R & \mathbf{C}_K \\ \mathbf{C}_K^T & \mathbf{C}_S \end{bmatrix} \right) \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_S \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_S \end{bmatrix} \mathbf{u}. \quad (19)$$

Note that in \mathbf{B}_S the rows corresponding to the promoted m internal nodes are still zero. Similarly to (7), the admittance is expressed as:

$$\underbrace{[(\mathbf{G}_S + s\mathbf{C}_S)]}_{\mathbf{Y}_S(s)} - \underbrace{(\mathbf{G}_K + s\mathbf{C}_K)^T (\mathbf{G}_R + s\mathbf{C}_R)^{-1} (\mathbf{G}_K + s\mathbf{C}_K)}_{\mathbf{Y}_R(s)} \mathbf{x}_S = \mathbf{B}_S \mathbf{u} \quad (20)$$

$$\mathbf{Y}(s) = \mathbf{Y}_S(s) - \mathbf{Y}_R(s). \quad (21)$$

Recall that reducing $\mathbf{Y}_I(s)$ directly from the simple partitioning (6) and (7) is not a method of choice, because by preserving $\mathbf{Y}_P(s)$ only, the DC and offset of $\mathbf{Y}(s)$ would not be accurately matched. Using instead the improved partitioning (19) and (20), one aims at better approximating the DC and offset of $\mathbf{Y}(s)$ by preserving $\mathbf{Y}_S(s)$ (which now encaptures not only the external nodes but also a subset of the internal nodes). Finding the partitioning (19) only requires a reordering of nodes, thus no Cholesky factorization or fill-introducing congruence transformation is needed prior to the MOR step. One can reduce $\mathbf{Y}_R(s)$ directly with modal approximation (via frequency dependent SAMDP), and preserve the sparsity of the extended port blocks from $\mathbf{Y}_S(s)$.

By interpolating k dominant eigenmodes from the symmetric eigendecomposition $[\Lambda_R, \mathbf{V}] = \text{eig}(-\mathbf{G}_R, \mathbf{C}_R)$, the reduced model is obtained:

$$\mathbf{Y}_k(s) = \mathbf{Y}_S(s) - \sum_{i=1}^k \frac{\mathbf{q}_i^T \mathbf{q}_i}{1 + s\lambda_i}, \quad \mathbf{q}_i^T = (\mathbf{G}_K + s\mathbf{C}_K)^T \mathbf{V}_{[:,1:k]}, \quad \lambda_i = \Lambda_{R[i,i]}. \quad (22)$$

In matrix terms, the reduced model is easily constructed by re-connecting the preserved selected matrix blocks to the reduced blocks:

$$\left(\begin{bmatrix} \widehat{\mathbf{G}}_R & \widehat{\mathbf{G}}_K \\ \widehat{\mathbf{G}}_K^T & \widehat{\mathbf{G}}_S \end{bmatrix} + s \begin{bmatrix} \widehat{\mathbf{C}}_R & \widehat{\mathbf{C}}_K \\ \widehat{\mathbf{C}}_K^T & \widehat{\mathbf{C}}_S \end{bmatrix} \right) \begin{bmatrix} \widehat{\mathbf{x}}_R \\ \widehat{\mathbf{x}}_S \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_S \end{bmatrix} \mathbf{u}, \quad (23)$$

where:

$$\widehat{\mathbf{G}}_R = \mathbf{V}_{[:,1:k]}^T \mathbf{G}_R \mathbf{V}_{[:,1:k]} \rightarrow \text{diagonal}, \quad \widehat{\mathbf{G}}_K = \mathbf{V}_{[:,1:k]}^T \mathbf{G}_K, \quad \mathbf{G}_S \rightarrow \text{sparse} \quad (24)$$

$$\widehat{\mathbf{C}}_R = \mathbf{V}_{[:,1:k]}^T \mathbf{C}_R \mathbf{V}_{[:,1:k]} \rightarrow \text{diagonal}, \quad \widehat{\mathbf{C}}_K = \mathbf{V}_{[:,1:k]}^T \mathbf{C}_K, \quad \mathbf{C}_S \rightarrow \text{sparse}. \quad (25)$$

The remaining problem is how to determine the selected nodes and the partitioning (19). Inspired from the results obtained for R networks, we propose to first find the permutation \mathbf{P} which identifies the strongly connected components (scs) of \mathbf{G} . Both \mathbf{G} and \mathbf{C} are reordered according to \mathbf{P} , revealing the structure (19). With this permutation, the circuit terminals are redistributed according to the scs of \mathbf{G} , and several clusters of nodes can be identified: a large component consisting of internal nodes and very few (or no) terminals, and clusters formed each by internal nodes plus some terminals. We propose to leave all clusters consisting of internal nodes and terminals intact, and denote these nodes as the *selected nodes* mentioned above. If there are still terminals outside these clusters, they are added to these selected nodes and complete the blocks \mathbf{G}_S , \mathbf{C}_S . The remaining cluster of internal nodes forms \mathbf{G}_R and \mathbf{C}_R . The model reduction step is performed on \mathbf{G}_R and \mathbf{C}_R (and implicitly on \mathbf{G}_K and \mathbf{C}_K). We also note that matrices \mathbf{G}_K and \mathbf{C}_K resulting from this partitioning usually have many zero columns, thus $\widehat{\mathbf{G}}_K$ and $\widehat{\mathbf{C}}_K$ will preserve these zero columns.

The procedure is illustrated in Sect. 4 through a medium-sized example. Larger netlists can be treated via a similar reordering and partitioning strategy, possibly in a recursive manner (for instance when after an initial reordering the number of selected nodes is too large, the same partitioning strategy could be re-applied to \mathbf{G}_S and \mathbf{C}_S and further reduce these blocks). Certainly, other reorderings of \mathbf{G} and \mathbf{C} could be exploited, for instance according to a permutation which identifies the scs of \mathbf{C} instead of \mathbf{G} . The choice for either using \mathbf{G} or \mathbf{C} to determine the permutation \mathbf{P} is made according to the structure of the underlying system and may depend on the application. We also emphasize that the reduced models for both PACT and SparseMA are passive and therefore also stable. Passivity is ensured by the fact that all transformations applied throughout are congruence transformations on symmetric positive definite matrices, thus the reduced system matrices remain symmetric positive definite.

4 Numerical results

The graph-based reduction procedures were applied on several networks resulting from parasitic extraction. We present results for both R and RC networks.

4.1 *R network reduction*

Table 1 shows results for three resistor networks of realistic interconnect layouts. The number of nodes is reduced by a factor > 10 and the number of resistors by a factor > 3 . As a result, the computing time for calculating path resistances in the original network (including nonlinear elements such as diodes) is 10 times smaller.

Table 1 Results of reduction algorithm

	Network I		Network II		Network III	
	Original	Reduced	Original	Reduced	Original	Reduced
#external nodes	274		3399		1978	
#internal nodes	5558	516	99112	6012	101571	1902
#resistors	8997	1505	161183	62685	164213	39011
CPU time	10 s	1 s	67 hrs	7 hrs	20 hrs	2 hrs
Speed up	10x		9.5x		10x	

4.2 *RC network reduction*

We reduce an *RC* netlist with $n = 3231$ internal nodes and $p = 22$ terminals (external nodes). The structure of the original \mathbf{G} and \mathbf{C} matrices is shown in Figures 2 and 3, where the $p = 22$ terminals correspond to their first 22 rows and columns.

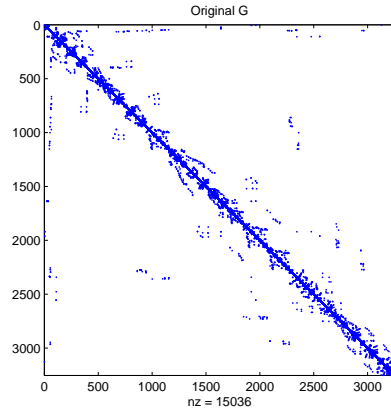


Fig. 2 Original \mathbf{G} matrix

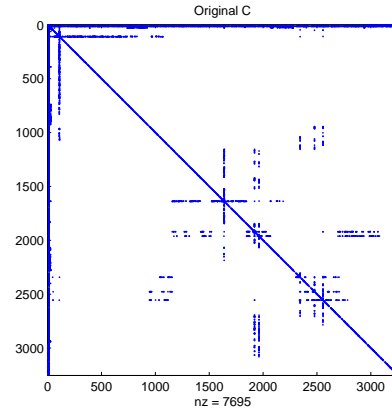


Fig. 3 Original \mathbf{C} matrix

The permutation revealing the strongly connected components of \mathbf{G} reorders the matrices as shown in Figures 4 and 5. The reordering is especially visible in the “arrow-form” capacitance matrix. There, the $p = 22$ terminal nodes together with $m = 40$ internal nodes are promoted to the border, revealing the 62 selected nodes that will be preserved in the reduced model [i. e. the \mathbf{G}_S and \mathbf{C}_S blocks in (19)]. The

first $n - m = 3191$ nodes are the remaining internal nodes and form the \mathbf{G}_R and \mathbf{C}_R blocks in (19). The \mathbf{G}_K block has only 1 non-zero column, and also in \mathbf{C}_K many zero columns can be identified.

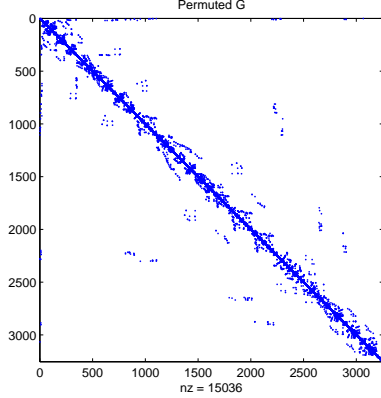


Fig. 4 Permutated \mathbf{G} according to $\text{scc}(\mathbf{G})$

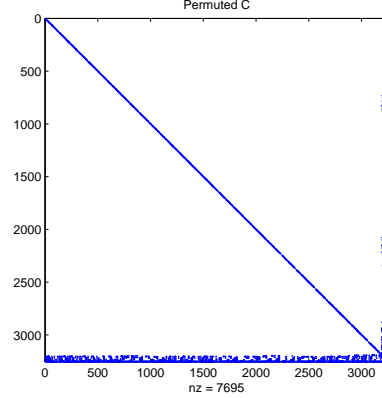


Fig. 5 Permutated \mathbf{C} according to $\text{scc}(\mathbf{G})$

The reduced SparseMA model is obtained according to (22) and (23) and is shown in Figures 6 and 7. The internal blocks \mathbf{G}_R and \mathbf{C}_R were reduced from dimension 3191 to $\hat{\mathbf{G}}_R$ and $\hat{\mathbf{C}}_R$ of dimension $k = 7$, by interpolating the 7 most dominant eigenmodes of $[\Lambda_R, \mathbf{V}] = \text{eig}(-\mathbf{G}_R, \mathbf{C}_R)$. Note that $\hat{\mathbf{G}}_R$ and $\hat{\mathbf{C}}_R$ are diagonal. The selected 62 nodes corresponding to the \mathbf{G}_S and \mathbf{C}_S blocks are preserved, evidently preserving sparsity. The only fill-in introduced by the proposed reduction procedure is in the non-zero columns of $\hat{\mathbf{G}}_K$ and $\hat{\mathbf{C}}_K$. It is worth noticing that $\hat{\mathbf{G}}_K$ only has 1 non-zero column, thus remains sparse.

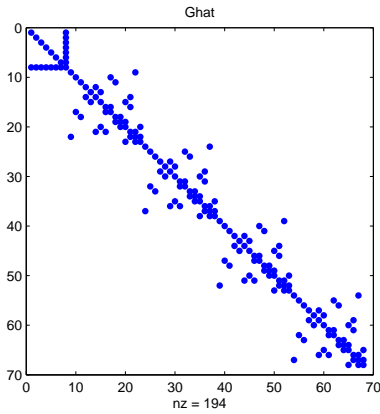


Fig. 6 Reduced \mathbf{G} matrix with Sparse MA

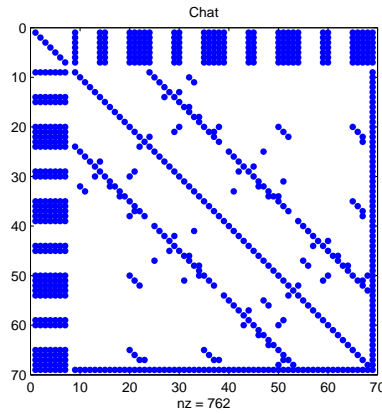


Fig. 7 Reduced \mathbf{C} matrix with Sparse MA

The sparsity structure of the PACT reduced model (18) is shown in Figures 13 and 9. The blocks corresponding to the first 22 nodes (the preserved external nodes) are full, as are the capacitive connection blocks to the reduced internal part. Only the reduced internal blocks remain sparse (diagonal).

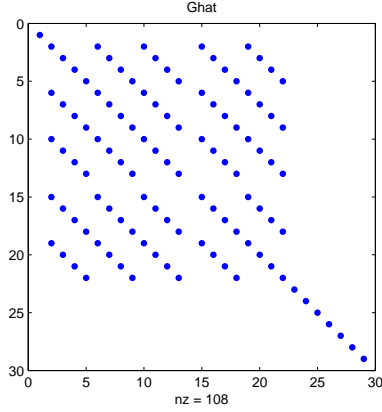


Fig. 8 Reduced G matrix with PACT

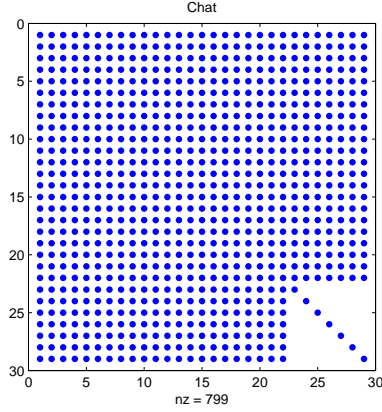


Fig. 9 Reduced C matrix with PACT

Aside from sparsity preservation, one is interested in the quality of the approximation for the reduced model. In Fig. 10, we show that the SparseMA model accurately matches the original response for a wide frequency range ($1\text{ Hz} \rightarrow 10\text{ THz}$). The Pstar [6] simulations of the synthesized model are identical to the Matlab simulations (the synthesized model was obtained via the RLCSYN unstamping procedure [7, 30]). In Fig. 11, the relative errors between the original model and three reduced models are presented: SparseMA, PACT and the commercial software Jivaro [18]. The SparseMA model is the most accurate for the entire frequency range.

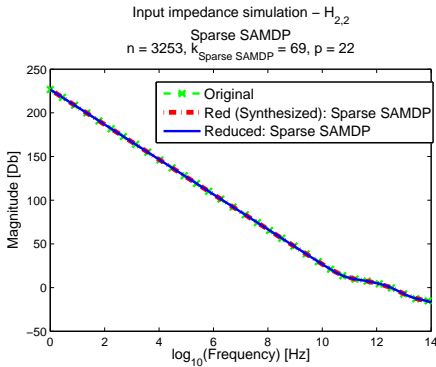


Fig. 10 AC Simulation 1: Original, reduced (Sparse MA) and synthesized model

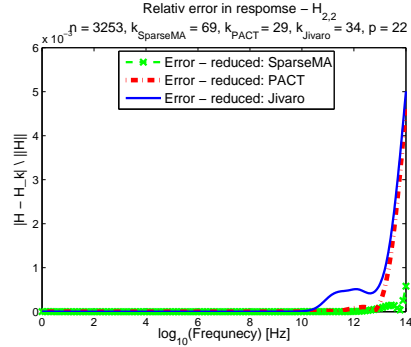


Fig. 11 AC Simulation 1: Relative error between original and reduced models (SparseMA, Pact, Jivaro)

Fig. 12 shows a different AC circuit simulation, where the SparseMA model performs comparably to the reduced model obtained with the commercial software Jivaro [18]. Finally, the transient simulation in Fig. 13 confirms that the SparseMA model is both accurate and stable.

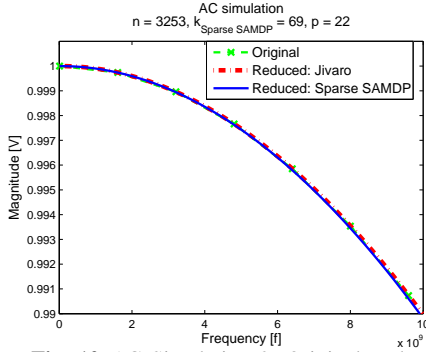


Fig. 12 AC Simulation 2: Original, reduced (Sparse MA) and reduced (Jivaro)

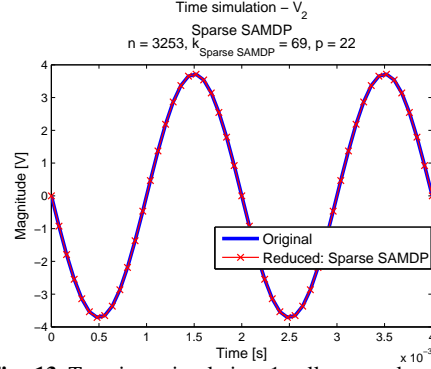


Fig. 13 Transient simulation 1: all external nodes grounded and voltage measured at node 2. Original and reduced (Sparse MA - synthesized)

Table 2 shows the reduction results for the RC network. For the 3 reduced models: SparseMA, PACT and Jivaro we assess the effect of the reduction by means of several factors. With all methods, both the number of nodes and the number of circuit elements was reduced significantly, resulting in at least 68x speed-up in AC simulation time. It should be noted that the SparseMA model and the Jivaro model have lower ratios of $\frac{\#elements}{\#unknowns}$ and $\frac{\#elements}{\#int.nodes}$ than the PACT model. Even though the Jivaro and the PACT model are faster to simulate for this network, the SparseMA model gives a good trade-off between approximation quality, sparsity preservation and CPU speed-up. Recall that the matrix blocks corresponding to the circuit terminals become dense with PACT, but remain sparse with SparseMA. As for circuits with more terminals $\sim O(10^3)$ the corresponding matrix blocks become larger, preserving their sparsity via SparseMA is an additional advantage. Hence, the improvement on simulation time could be greater with SparseMA when applied on larger models with many terminals.

5 Concluding remarks

New approaches were presented for reducing R and RC circuits with multi-terminals, using tools from graph theory. It was shown how netlist partitioning and node re-ordering strategies can be combined with existing model reduction techniques, to improve the sparsity of the reduced RC models and implicitly their simulation time. The proposed sparsity preserving method, SparseMA, performs comparably to the

Table 2 Results with SparseMA reduction on *RC* netlist

	Original	Red. SparseMA	Red. PACT	Red. Jivaro
#external nodes		22		
#internal nodes	3231	47	7	12
#unknowns	3253	69	29	34
#resistors	7944	78	68	28
#capacitors	3466	383	414	97
#elements	3.53	9.8	68.8	10.4
#int. nodes #elements	3.5	6.7	16.6	3.67
#unknowns				
CPU time	6.8 s	0.1 s	0.06	0.02 s
Speed up		68x	113x	340x

commercial tool Jivaro. Future work will investigate how similar strategies can be applied to *RC* models with many more terminals [$\sim O(10^3)$] and to *RLCk* netlists.

Acknowledgements The first author acknowledges the Marie Curie Fellowship Programme, COMSON project MRTN-CT-2005-019417, for the work within TU/Eindhoven and NXP Semiconductors, Eindhoven, the Netherlands. The second author was supported by the O-MOORE-NICE! project, Marie Curie Actions of FP6, MTKI-CT-2006-042477.

References

1. Kerns, K.J., Yang, A.T.: Stable and efficient reduction of large, multiport networks by pole analysis via congruence transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **16**(7), 734–744 (1997)
2. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D. thesis, Utrecht University (2007)
3. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration *IEEE Trans. Power Syst.* **21**(4), 1471–1483 (2006)
4. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration *IEEE Trans. Power Syst.* **21**(3), 1218–1226 (2006)
5. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. Tech. Rep. 2008/00316, NXP Semiconductors (2009)
6. NXP Semiconductors: Pstar. <http://www.nxp.com>
7. Yang, F., Zeng, X., Su, Y., Zhou, D.: RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI. *Commun. Comput. Phys* **3**(2), 376–396 (2008)
8. J. Rommes, P. Lenaers, and W. H. A. Schilders, “Reduction of large resistor networks,” Presented at SCEE 2008, Espoo, Finland (2008).
9. J. Rommes, and W. H. A. Schilders, “Efficient methods for large resistor networks” Submitted for publication (2008).
10. J. Rommes, “Model reduction using modal approximation” COMSON Handbook (2009).
11. W. J. McCalla, *Fundamentals of Computer Aided Circuit Simulation*, 1st ed. Kluwer Academic Publishers, 1988.
12. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*, 1st ed. The MIT Press, 1990.
13. T. A. Davis, “Suite Sparse: a Suite of Sparse matrix packages,” <http://www.cise.ufl.edu/research/sparse/SuiteSparse/>.
14. Kolyer, J.M., Watson, D.: *ESD: From A To Z*. Springer (1996)

15. Electrostatic discharge association. <http://www.esda.org>
16. Chua, L.O., Lin, P.: Computer aided analysis of electric circuits: algorithms and computational techniques, first edn. Prentice Hall (1975)
17. Phillips, J.R., Silveira, L.M.: Poor man's tbr: A simple model reduction scheme. *IEEE Trans. CAD Circ. Syst.* **24**(1), 283–288 (2005)
18. Edxact: Jivaro. <http://www.edxact.com>
19. Cadence: AssuraRCX. <http://www.cadence.com>
20. Miettinen, P., Honkala, M., Roos, J.: Using metis and hmetis algorithms in circuit partitioning. *Circuit Theory Laboratory Report Series CT-49*, Helsinki University of Technology (2006)
21. Karypis, G., Kumar, V.: METIS, A software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. <http://glaros.dtc.umn.edu/gkhome/metis/>
22. Freund, R.W.: PRIM: Structure-preserving reduced-order interconnect macromodeling. In: *Technical Digest of the 2004 IEEE/ACM International Conference on CAD*, pp. 80–87 (2004)
23. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): *Model Order Reduction: Theory, Research Aspects and Applications*, *Mathematics in Industry*, vol. 13. Springer (2008)
24. Amestoy, P.R., Davis, T.A., Duff, I.S.: An Approximate Minimum Degree Ordering Algorithm. *SIAM J. Matrix Anal. Appl.* **17**(4), 886–905 (1996)
25. Zečević, A.I., Šiljak, D.D.: Balanced decompositions of sparse systems for multilevel prallel processing. *IEEE Trans. Circ. Syst.–I:Fund. Theory and Appl.* **41**(3), 220–233 (1994)
26. Zhou, Q., Sun, K., Mohanram, K., Sorensen, D.C.: Large power grid analysis using domain decomposition. In: *Proc. Design Automation and Test in Europe*, pp. 27–32 (2006)
27. Duff, I.S., Erisman, A.M., Reid, J.K.: *Direct methods for sparse matrices*. Oxford: Clarendon Press (1986)
28. Golub, G.H., van Loan, C.F.: *Matrix Computations*, third edn. John Hopkins University Press (1996)
29. Ionutiu, R., Rommes, J.: “A framework for synthesis of reduced order models” *COMSON Handbook* (2009).
30. Ionutiu, R., Rommes, J.: “Passivity Preserving Model Reduction using the Dominant Spectral Zero Method” *COMSON Handbook* (2009).