Graph-Based Sparsification and Synthesis of Dense Matrices in the Reduction of RLC Circuits

Charalampos Antoniadis, *Associate Member, IEEE,* Nestor Evmorfopoulos, and Georgios Stamoulis

***Abstract*—The integration of more components into modern ICs has led to very large RLC parasitic networks consisting of million of nodes, which have to be simulated in many times or frequencies to verify the proper operation of the chip. Model Order Reduction techniques have been employed routinely to substitute the large scale parasitic model by a model of lower order with similar response at the input/output ports. However, established MOR techniques generally result in dense system matrices that render their simulation impractical. To this end, in this paper we propose a methodology for the sparsification of the dense circuit matrices resulting from Model Order Reduction of general RLC circuits, which employs a sequence of algorithms based on the computation of the nearest diagonally dominant matrix and the sparsification of the corresponding graph. In addition, we describe a procedure for synthesizing the sparsified reduced-order model into an RLC circuit with only positive elements. Experimental results indicate that a high sparsity ratio of the reduced system matrices can be achieved with very small loss of accuracy.**

***Index Terms*—MOR, Laplacian matrix, Sparsification, RLC Circuit Synthesis**

I. Introduction

A very crucial step in a typical integrated circuit (IC) design flow entails the simulation of the electrical models extracted from the key IC subsystems (power grid, interconnect structures, substrate regions). Due to the contemporary levels of technology scaling all these electrical models become enormous, and their their functional simulation requires solving systems of equations with dimension that can reach several millions or even billions. Model Order Reduction (MOR) techniques can be harnessed to deal with such huge systems, replacing the very large electrical models with much smaller ones that approximate the input/output behavior of the original models.

The most common and successful MOR methods rely on the mathematical description of a circuit as a linear time-invariant (LTI) system, and employ the concept of projection onto lower-dimensional spaces to obtain the reduced-order models. Most prominent among them are the classes of moment-matching [1] and truncation-type methods [2]. The process of projection, however, yields reduced-order models with dense matrices whose cost of employing in simulation can easily overshadow the benefits from the order reduction (as is confirmed by experiments in [3]).

Previous methods attempting to address the problem of dense matrices resulting from MOR have been proposed in the literature. In particular, [4] firstly divides the circuit nodes into a group corresponding to ports that have to be preserved and a group of internal nodes that can be eliminated, and then finds the Schur complement of the system and performs sparse matrix manipulations on top of it. The approach in [5] employs partitioning of the circuit into subcircuits and then applies a similar methodology to [4] on each individual subcircuit. Also, [6] utilizes circuit partitioning and then substitutes each subcircuit by suitable RLC macromodels that capture the low-order moments of the admittance matrix of the subcircuit. Furthermore, [7] after deriving a description of the circuit with respect to node voltages and magnetic flux, reorders system matrices in Bordered Block Diagonal form and then applies a standard MOR procedure. A major problem of the aforementioned methods is that they rely more or less on circuit partitioning, which needs to utilize circuit-specific information and its effectiveness is heavily dependent on the given circuit. None of them provides a general and circuit-independent framework to address the sparsification problem of the dense matrices resulting from the application of MOR procedures.

There exists another class of MOR methods, namely node elimination methods [8][9], which are based on successively eliminating nodes of the initial circuit and distributing the error of approximation on the components incident to the remaining nodes. These methods produce by nature sparse reduced-order models, but their efficiency varies widely among different circuits and they typically do not match the accuracy and reduction levels of projection-based MOR methods.

In this paper we focus on projection-based MOR and propose an approach for the sparsification of the resultant dense matrices, which employs a sequence of algorithms based on the computation of the nearest diagonally dominant matrix and the subsequent sparsification of the corresponding graph. The key idea in our methodology is that we transform the problem of the sparsification of the reduced model matrices to the sparsification of the nearest matrices corresponding to a graph, in order to take advantage of efficient graph sparsification techniques. In addition we describe how the resulting sparse model can be realized into a circuit with only positive elements, which to the best of our knowledge is not offered by any of previous sparsification works for projection-based MOR methods. Experimental results demonstrate the efficiency and accuracy of the sparse reduced models that are produced by the proposed methodology.

Preliminary versions of this paper appeared in [10] and [11]. This paper extends and improves these works in the following major points: (i) Appropriate congruence transformations are

Charalampos Antoniadis, Nestor Evmorfopoulos and Georgios Stamoulis are with the Department of Electrical and Computer Engineering, University of Thessaly, 38221 Volos, Greece, e-mail:({haadonia,nestevmo,georges}@e-ce.uth.gr)

introduced (Section III-B) by which the conductance matrix is left intact (only the capacitance and inductance matrices are approximated by their nearest diagonally dominant matrices) and thus the accuracy of the proposed methodology is significantly improved (ii) The procedure of conversion of a general diagonally dominant matrix to a graph-corresponding matrix (Section III-C) has been improved in a way that eliminates a computational post-step and has straightforward mapping to the circuit synthesis procedure (iii) The graph sparsification algorithm (Section IV) has been improved by replacing its inherent random sampling by an equivalent closed formula based on probabilistic expectation, in a way that reduces the computational and storage requirements and at the same time eliminates the algorithm randomness so that it always produces the same sparse graph (iv) An entirely new section has been added (Section VI, along with an appendix containing a relevant proof) that describes the synthesis procedure of the resulting sparse model to an RLC circuit with positive elements, and the description is supplemented by a running example (v) The experimental section (Section VII) has been enriched with results that justify the improved accuracy of the sparse reduced models.

The rest of the paper is organized as follows. Section II sets up the necessary framework, makes an overview of MOR and introduces the important class of Laplacian matrices. Section III firstly presents an algorithm that computes the nearest Symmetric and Diagonally Dominant (SDD) matrix to a given matrix and then gives a procedure to convert an SDD matrix to Laplacian. Section IV presents a sparsification algorithm of a Laplacian matrix originating from spectral graph theory and the concept of effective resistance. Section V describes the full proposed algorithm to efficiently sparsify the dense matrices resulting from MOR. Section VI presents how the sparse model produced by the proposed methodology can be synthesized into an RLC circuit. Section VII presents the experimental results of the proposed methodology, and conclusions are drawn in Section VIII.

II. Background

A. Description of circuit models

Consider an RLC circuit with n nodes (excluding ground), m inductive branches (with possible mutual coupling between them), g conductive branches, c capacitive branches, p inputs and q outputs, which is described according to the Modified Nodal Analysis (MNA) formulation in the time domain as follows:

$$\begin{bmatrix} \mathbf{A}_{cr}\mathbf{C}_d\mathbf{A}_{cr}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{v}(t)}{dt} \\ \frac{d\mathbf{i}(t)}{dt} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{gr}\mathbf{G}_d\mathbf{A}_{gr}^T & \mathbf{A}_m \\ -\mathbf{A}_m^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{i}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(t)$$
$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{E} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{i}(t) \end{bmatrix} \tag{1}$$

where $\mathbf{v} \in \Re^n$, $\mathbf{i} \in \Re^m$, $\mathbf{u} \in \Re^p$, $\mathbf{y} \in \Re^q$ are the vectors of node voltages, inductive branch currents, input excitations (with voltage sources transformed into Norton-equivalent current sources) and output measurements respectively, $\mathbf{B} \in \Re^{n\times p}$ and $\mathbf{E} \in \Re^{q\times n}$ are the input-to-node and node-to-output connectivity matrices, $\mathbf{A}_{gr} \in \Re^{n\times g}$ and $\mathbf{A}_{cr} \in \Re^{n\times c}$ are the node-to-branch *reduced* incidence matrices (without the row corresponding to ground node), $\mathbf{G}_d \in \Re^{g\times g}$ and $\mathbf{C}_d \in \Re^{c\times c}$ are positive diagonal matrices containing the branch conductances and branch capacitances respectively, $\mathbf{M} \in \Re^{m\times m}$ is the branch inductance matrix (with self-inductances at the diagonal and mutual inductances off-diagonally) and $\mathbf{A}_m \in \Re^{n\times m}$ is the corresponding node-to-branch incidence matrix. It is well-known [12] that the products

$$\mathbf{G}_n \equiv \mathbf{A}_{gr}\mathbf{G}_d\mathbf{A}_{gr}^T, \quad \mathbf{C}_n \equiv \mathbf{A}_{cr}\mathbf{C}_d\mathbf{A}_{cr}^T \tag{2}$$

are symmetric and diagonally dominant (SDD) matrices with nonnegative diagonal elements and nonpositive off-diagonal elements, i.e. $g_{ii} \geq 0$, $c_{ii} \geq 0$ (sum of conductances or capacitances connected to node i), $g_{ij} = g_{ji} \leq 0$, $c_{ij} = c_{ji} \leq 0$ (opposite of conductance or capacitance between nodes i and j), and $g_{ii} \geq -\sum_{\substack{j=1 \\ j\neq i}}^{n} g_{ij}$, $c_{ii} \geq -\sum_{\substack{j=1 \\ j\neq i}}^{n} c_{ij}$, $\forall i = 1, \ldots n$. These types of matrices are subsets of the broader class of *M-matrices* [13], and we are going to refer to them as "circuit-type" M-matrices. The branch inductance matrix $\mathbf{M}$ is not diagonally dominant, but it is also well-known [14] that its inverse $\mathbf{M}^{-1}$ (called the *reluctance* matrix) is SDD with nonnegative diagonal elements and nonpositive off-diagonal elements, i.e a circuit-type M-matrix.

The fact that $\mathbf{A}_{gr}$ and $\mathbf{A}_{cr}$ are reduced incidence matrices means that both $\mathbf{G}_n = \mathbf{A}_{gr}\mathbf{G}_d\mathbf{A}_{gr}^T$ and $\mathbf{C}_n = \mathbf{A}_{cr}\mathbf{C}_d\mathbf{A}_{cr}^T$ will have at least one row sum greater than zero, i.e. $g_{ii} > -\sum_{j=1,j\neq i}^{n} g_{ij}$ and $c_{ii} > -\sum_{j=1,j\neq i}^{n} c_{ij}$ for the nodes i where a conductance or capacitance is connected to ground. These can be written as $\mathbf{G}_n = \mathbf{D}_g + \mathbf{A}_g\mathbf{G}\mathbf{A}_g^T$ and $\mathbf{C}_n = \mathbf{D}_c + \mathbf{A}_c\mathbf{C}\mathbf{A}_c^T$, where $\mathbf{D}_g$ and $\mathbf{D}_c$ are diagonal matrices with the conductances and capacitances to ground, $\mathbf{A}_g$ and $\mathbf{A}_c$ are normal (not reduced) incidence matrices for the branches remaining after ground node (and all branches connected to it) is removed, and $\mathbf{G}$, $\mathbf{C}$ are diagonal matrices with the conductances and capacitances of these branches. The products:

$$\mathbf{L}_g \equiv \mathbf{A}_g\mathbf{G}\mathbf{A}_g^T, \quad \mathbf{L}_c \equiv \mathbf{A}_c\mathbf{C}\mathbf{A}_c^T \tag{3}$$

are known as the *Laplacian* matrices [15] of the weighted graphs consisting of the conductive branches and the capacitive branches not connected to ground, and having as weights the conductances and capacitances of these branches. Thus, any circuit-type M-matrix can be written as the sum of a nonnegative diagonal matrix and the Laplacian matrix of a weighted graph.

B. Laplacian matrices

Given a weighted graph $G = (V, E, w)$ with set of vertices (nodes) $V = \{1, 2, \ldots, n\}$, set of edges (branches) $E = \{(i, j) |\ i, j \in V\}$, and weight function $w(i, j)$, the Laplacian matrix of G is defined as follows:

$$\mathbf{L}_G = \mathbf{A}_G \mathbf{W} \mathbf{A}_G^T = \sum_{(i,j)\in E} w(i,j)(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^T \quad (4)$$

where $\mathbf{W}$ is a diagonal matrix with the edge weights and $\mathbf{A}_G$ is the vertex-to-edge incidence matrix with columns for every edge (i,j) that equal $\mathbf{e}_i - \mathbf{e}_j$, where $\mathbf{e}_i$ is the elementary unit vector with 1 at the position i and 0's everywhere else. Because each row and column sum of $\mathbf{L}_G$ equals zero, a Laplacian matrix is always rank-deficient and there does not exist a regular inverse of $\mathbf{L}_G$. However, the algorithms presented in this paper employ $\mathbf{L}_G$ only on vectors existing in its column space $\mathcal{R}(\mathbf{L}_G)$ (like $(\mathbf{e}_i - \mathbf{e}_j)$), where the *Moore-Penrose pseudoinverse* $\mathbf{L}_G^+$ acts as a normal inverse, i.e. for each $\mathbf{y} \in \mathcal{R}(\mathbf{L}_G)$, $\mathbf{L}_G^+\mathbf{y}$ is the unique $\mathbf{x} \in \mathcal{R}(\mathbf{L}_G)$ such that $\mathbf{L}_G\mathbf{x} = \mathbf{y}$.

C. Model Order Reduction (MOR)

MOR techniques aim at approximating the model of (1) by another model of reduced order $r < n+m$, through a process of projecting the model matrices onto a lower-dimensional subspace of dimension r. Instead of projecting the original block matrices of (1), structure-preserving MOR techniques like [16] and [7] work on the individual blocks $\mathbf{G}_n$, $\mathbf{C}_n$, $\mathbf{M}$, the difference between them being that the former involves the inductance matrix $\mathbf{M}$ while the latter deals with the reluctance matrix $\mathbf{M}^{-1}$.

A different approach is to solve the lower m equations of (1) with respect to $\mathbf{i}(t)$ and substitute to the upper n equations, in order to obtain the *second-order* expression (assuming initial condition $\mathbf{i}(0) = \mathbf{0}$):

$$\begin{aligned} \mathbf{C}_n \frac{d\mathbf{v}(t)}{dt} + \mathbf{G}_n \mathbf{v}(t) + \mathbf{\Gamma} \int_0^t \mathbf{v}(t)dt &= \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{E}\mathbf{v}(t) \end{aligned} \quad (5)$$

where $\mathbf{\Gamma} = \mathbf{A}_m \mathbf{M}^{-1} \mathbf{A}_m^T$.

MOR techniques like [17] which work on the above second-order formulation, obtain a reduced-order model through the following matrix transformations:

$$\begin{aligned} \widehat{\mathbf{G}}_n = \mathbf{U}^T \mathbf{G}_n \mathbf{V}, \quad \widehat{\mathbf{C}}_n = \mathbf{U}^T \mathbf{C}_n \mathbf{V} \\ \widehat{\mathbf{\Gamma}} = \mathbf{U}^T \mathbf{\Gamma} \mathbf{V} = (\mathbf{A}_m^T \mathbf{U})^T \mathbf{M}^{-1} (\mathbf{A}_m^T \mathbf{V}) \\ \widehat{\mathbf{B}} = \mathbf{U}^T \mathbf{B}, \quad \widehat{\mathbf{E}} = \mathbf{E}\mathbf{V} \end{aligned} \quad (6)$$

where $\mathbf{U}$, $\mathbf{V} \in \Re^{n\times r}$ are suitable projection matrices that satisfy $\mathbf{U}^T\mathbf{V} = \mathbf{I}_r$ ($\mathbf{I}_r$ is the $r \times r$ identity matrix), so that $\mathbf{V}\mathbf{U}^T$ forms a projection onto the r-dimensional column space of $\mathbf{V}$ and along the nullspace of $\mathbf{U}^T$ [18]. The projection is orthogonal if $\mathbf{U} = \mathbf{V}$ while it is oblique if $\mathbf{U} \neq \mathbf{V}$.

The above projections do not generally preserve the properties of circuit-type M-matrices, i.e. diagonal dominance and nonpositive off-diagonal elements (nonnegative diagonal elements are preserved because MOR matrices are typically positive semi-definite and passive), which allow them to have a direct correspondence to circuits. The biggest problem however, from the projections inherent in MOR is that sparsity is lost, which can render impractical any time or frequency domain simulation involving the solution of linear systems in the MOR matrices, and offsets any benefits from the reduction of order. We will address these problems in the remainder of the paper.

III. Approximation of Projected MOR Matrices by Circuit-type M-matrices

A. Back projection to the nearest SDD matrix

The matrices $\widehat{\mathbf{G}}_n$, $\widehat{\mathbf{C}}_n$, $\widehat{\mathbf{\Gamma}}$ resulting from the projections (6) are not diagonally dominant, but they are expected to be close to a DD matrix (in a suitable matrix norm), since the MOR projections are constructed so that the reduced matrices form good approximations and preserve the dominant eigenvalues of the initial DD matrices $\mathbf{G}_n$, $\mathbf{C}_n$, $\mathbf{M}^{-1}$ (we present experimental confirmation of the proximity of $\widehat{\mathbf{G}}_n$, $\widehat{\mathbf{C}}_n$, $\widehat{\mathbf{\Gamma}}$ to DD matrices in section VII). Therefore, in this section we consider the problem of finding the nearest SDD matrix to a given $r \times r$ matrix $\mathbf{A}$ with nonnegative diagonal elements, in some suitable matrix norm which is chosen to be the Frobenius norm, i.e. the following least-squares problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{A} - \mathbf{X}\|_F = \sqrt{\sum_{i=1}^{r}\sum_{j=1}^{r}(a_{ij} - x_{ij})^2} \\ \text{s.t.} \quad & \mathbf{X} \in \text{SDD} \end{aligned} \quad (7)$$

This constrained optimization problem can be solved by Algorithm 1 [19] which (after initializing $\mathbf{X} = \mathbf{A}$) alternatingly projects $\mathbf{X}$ onto the set of symmetric matrices by $\frac{\mathbf{X}+\mathbf{X}^T}{2}$, and the set of diagonally dominant matrices by Algorithm 2 (whose non-empty intersection determines the feasible set of solutions), until the distance between two consecutive projections reaches a pre-specified tolerance. The DD-projection Algorithm 2 (algorithm NQ from [19]) computes for each non-DD row an "average" of the off-diagonal elements which then adds to the diagonal element while subtracting it from the rest of the elements for this particular row.

The Algorithm 1 can be proven to converge to the nearest SDD matrix, while an analysis of the convergence rate and techniques for improving it have been presented in [20]. As for the computational complexity of every iteration itself, both the symmetric projection $\frac{\mathbf{X}+\mathbf{X}^T}{2}$ and the DD projection entail $O(r^2)$ flops (the "while" loop in Algorithm 3 only takes a few iterations in practice [19]), and also exhibit significant degree of parallelism (since each row is treated independently of the others in Algorithm 2). The quadratic complexity is not a problem, however, since the dimension r of the reduced models dealt with in practice is in the order of at most a few thousand.

B. Accuracy improvement by congruence transformation

It is possible to apply an appropriate congruence transformation (of the type $\mathbf{S}^T\mathbf{A}\mathbf{S}$) to the original matrices $\mathbf{G}_n$, $\mathbf{C}_n$, $\mathbf{\Gamma}$ of (5), which obviates the need to approximate the reduced $\widehat{\mathbf{G}}_n$ by its nearest SDD matrix after MOR. For that we should

Algorithm 1 Find the nearest SDD matrix to an $r \times r$ matrix $\mathbf{A}$ under the Frobenius norm

function $\mathbf{X} = \text{PRJSDD}(\mathbf{A}, tol)$
  $\mathbf{Z} = \mathbf{0}$; $\mathbf{X} = \mathbf{A}$
  repeat
    $\mathbf{S} = \frac{\mathbf{X}+\mathbf{X}^T}{2}$
    $\mathbf{X}_{prv} = \mathbf{X}$
    $\mathbf{X} = prjDD(\mathbf{S} - \mathbf{Z})$
    $\mathbf{Z} = \mathbf{X} - \mathbf{S} + \mathbf{Z}$
  until $\|\mathbf{X} - \mathbf{X}_{prv}\|_F \leq tol$
end function

Algorithm 2 Project an $r \times r$ matrix $\mathbf{A}$ with nonnegative diagonal elements onto the set of DD matrices

function $\mathbf{X} = \text{PRJDD}(\mathbf{A})$
  for i =1 to r **do**
    $\mathbf{a} = \mathbf{A}(i,:)$
    $s = \sum_{j=1, j\neq i}^{r} |\mathbf{a}(j)|$
    if $\mathbf{a}(i) \geq s$ **then**
      x = a
    end if
    if $\mathbf{a}(i) \geq 0$ && $\mathbf{a}(i) < s$ **then**
      x = $prjDDrow(\mathbf{a}, i)$
    end if
    X(*i*,:) **= x**
  end for
end function

enumerate first the $p+q$ port (input/output) nodes[1] and then the $n-(p+q)$ internal nodes, which leads to the following formulation of the second-order description (5):

$$\begin{bmatrix} \mathbf{C}_p & \mathbf{C}_c^T \\ \mathbf{C}_c & \mathbf{C}_i \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{v}_p(t)}{dt} \\ \frac{d\mathbf{v}_i(t)}{dt} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_p & \mathbf{G}_c^T \\ \mathbf{G}_c & \mathbf{G}_i \end{bmatrix} \begin{bmatrix} \mathbf{v}_p(t) \\ \mathbf{v}_i(t) \end{bmatrix} + \begin{bmatrix} \mathbf{\Gamma}_p & \mathbf{\Gamma}_c^T \\ \mathbf{\Gamma}_c & \mathbf{\Gamma}_i \end{bmatrix} \begin{bmatrix} \int_0^t \mathbf{v}_p(t)dt \\ \int_0^t \mathbf{v}_i(t)dt \end{bmatrix} = \begin{bmatrix} \mathbf{B}_p \\ \mathbf{0} \end{bmatrix} \mathbf{u}(t)$$
$$\mathbf{y}(t) = \begin{bmatrix} \mathbf{E}_p & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v}_p(t) \\ \mathbf{v}_i(t) \end{bmatrix} \tag{8}$$

where $\mathbf{v}_p \in \Re^{p+q}$, $\mathbf{v}_i \in \Re^{n-(p+q)}$ are the port and the internal node voltages respectively, $\mathbf{G}_p$, $\mathbf{C}_p$, $\mathbf{\Gamma}_p \in \Re^{(p+q)\times(p+q)}$ represent the conductive, capacitive, and inductive connections between port nodes, $\mathbf{G}_i$, $\mathbf{C}_i$, $\mathbf{\Gamma}_i \in \Re^{(n-(p+q))\times(n-(p+q))}$ describe the corresponding connections between internal nodes, and $\mathbf{G}_c$, $\mathbf{C}_c$, $\mathbf{\Gamma}_c \in \Re^{(n-(p+q))\times(p+q)}$ represent the conductive, capacitive, and inductive connections between port and internal nodes.

Since the $p+q$ port nodes must also exist in the reduced-order model, the process of MOR can be confined to reducing the internal node matrices $\mathbf{G}_i$, $\mathbf{C}_i$, $\mathbf{\Gamma}_i$ (leaving the input/output structure intact), which can be performed via the following projection matrices (with $\mathbf{U}_i^T\mathbf{V}_i = \mathbf{I}_r$):

[1]The number of ports may be less than $p+q$ in case some (or all) of the input and output nodes coincide.

Algorithm 3 Compute the row of the projection of an $r \times r$ matrix $\mathbf{A}$ onto the set of DD matrices

function $\mathbf{x} = \text{PRJDDROW}(\mathbf{a}, i)$
  $d = \sum_{j=1, j\neq i}^{r} |\mathbf{a}(j)|$
  $d = d - \mathbf{a}(i)$
  $\mathbf{a}_{off} = [\mathbf{a}(1), \ldots, \mathbf{a}(i-1), \mathbf{a}(i+1), \ldots, \mathbf{a}(r)]$
  $c =$ #nonzeros($\mathbf{a}_{off}$) + 1
  $b = d/c$
  $s = 1$
  while $s == 1$ **do**
    $s = 0$
    for j =1 to r **do**
      if $\mathbf{a}(j) \neq 0$ && $j \neq i$ **then**
        $aux = \mathbf{a}(j) -$ sign($\mathbf{a}(j)$)$\cdot b$
        if sign(aux)$\cdot$sign($\mathbf{a}(j)$) < 0 **then**
          $d = d - |\mathbf{a}(j)|$
          $c = c - 1$
          $\mathbf{a}(j) = 0$
          $s = 1$
        end if
      end if
    end for
    $b = d/c$
  end while
  for j =1 to r **do**
    if $j \neq i$ **then**
      if $\mathbf{a}(j) == 0$ **then**
        $\mathbf{x}(j) = 0$
      else if $\mathbf{a}(j) > 0$ **then**
        $\mathbf{x}(j) = \mathbf{a}(j) - b$
      else
        $\mathbf{x}(j) = \mathbf{a}(j) + b$
      end if
    else
      $\mathbf{x}(i) = \mathbf{a}(i) + b$
    end if
  end for
end function

$$\mathbf{U} = \begin{bmatrix} \mathbf{I}_{p+q} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_i \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{I}_{p+q} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_i \end{bmatrix} \tag{9}$$

As suggested in [21], the submatrices $\mathbf{U}_i$, $\mathbf{V}_i$ can be derived by computing the full $n \times r$ projection matrices for (8) by any MOR method (e.g. moment-matching), and then partitioning them as $\begin{bmatrix} \mathbf{U}_p \\ \mathbf{U}_i \end{bmatrix}$, $\begin{bmatrix} \mathbf{V}_p \\ \mathbf{V}_i \end{bmatrix}$ and extracting the $\mathbf{U}_i$, $\mathbf{V}_i$ parts to construct (9), instead of applying them in full. Note that in this case, the order of the reduced model will be $(p+q)+r$ instead of r, which should not be a problem in practice if the number of ports is much smaller than the size n of the original model.

Now before the MOR projection of (8) with (9), we can apply the congruence transformation introduced in [22], which preserves the transfer function of the original model while presenting significant accuracy benefits for our purposes. Specifically, if $\mathbf{F}$ is the Cholesky factor of $\mathbf{G}_i$ ($\mathbf{G}_i = \mathbf{F}\mathbf{F}^T$),

then by applying the congruence transformation with matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{I}_{p+q} & \mathbf{0} \\ -\mathbf{G}_i^{-1}\mathbf{G}_c & \mathbf{F}^{-T} \end{bmatrix}$$

on $\mathbf{G}_n$, $\mathbf{C}_n$, $\mathbf{\Gamma}$ we obtain:

$$\begin{aligned} \mathbf{G}_n^{'} = \mathbf{S}^T\mathbf{G}_n\mathbf{S} &= \begin{bmatrix} \mathbf{G}_p - \mathbf{G}_c^T\mathbf{G}_i^{-1}\mathbf{G}_c & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-(p+q)} \end{bmatrix} \\ &\equiv \begin{bmatrix} \mathbf{G}_p^{'} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-(p+q)} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \mathbf{C}_n^{'} = \mathbf{S}^T\mathbf{C}_n\mathbf{S} &= \begin{bmatrix} \mathbf{C}_p - \mathbf{R}_c^T\mathbf{G}_i^{-1}\mathbf{G}_c - \mathbf{G}_c^T\mathbf{G}_i^{-T}\mathbf{C}_c & \mathbf{R}_c^T\mathbf{F}^{-T} \\ \mathbf{F}^{-1}\mathbf{R}_c & \mathbf{F}^{-1}\mathbf{C}_i\mathbf{F}^{-T} \end{bmatrix} \\ &\equiv \begin{bmatrix} \mathbf{C}_p^{'} & \mathbf{C}_c^{'T} \\ \mathbf{C}_c^{'} & \mathbf{C}_i^{'} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \mathbf{\Gamma}^{'} = \mathbf{S}^T\mathbf{\Gamma}\mathbf{S} &= \begin{bmatrix} \mathbf{\Gamma}_p - \mathbf{R}_\gamma^T\mathbf{G}_i^{-1}\mathbf{G}_c - \mathbf{G}_c^T\mathbf{G}_i^{-T}\mathbf{\Gamma}_c & \mathbf{R}_\gamma^T\mathbf{F}^{-T} \\ \mathbf{F}^{-1}\mathbf{R}_\gamma & \mathbf{F}^{-1}\mathbf{\Gamma}_i\mathbf{F}^{-T} \end{bmatrix} \\ &\equiv \begin{bmatrix} \mathbf{\Gamma}_p^{'} & \mathbf{\Gamma}_c^{'T} \\ \mathbf{\Gamma}_c^{'} & \mathbf{\Gamma}_i^{'} \end{bmatrix} \end{aligned} \tag{10}$$

where $\mathbf{R}_c \equiv \mathbf{C}_c - \mathbf{C}_i\mathbf{G}_i^{-1}\mathbf{G}_c$ and $\mathbf{R}_\gamma \equiv \mathbf{\Gamma}_c - \mathbf{\Gamma}_i\mathbf{G}_i^{-1}\mathbf{G}_c$. By further applying (9) on (10) we obtain the following reduced-order transformed matrices:

$$\begin{aligned} \widehat{\mathbf{G}}_n^{'} &= \mathbf{U}^T\mathbf{G}_n^{'}\mathbf{V} = \begin{bmatrix} \mathbf{G}_p^{'} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix} \\ \widehat{\mathbf{C}}_n^{'} &= \mathbf{U}^T\mathbf{C}_n^{'}\mathbf{V} = \begin{bmatrix} \mathbf{C}_p^{'} & \mathbf{C}_c^{'T}\mathbf{V}_i \\ \mathbf{U}_i^T\mathbf{C}_c^{'} & \mathbf{U}_i^T\mathbf{C}_i^{'}\mathbf{V}_i \end{bmatrix} \\ \widehat{\mathbf{\Gamma}}^{'} &= \mathbf{U}^T\mathbf{\Gamma}_n^{'}\mathbf{V} = \begin{bmatrix} \mathbf{\Gamma}_p^{'} & \mathbf{\Gamma}_c^{'T}\mathbf{V}_i \\ \mathbf{U}_i^T\mathbf{\Gamma}_c^{'} & \mathbf{U}_i^T\mathbf{\Gamma}_i^{'}\mathbf{V}_i \end{bmatrix} \end{aligned} \tag{11}$$

The important observation in our case is that $\widehat{\mathbf{G}}_n^{'}$ has only the $(p+q)\times(p+q)$ dense submatrix $\mathbf{G}_p^{'} = \mathbf{G}_p - \mathbf{G}_c^T\mathbf{G}_i^{-1}\mathbf{G}_c$, which is also equal to the Schur complement of $\mathbf{G}_i$ in the original M-matrix $\mathbf{G}_n$. A result proven in [23] states that the Schur complement of a block of an M-matrix is also M-matrix, and thus retains all associated properties and specifically the diagonal dominance. Since the subsequent approximation of $\widehat{\mathbf{G}}_n^{'}$, $\widehat{\mathbf{C}}_n^{'}$, $\widehat{\mathbf{\Gamma}}^{'}$ with their nearest SDD matrices does entail a certain error, this can be avoided for the matrix $\widehat{\mathbf{G}}_n^{'}$ (which is commonly the most important of the three matrices), leading to a significant improvement in accuracy w.r.t. the non-transformed case, where all the $\widehat{\mathbf{G}}_n$, $\widehat{\mathbf{C}}_n$, $\widehat{\mathbf{\Gamma}}$ of (6) would need to be approximated by nearest SDD matrices.

C. Conversion of a general SDD matrix to a circuit-type M-matrix

The back projection procedure to the nearest SDD matrix generally leads to off-diagonal elements with both positive and negative signs. However, in order to exploit the sparsification technique of the next section all off-diagonal elements must be non-positive, so that the SDD matrix is actually a circuit-type M-matrix and is related to a graph with positive weights. In this section we show that, for any given SDD matrix, it is possible to construct an equivalent SDD matrix with nonpositive off-diagonal elements and twice the size. Specifically, let a general matrix $\mathbf{A}$ be decomposed into the sum $\mathbf{A} = \mathbf{A}_1 + \mathbf{A}_2$. Then the linear systems $\mathbf{Ax} = \mathbf{b}$ and $-\mathbf{Ax} = -\mathbf{b}$ can be written as:

$$\begin{aligned} \mathbf{A}_1\mathbf{x} + \mathbf{A}_2\mathbf{x} = \mathbf{b} &\Rightarrow \mathbf{A}_1\mathbf{x} - \mathbf{A}_2(-\mathbf{x}) = \mathbf{b} \\ -\mathbf{A}_1\mathbf{x} - \mathbf{A}_2\mathbf{x} = -\mathbf{b} &\Rightarrow -\mathbf{A}_2\mathbf{x} + \mathbf{A}_1(-\mathbf{x}) = -\mathbf{b} \end{aligned}$$

Now if $\mathbf{A}$ is SDD, then it can be decomposed into the sum of $\mathbf{A}_1 = \mathbf{A}_{dn}$ and $\mathbf{A}_2 = \mathbf{A}_p$, where $\mathbf{A}_{dn}$ contains the diagonal elements and the negative off-diagonal elements while $\mathbf{A}_p$ contains the positive off-diagonal elements (with zeros everywhere else), and the previous systems become

$$\begin{aligned} \mathbf{A}_{dn}\mathbf{x} - \mathbf{A}_p(-\mathbf{x}) = \mathbf{b} \\ -\mathbf{A}_p\mathbf{x} + \mathbf{A}_{dn}(-\mathbf{x}) = -\mathbf{b} \end{aligned}$$

or the augmented system:

$$\begin{bmatrix} \mathbf{A}_{dn} & -\mathbf{A}_p \\ -\mathbf{A}_p & \mathbf{A}_{dn} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ -\mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{b} \end{bmatrix} \tag{12}$$

whose matrix is SDD with nonpositive off-diagonal elements. Thus, for all simulation purposes, an SDD matrix can be substituted by a circuit-type M-matrix with equivalent solution and twice the size (which is hardly an issue for reduced order models whose dimension is at most a few thousand and where matrix density is the major problem).

IV. Sparsification of Circuit-type M-matrices

The matrices resulting from the nearest projection and conversion procedures of the previous section are dense circuit-type M-matrices, and thus they can be decomposed into the sum of a nonnegative diagonal matrix and a dense Laplacian matrix. In this section we focus on sparsifying those dense Laplacian matrices. Because a Laplacian matrix $\mathbf{L}_G$ has a direct correspondence to a weighed graph, by considering every non-zero off-diagonal element l_{ij}^G $(i \neq j)$ as an edge (i,j) between vertices i and j with weight $w(i,j) = -l_{ij}^G$, we can cast the problem of sparsifying $\mathbf{L}_G$ to sparsifying its corresponding graph. As there are many notions of graph sparsification in the literature (one, for instance, can naively truncate edges with small weights), the appropriate one for graphs whose Laplacian matrices $\mathbf{L}_G$ appear in a dynamical system like (5) is the *spectral* sparsification [24] which aims at preserving the eigenvalues λ_i^G of $\mathbf{L}_G$ (since these determine the effect of the matrices on the impulse response, or the transfer function of the system). An efficient spectral graph sparsification method which, for a given dense graph $G = (V, E, w)$ with $O(n^2)$ edges (where n is the number of vertices) constructs a sparse subgraph $H = (V, E^H, w^H)$ of G with $O(n\log n)$ edges, in a way that the spectrum of G is preserved in H to the largest extent possible, is presented in [25]. The proposed algorithm is based on the concept of "effective resistance" of an edge (i,j) which is defined as follows:

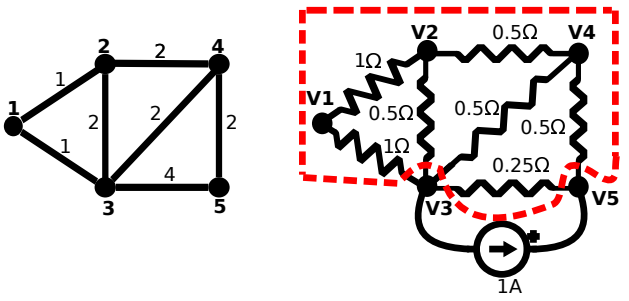

Fig. 1: Electrical circuit analogy of graph, where each edge has resistance the inverse of its weight. The "effective resistance" of edge $(3,5)$ is the voltage drop across the vertices 3 and 5 when we apply a current source of 1A between them.

$$R^{eff}_{(i,j)} = (\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{L}_G^+ (\mathbf{e}_i - \mathbf{e}_j) \tag{13}$$

where $\mathbf{L}_G^+$ is the pseudoinverse of $\mathbf{L}_G$ and $\mathbf{e}_i$, $\mathbf{e}_j$ are elementary unit vectors with 1 at positions i and j respectively. In electrical network analogy the effective resistance of edge (i,j) represents the voltage drop across (i,j) when we apply a unit current source between its endpoint vertices i and j (see Fig. 1).

It is proven in [25] that, for a given $\epsilon \in [\frac{1}{\sqrt{n}}, 1]$ representing a tradeoff between sparsity and accuracy, a random sampling of the edges of G in $q = \frac{9C^2 n \log n}{\epsilon^2}$ trials with probability mass function (PMF) $p(i,j) = \frac{w(i,j)R^{eff}_{(i,j)}}{n-1}$ to include in H, will preserve the eigenvalues of Laplacian $\mathbf{L}_G$ in Laplacian $\mathbf{L}_H$ as:

$$(1-\epsilon)\lambda_i^G \leq \lambda_i^H \leq (1+\epsilon)\lambda_i^G, \quad \forall i = 1, \ldots, n$$

The above is essentially a probabilistic and asymptotic result, in that it holds with a high enough probability and for sufficiently large n. Also the constant C must be chosen to satisfy a specific intrinsic inequality, and through experimentation we have settled to a value[2] $C = 0.1$.

Below, we have modified and improved the algorithm of [25] (Algorithm 4) in two ways. First, since the random sampling of edges might result in a disjoint sparse graph H, we provide the option to initialize H by the "maximum likelihood" spanning tree (MLST) which includes the $(n$-1) most probable edges that guarantee a connected graph. Second and more importantly, instead of adding edges to the sparse graph by random trials (which would require significant memory and computational effort), we calculate first the expected times an edge would be selected if random sampling of q trials were performed, via multiplying q with the probability $p(i,j)$ and rounding to the nearest integer. Then we add to H those edges whose expected times to be selected are greater than 0. By doing so, we also remove the randomness from the algorithm of [25], and thus the proposed sparsification always results in the same sparse graph.

[2]The inequality to be satisfied is $C\sqrt{(\log q)(n-1)/q} < 1/2$ (Lemma 5 of [25]), and $C = 0.1$ makes a valid choice for every possible n and $\epsilon \in [\frac{1}{\sqrt{n}}, 1]$.

Algorithm 4 For a dense graph $G = (V, E, w)$ with Laplacian matrix $\mathbf{L}_G$, construct a sparse subgraph $H = (V, E^H, w^H)$ with Laplacian matrix $\mathbf{L}_H$

function $\mathbf{L}_H = \text{SPARLAP}(\mathbf{L}_G, \epsilon, s)$
  $n = |V|$; $C = 0.1$
  $q = \frac{9C^2 n \log n}{\epsilon^2}$
  $\mathbf{L}_H = \mathbf{0}$; $E^H = \{\}$
  for $(i,j) \in E$ **do**
    Calculate effective resistance $R^{eff}_{(i,j)}$ from (13)
    $p(i,j) = \frac{w(i,j)R^{eff}_{(i,j)}}{n-1}$
  end for
  if $s == 1$ **then**
    $H = MST(G')$ where $G' = (V, E, p)$ and *MST*: Max Spanning Tree
    for $(i,j) \in E^H$ **do**
      $l^H_{ij} = l^G_{ij}$; $l^H_{ji} = l^G_{ji}$
    end for
    $q = q - (n-1)$
  end if
  for $(i,j) \in (E - E^H)$ **do**
    if $round(p(i,j) * q) > 0$ **then**
      $l^H_{ij} = l^G_{ij}$; $l^H_{ji} = l^G_{ji}$
    end if
  end for
end function

The most expensive operation in Algorithm 4 is the computation of effective resistances through the pseudoinverse $\mathbf{L}_G^+$ of $\mathbf{L}_G$, which entails solving n linear systems in $\mathbf{L}_G$. To reduce the computational cost, a procedure has also been given in [25] that approximately calculates effective resistance within certain (probabilistic) bounds and which asymptotically requires solution of $O(\log n)$ linear systems. It must also be noted that a similar graph sparsification approach was proposed recently in [26], which is based on perturbation analysis of the eigenvalues of $\mathbf{L}_G$, rather than computation of effective resistances, and may provide computational benefits in certain contexts (although more experiments are needed to verify its effectiveness across a wide range of problems).

V. Proposed Methodology for Sparsification of Dense MOR Models

Combining all the algorithms presented in the previous sections, the complete methodology for sparsifying dense MOR models is given in Algorithm 5. Note that especially for the sparse conductance matrix $\widehat{\mathbf{G}}_{sp}$ it is advisable to start from the MLST (i.e. call the *SparLap* algorithm with option $s = 1$), both because the original circuit typically has a connected conductance graph and as to prevent the sparsified reduced-order circuit from being disjoint. Also, the sparsity-accuracy tradeoff ϵ is usually set at the smallest possible value $\epsilon = \frac{1}{\sqrt{2r}}$ (where r is the order of the reduced model), since this provides the most accurate approximation while it is shown experimentally to give sparsities near or over 90% in all cases.

Algorithm 5 Sparsification of dense models resulting from MOR

function $\widehat{\mathbf{G}}_{sp}, \widehat{\mathbf{C}}_{sp}, \widehat{\mathbf{\Gamma}}_{sp} = \text{MORSPARSE}(\widehat{\mathbf{G}}_n, \widehat{\mathbf{C}}_n, \widehat{\mathbf{\Gamma}}, \epsilon)$
  $\widehat{\mathbf{G}}_{dd}$ = $prjSDD(\widehat{\mathbf{G}}_n$, 1e-6)
  $\widehat{\mathbf{C}}_{dd}$ = $prjSDD(\widehat{\mathbf{C}}_n$, 1e-6)
  $\widehat{\mathbf{\Gamma}}_{dd}$ = $prjSDD(\widehat{\mathbf{\Gamma}}$, 1e-6)
  Apply conversion (12) to $\widehat{\mathbf{G}}_{dd}$, $\widehat{\mathbf{C}}_{dd}$, $\widehat{\mathbf{\Gamma}}_{dd} \in \Re^{r\times r}$, to obtain the matrices $\widehat{\mathbf{G}}_M$, $\widehat{\mathbf{C}}_M$, $\widehat{\mathbf{\Gamma}}_M \in \Re^{2r\times 2r}$ respectively
  Extract Laplacian part of $\widehat{\mathbf{G}}_M$, $\widehat{\mathbf{C}}_M$, $\widehat{\mathbf{\Gamma}}_M$ in matrices $\mathbf{L}_g$, $\mathbf{L}_c$, $\mathbf{L}_\gamma$, and store diagonal remainder of strictly diagonally dominant rows in matrices $\mathbf{D}_g$, $\mathbf{D}_c$, $\mathbf{D}_\gamma$
  $\widehat{\mathbf{G}}_{sp}$ = $SparLap(\mathbf{L}_g, \epsilon, 1)$ + $\mathbf{D}_g$
  $\widehat{\mathbf{C}}_{sp}$ = $SparLap(\mathbf{L}_c, \epsilon, 0)$ + $\mathbf{D}_c$
  $\widehat{\mathbf{\Gamma}}_{sp}$ = $SparLap(\mathbf{L}_\gamma, \epsilon, 0)$ + $\mathbf{D}_\gamma$
end function

VI. Synthesis of Sparse MOR Models

Because the sparsified output matrices $\widehat{\mathbf{G}}_{sp}$, $\widehat{\mathbf{C}}_{sp}$, $\widehat{\mathbf{\Gamma}}_{sp}$ of Algorithm 5 have resulted from the conversion (12), they have the form:

$$\widehat{\mathbf{G}}_{sp} = \begin{bmatrix} \widehat{\mathbf{G}}_{dn} & -\widehat{\mathbf{G}}_p \\ -\widehat{\mathbf{G}}_p & \widehat{\mathbf{G}}_{dn} \end{bmatrix}, \widehat{\mathbf{C}}_{sp} = \begin{bmatrix} \widehat{\mathbf{C}}_{dn} & -\widehat{\mathbf{C}}_p \\ -\widehat{\mathbf{C}}_p & \widehat{\mathbf{C}}_{dn} \end{bmatrix}, \widehat{\mathbf{\Gamma}}_{sp} = \begin{bmatrix} \widehat{\mathbf{\Gamma}}_{dn} & -\widehat{\mathbf{\Gamma}}_p \\ -\widehat{\mathbf{\Gamma}}_p & \widehat{\mathbf{\Gamma}}_{dn} \end{bmatrix} \tag{14}$$

where $\widehat{\mathbf{G}}_{dd} = \widehat{\mathbf{G}}_{dn} + \widehat{\mathbf{G}}_p$, $\widehat{\mathbf{C}}_{dd} = \widehat{\mathbf{C}}_{dn} + \widehat{\mathbf{C}}_p$, $\widehat{\mathbf{\Gamma}}_{dd} = \widehat{\mathbf{\Gamma}}_{dn} + \widehat{\mathbf{\Gamma}}_p$ are the decomposition of the nearest SDD approximations of the reduced-order matrices $\widehat{\mathbf{G}}_n$, $\widehat{\mathbf{C}}_n$, $\widehat{\mathbf{\Gamma}} \in \Re^{r\times r}$ into the sums of $\widehat{\mathbf{G}}_{dn}$, $\widehat{\mathbf{C}}_{dn}$, $\widehat{\mathbf{\Gamma}}_{dn}$ (diagonal and - eventually sparsified - negative off-diagonal elements) and $\widehat{\mathbf{G}}_p$, $\widehat{\mathbf{C}}_p$, $\widehat{\mathbf{\Gamma}}_p$ (eventually sparsified positive off-diagonal elements). Thus, the reduced second-order model after MOR and the application of the procedure encompassed in Algorithm 5 will be:

$$\begin{bmatrix} \widehat{\mathbf{C}}_{dn} & -\widehat{\mathbf{C}}_p \\ -\widehat{\mathbf{C}}_p & \widehat{\mathbf{C}}_{dn} \end{bmatrix} \begin{bmatrix} \frac{d\widehat{\mathbf{v}}(t)}{dt} \\ \frac{-d\widehat{\mathbf{v}}(t)}{dt} \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{G}}_{dn} & -\widehat{\mathbf{G}}_p \\ -\widehat{\mathbf{G}}_p & \widehat{\mathbf{G}}_{dn} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{v}}(t) \\ -\widehat{\mathbf{v}}(t) \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{\Gamma}}_{dn} & -\widehat{\mathbf{\Gamma}}_p \\ -\widehat{\mathbf{\Gamma}}_p & \widehat{\mathbf{\Gamma}}_{dn} \end{bmatrix} \begin{bmatrix} \int_0^t \widehat{\mathbf{v}}(t)dt \\ -\int_0^t \widehat{\mathbf{v}}(t)dt \end{bmatrix} = \begin{bmatrix} \mathbf{U}^T\mathbf{B} \\ -\mathbf{U}^T\mathbf{B} \end{bmatrix} \mathbf{u}(t)$$

$$\mathbf{y}(t) = \mathbf{E}\mathbf{V}\widehat{\mathbf{v}}(t) \tag{15}$$

with $\mathbf{U}$, $\mathbf{V} \in \Re^{n\times r}$ being the MOR projectors to the r-dimensional subspace. Note here the important fact that the two copies of each kind of block in the compound matrices remain identical after applying Algorithm 4 (a proof of this is provided in Appendix A), so that (12) and (15) still hold after sparsification.

Since the matrices $\widehat{\mathbf{G}}_{sp}$, $\widehat{\mathbf{C}}_{sp}$, $\widehat{\mathbf{\Gamma}}_{sp}$ of (14) can be written as the sum of a Laplacian and a diagonal matrix, the proposed procedure has the additional benefit that the sparsified reduced-order model (15) can be readily synthesized into an RLC circuit with only positive elements. Such a synthesized circuit will consist of two identical network parts corresponding to the matrix blocks $\widehat{\mathbf{G}}_{dn}$, $\widehat{\mathbf{C}}_{dn}$, $\widehat{\mathbf{\Gamma}}_{dn}$, interconnected by a network part corresponding to the blocks $\widehat{\mathbf{G}}_p$, $\widehat{\mathbf{C}}_p$, $\widehat{\mathbf{\Gamma}}_p$, and having current sources $\mathbf{U}^T\mathbf{B}\mathbf{u}(t)$ injected to all nodes of one of the copies (where the node voltages $\widehat{\mathbf{v}}(t)$ are received from) while leaving the other copy. For each network part, a non-zero matrix element g_{ij} or c_{ij} or γ_{ij} corresponds to either a resistance $\frac{-1}{g_{ij}}$ or a capacitance $-c_{ij}$ or an inductance $\frac{-1}{\gamma_{ij}}$ between nodes i and j, while for the strictly diagonally dominant rows i there is a resistance $\frac{1}{g_{ii}+\sum_{j\neq i} g_{ij}}$ or a capacitance $c_{ii} + \sum_{j\neq i} c_{ij}$ or an inductance $\frac{1}{\gamma_{ii}+\sum_{j\neq i} \gamma_{ij}}$ from node i to ground [21]. The synthesis configuration is shown schematically in Fig. 2.

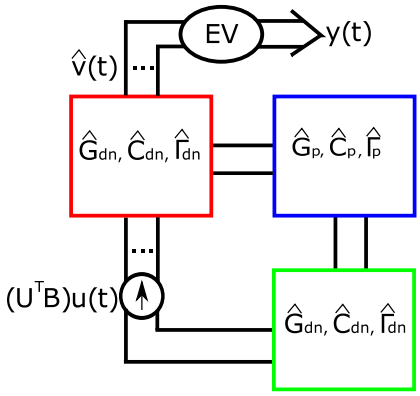

Fig. 2: Structure of synthesized sparse MOR models.

As a synthesis example, let the following 4×4 matrices be the outputs of the projections of the MOR matrices $\widehat{\mathbf{G}}_n$, $\widehat{\mathbf{C}}_n$, $\widehat{\mathbf{\Gamma}}$ to the nearest SDD matrices (where X's correspond to elements that will be eventually eliminated by the sparsification process of Algorithm 4):

$$\widehat{\mathbf{G}}_{dd} = \begin{bmatrix} 0.92 & 0.11 & -0.71 & X \\ 0.11 & 1 & -0.3 & 0.59 \\ -0.71 & -0.3 & 1.21 & X \\ X & 0.59 & X & 1.59 \end{bmatrix}$$

$$\widehat{\mathbf{C}}_{dd} = \begin{bmatrix} 1 & 0.72 & X & X \\ 0.72 & 1.63 & X & -0.91 \\ X & X & 1 & X \\ X & -0.91 & X & 1 \end{bmatrix}$$

$$\widehat{\mathbf{\Gamma}}_{dd} = \begin{bmatrix} 0.12 & X & X & 0.12 \\ X & 0.3 & X & -0.2 \\ X & X & 0.5 & X \\ 0.12 & -0.2 & X & 0.32 \end{bmatrix}$$

By applying conversion (12) to the above matrices, and after sparsification by *SparLap*, we obtain:

$$\widehat{\mathbf{G}}_{sp} = \begin{bmatrix} 0.92 & 0 & -0.71 & 0 & 0 & -0.11 & 0 & 0 \\ 0 & 1 & -0.3 & 0 & -0.11 & 0 & 0 & -0.59 \\ -0.71 & -0.3 & 1.21 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.59 & 0 & -0.59 & 0 & 0 \\ 0 & -0.11 & 0 & 0 & 0.92 & 0 & -0.71 & 0 \\ -0.11 & 0 & 0 & -0.59 & 0 & 1 & -0.3 & 0 \\ 0 & 0 & 0 & 0 & -0.71 & -0.3 & 1.21 & 0 \\ 0 & -0.59 & 0 & 0 & 0 & 0 & 0 & 1.59 \end{bmatrix}$$

$$\widehat{\mathbf{C}}_{sp} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -0.72 & 0 & 0 \\ 0 & 1.63 & 0 & -0.91 & -0.72 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.91 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -0.72 & 0 & 0 & 1 & 0 & 0 & 0 \\ -0.72 & 0 & 0 & 0 & 0 & 1.63 & 0 & -0.91 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.91 & 0 & 1 \end{bmatrix}$$

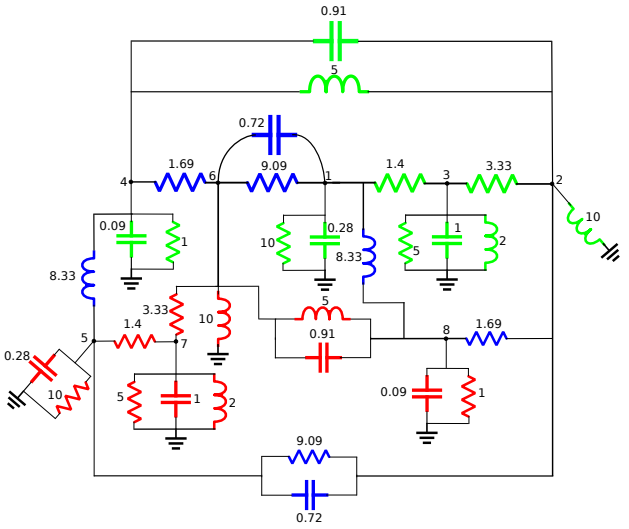

Fig. 3: Synthesized circuit of the example matrices $\widehat{\mathbf{G}}_{sp}$, $\widehat{\mathbf{C}}_{sp}$, $\widehat{\mathbf{\Gamma}}_{sp}$. The electrical elements are resistances, capacitances and inductances, with colors corresponding to blocks of Fig.2, while the nodes are numbered according to rows/columns of the matrices.

$$\widehat{\mathbf{\Gamma}}_{sp} = \begin{bmatrix} 0.12 & 0 & 0 & 0 & 0 & 0 & 0 & -0.12 \\ 0 & 0.3 & 0 & -0.2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & -0.2 & 0 & 0.32 & -0.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.12 & 0.12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & -0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 \\ -0.12 & 0 & 0 & 0 & 0 & -0.2 & 0 & 0.32 \end{bmatrix}$$

The circuit representation corresponding to Fig. 2 for these matrices is shown in Fig. 3 (where the RLC elements have been derived as described previously). Since the sparsified reduced-order model (15) can be synthesized by an RLC circuit with positive elements, it also constitutes a passive model (under suitable assumptions on the matrices $\mathbf{U}^T\mathbf{B}$ and $\mathbf{EV}$, which should hold if the reduced-order model represented by the matrices (6) is itself passive - e.g. sufficient conditions are $\mathbf{U} = \mathbf{V}$ and $\mathbf{E} = \mathbf{B}^T$ [1]). The synthesized circuit can be considered in a sense (that of matrix proximity) the closest passive circuit to the reduced-order model represented by (6). If synthesis to a passive circuit is not needed, we can even convert (15) to a sparsified reduced-order model of size r (whose matrices are SDD but not M-matrices) by taking the first row of (15) as:

$$\begin{aligned}(\widehat{\mathbf{C}}_{dn} + \widehat{\mathbf{C}}_p)\frac{d\widehat{\mathbf{v}}(t)}{dt} + (\widehat{\mathbf{G}}_{dn} + \widehat{\mathbf{G}}_p)\widehat{\mathbf{v}}(t) \\ + (\widehat{\mathbf{\Gamma}}_{dn} + \widehat{\mathbf{\Gamma}}_p)\int_0^t \widehat{\mathbf{v}}(t)dt = \mathbf{U}^T\mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{EV}\widehat{\mathbf{v}}(t)\end{aligned}$$

where $\widehat{\mathbf{G}}_{dn}$, $\widehat{\mathbf{G}}_p$, $\widehat{\mathbf{C}}_{dn}$, $\widehat{\mathbf{C}}_p$, $\widehat{\mathbf{\Gamma}}_{dn}$, $\widehat{\mathbf{\Gamma}}_p$ are the sparsified blocks of matrices (14).

VII. Experimental Evaluation

For the experimental evaluation of MORSparse presented in Algorithm 5 we consider a set of synthetic RLC interconnects as well as a set of RC netlists extracted from real designs. The RLC benchmarks consist of multiple layers of parallel bus lines, which are segmented lengthwise into RL branches and extracted by FastHenry [27] (with average resistance and inductance values of 3.3156Ω and $14.5pH$ respectively). These are connected together in y and z dimensions by resistors of value 3Ω, while capacitors of value $20fF$ are connected to ground at all intersection points. The RC benchmarks are the designs presented in [5], and include a Low Noise Amplifier (LNA), an RF mixer (MX3) and an RC interconnect (RCintc). We provide in Table I all the structural details of each benchmark.

TABLE I: Structural details of benchmark circuits.

Bench.	Branches	Ports	Nodes
interc1	256	32	544
interc2	768	96	1632
interc3	2048	256	4352
interc4	4096	512	8704
LNA	NA	79	29885
MX3	NA	110	867
RCintc	NA	646	16862

In all experiments we constructed the original model matrices, and then separated port from internal nodes and performed congruence transformation (10) before applying MOR on the internal node matrices (as described in Section III-B). To obtain the reduced-order models (ROMs) we employed the established MOR methods of SAPOR (expanded around the point 1e+9) for the RLC benchmarks and PACT for the RC benchmarks. Note here that while PACT is not a projection-based MOR method, it was the one that initially introduced the transformation (10) and naturally produces matrices in the required form, but we could have chosen any projection-based method instead (e.g. [1]). After the MOR process we executed our proposed MORSparse method of Algorithm 5 to obtain the sparsified ROMs, and also performed comparisons with SparceRC [5] for the RC benchmarks. The execution of MORSparse was done in all cases with the accuracy-sparsity tradeoff value at $\epsilon = \frac{1}{\sqrt{2r}}$ (where r is the ROM order), which provides the best accuracy while still leading to substantial degrees of sparsity. We implemented the whole procedure in MATLAB 2017a and ran all our experiments on a system equipped with a 3.60GHz Intel Core i7 CPU and 16GB memory.

To assess the two main approximations of MORSparse, namely the projection of ROMs to the nearest SDD and the sparsification of the dense M-matrices, we report in Table II the relative Frobenius distance between the reduced and the SDD-projected matrices (only for $\widehat{\mathbf{C}}_n$ and $\widehat{\mathbf{\Gamma}}$, as $||\widehat{\mathbf{G}}_{dd} - \widehat{\mathbf{G}}_n^{'}||_F = 0$ from (11)), as well as the maximum eigenvalues (which coincide with the spectral norms) and the relative spectral distance between the dense and sparse M-matrices. It can be perceived that the spectral distances and max eigenvalue approximations are excellent (in fact all eigenvalues can be shown to be preserved) while the

Frobenius distances are generally good (with the exception of RCintc, which nevertheless is not reflected as discrepancy in the worst-case waveforms shown later). In Table III we report the ROM order for each benchmark and compare the sparsity ratios ($\frac{\#zeros}{\#elements}$) of the SAPOR/PACT ROMs with the ROMs after MORSparse as well as SparseRC. In all cases MORSparse significantly increased sparsity (especially for the $\widehat{\mathbf{C}}_n$ and $\widehat{\mathbf{\Gamma}}$ matrices) and in most cases it achieved ratios of over 99%. Note that the sparsity of the conductance matrix was already high for the SAPOR/PACT ROMs, due to the form of $\widehat{\mathbf{G}}_n^{'}$ in (11) with only one dense block after the congruence transformation (10). Still MORSparse increased sparsity even for the $\widehat{\mathbf{G}}_n^{'}$ matrix, which can be quite significant for potentially larger ROMs. Note also that SparseRC cannot handle RLC circuits and cannot synthesize the sparse ROMs into an equivalent circuit with only positive elements.

Finally, to evaluate the sparse ROMs created by MORSparse in terms of simulation accuracy and runtime, we formed an experimental setup where we applied a 1A step current source at all ports of every benchmark and executed simulation for 10^6 timepoints. Table IV reports the time (*Sparsif. time*) for the whole execution of MORSparse, including the nearest SDD projection (Algorithm 1) and sparsification (Algorithm 4), the simulation time of the initial SAPOR/PACT ROMs and sparsified ROMs after MORSparse, as well as the maximum error of MORSparse over all ports and waveform points. Fig. 4 compares the voltage responses of SAPOR and MORSparse ROMs for RLC benchmarks interc1, interc3 at the ports exhibiting the maximum waveform-wise root-mean-square (rms) error. Fig. 5 compares the voltage responses of PACT, MORSparse and SparseRC ROMs for RC benchmarks MX3, LNA, RCintc, and Fig. 6 compares the frequency response of the same ROMs for benchmark MX3, all at the ports with the maximum waveform-wise rms error. In all cases the MORSparse waveforms exhibit excellent match to those of SAPOR/PACT and SparseRC, and especially for the worst port waveforms shown in Fig. 5 the maximum errors of MORSparse were 1.6e-3V, 1.29e-5V, and 3.21e-9V for MX3, LNA, and RCintc respectively. Compared also to the experimental figures in [10] and [11], where the sparse ROMs were produced without the congruence transformation (10) and whose voltage responses may come close to the dense ROMs but exhibit a discernible deviation, we can conclude that there is a definite improvement in accuracy when we avoid the nearest SDD approximation of $\widehat{\mathbf{G}}_n^{'}$ by using the transformation (10). On the other hand, MORSparse achieves much higher levels of sparsity than both SAPOR/PACT and SparseRC, which lead to speedups up to $\times$4.1 for the tested benchmarks, while these are expected to grow even larger for the typical ROMs in the order of a few thousand encountered in practice. The time to construct the sparse ROMs with MORSparse is, of course, an additional overhead, but this needs to be borne only once for every circuit model, and will be eclipsed by the gains to be had in simulation of the sparse versus the dense ROM over multiple simulations and numerous (sometimes millions) timepoints per simulation session.

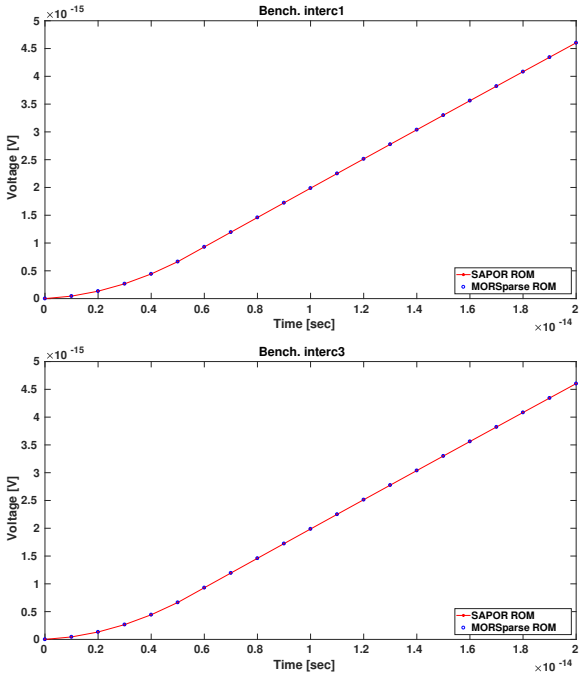

Fig. 4: Voltage response of SAPOR and MORSparse ROMs at the port with the worst-case rms error for benchmarks interc1 and interc3.

TABLE IV: Runtime and accuracy results for simulation of SAPOR/PACT and MORSparse ROMs.

Bench.	Sparsif. time (s)	ROM sim. time (s)	MORSparse sim. time (s)	Speedup	Max error (V)
interc1	0.15	20	14	$\times$1.4	8.72e-19
interc2	0.73	91	22	$\times$4.1	8.89e-19
interc3	4.8	397	143	$\times$2.8	8.83e-19
interc4	50	1372	546	$\times$2.5	8.92e-19
LNA	0.32	31	24	$\times$1.3	4e-4
MX3	0.41	53	32	$\times$1.65	5e-3
RCintc	4.5	1290	319	$\times$4	2.7e-3

VIII. Conclusion

In this paper we presented a rigorous mathematical approach for the sparsification of dense matrices in MOR of RC/RLC circuits. We derive a second-order formulation of the original model, so that the resulting MOR matrices are close to DD matrices, and then exploit a rigorous sparsification methodology, which entails the computation of the nearest Laplacian matrices of the reduced model matrices under the Frobenius norm and a graph sparsification algorithm, to sparsify these matrices. We close by demonstrating that the sparsified reduced models can be synthesized straightforwardly into an RLC circuit with only positive elements. Experimental results indicated that high sparsity ratios can be obtained while a small error is introduced.

TABLE II: Relative Frobenius distance between the reduced-order and the SDD-projected matrices, and spectral norm (max eigenvalue) and relative spectral distance of dense and sparse reduced M-matrices

Bench.	SDD Frobenius distance ($\|\|\cdot\|\|_F$)		Max dense & sparse M-matrix eigenvalue ($\|\|\cdot\|\|_2$)						M-matrix spectral distance ($\|\|\cdot\|\|_2$)		
	$\frac{\|\|\widehat{\mathbf{C}}_{dd}-\widehat{\mathbf{C}}_n^{'}\|\|}{\|\|\widehat{\mathbf{C}}_n^{'}\|\|}$	$\frac{\|\|\widehat{\mathbf{\Gamma}}_{dd}-\widehat{\mathbf{\Gamma}}^{'}\|\|}{\|\|\widehat{\mathbf{\Gamma}}^{'}\|\|}$	$\|\|\widehat{\mathbf{G}}_M\|\|$	$\|\|\widehat{\mathbf{G}}_{sp}\|\|$	$\|\|\widehat{\mathbf{C}}_M\|\|$	$\|\|\widehat{\mathbf{C}}_{sp}\|\|$	$\|\|\widehat{\mathbf{\Gamma}}_M\|\|$	$\|\|\widehat{\mathbf{\Gamma}}_{sp}\|\|$	$\frac{\|\|\widehat{\mathbf{G}}_M-\widehat{\mathbf{G}}_{sp}\|\|}{\|\|\widehat{\mathbf{G}}_M\|\|}$	$\frac{\|\|\widehat{\mathbf{C}}_M-\widehat{\mathbf{C}}_{sp}\|\|}{\|\|\widehat{\mathbf{C}}_M\|\|}$	$\frac{\|\|\widehat{\mathbf{\Gamma}}_M-\widehat{\mathbf{\Gamma}}_{sp}\|\|}{\|\|\widehat{\mathbf{\Gamma}}_M\|\|}$
interc1	0.02	0.02	1.23	1.32	2.25e-14	2.25e-14	1.824e11	1.829e11	0.07	3e-4	5.45e-4
interc2	0.05	0.05	2.14	2.18	2.25e-14	2.25e-14	1.8e11	1.8e11	0.026	1e-3	1e-3
interc3	0.02	0.04	1.84	1.85	2.25e-14	2.25e-14	1.8e11	1.8e11	1e-3	5.3e-4	2.9e-5
interc4	0.01	0.04	2.268	2.274	2.25e-14	2.25e-14	1.81e11	1.81e11	7.4e-4	2.1e-4	1e-4
LNA	0.029	NA	640	689	2.9e-11	2.89e-11	NA	NA	0.08	2e-4	NA
MX3	0.046	NA	193	191.8	3.14e-12	3.14e-12	NA	NA	0.05	1e-3	NA
RCint	0.54	NA	1.77e3	1.76e3	7.7e-13	7.7e-13	NA	NA	3e-3	5.1e-4	NA

TABLE III: Comparison of sparsity ratios between SAPOR/PACT, MORSparse, and SparceRC ROMs.

Bench.	ROM order	ROM Sparsity			ROM Sparsity after MORSparse			SparseRC Sparsity		
		$\widehat{\mathbf{G}}_n^{'}$	$\widehat{\mathbf{C}}_n^{'}$	$\widehat{\mathbf{\Gamma}}^{'}$	$\widehat{\mathbf{G}}_{sp}$	$\widehat{\mathbf{C}}_{sp}$	$\widehat{\mathbf{\Gamma}}_{sp}$	$\widehat{\mathbf{G}}_n^{'}$	$\widehat{\mathbf{C}}_n^{'}$	$\widehat{\mathbf{\Gamma}}^{'}$
interc1	64	98.5%	74.6%	1.4%	99.2%	99.2%	98.8%	NA	NA	NA
interc2	192	98.6%	74.7%	11.1%	99.3%	99.5%	99.1%	NA	NA	NA
interc3	512	99.5%	74.9%	8.1%	99.8%	99.3%	99.4%	NA	NA	NA
interc4	1024	99.7%	75%	11.1%	99.9%	99.7%	99.7%	NA	NA	NA
LNA	79	95.2%	64.1%	NA	98.7%	94%	NA	95.7%	67.5%	NA
MX3	110	97.3%	80.6%	NA	99.1%	94.2%	NA	98.1%	92.3%	NA
RCintc	663	96.7%	69.5%	NA	99.4%	97%	NA	98.6%	88.5%	NA

APPENDIX A
PROOF THAT THE COPIES OF POSITIVE AND NEGATIVE BLOCKS OF (12) REMAIN EQUAL AFTER EXECUTION OF ALGORITHM 4

The block doubling process of (12) creates a pair of graph edges for each off-diagonal element $a_{ij} = a_{ji}$ of $\mathbf{A} = \mathbf{A}_{dn} + \mathbf{A}_p$. In order that the two copies of $\mathbf{A}_{dn}$ and $\mathbf{A}_p$ remain identical after sparsification, we have to show that both edges of each pair are either kept or discarded by Algorithm 4. This will be true if both edges have equal probabilities in step 7 of the algorithm, so that they have equal expected number of selections in q trials (step 17)[3]. Since the weights of these edges are equal, it suffices to show that their effective resistances are also equal. Let in the definition of effective resistance (13):

$$\mathbf{L}_G = \begin{bmatrix} \mathbf{A}_{dn} & -\mathbf{A}_p \\ -\mathbf{A}_p & \mathbf{A}_{dn} \end{bmatrix}$$

with $\mathbf{A}_{dn}, \mathbf{A}_p \in \Re^{r\times r}$.
Then by blockwise inversion we have:

$$\mathbf{L}_G^+ = \begin{bmatrix} \mathbf{S} & \mathbf{T} \\ \mathbf{T} & \mathbf{S} \end{bmatrix} \tag{16}$$

where $\mathbf{S} = \left(\mathbf{A}_{dn} - \mathbf{A}_p\mathbf{A}_{dn}^+\mathbf{A}_p\right)^+$ and $\mathbf{T} = \mathbf{A}_{dn}^+\mathbf{A}_p\mathbf{S} = \mathbf{S}\mathbf{A}_p\mathbf{A}_{dn}^+$.

Suppose now a_{ij} $(i \neq j)$ is an off-diagonal element of $\mathbf{A} = \mathbf{A}_{dn} + \mathbf{A}_p$.

If $a_{ij} > 0$ then it is contained in $\mathbf{A}_p$ and corresponds to edges $(i, r+j)$ and $(r+i, j)$ in the double-sized Laplacian matrix $\mathbf{L}_G$. Then the $2r \times 1$ vectors $(\mathbf{e}_i - \mathbf{e}_{r+j}), (\mathbf{e}_{r+i} - \mathbf{e}_j)$ have the forms:

$$\mathbf{e}_i - \mathbf{e}_{r+j} = \begin{bmatrix} \mathbf{e}_i^{'} \\ -\mathbf{e}_j^{'} \end{bmatrix}, \quad \mathbf{e}_{r+i} - \mathbf{e}_j = \begin{bmatrix} -\mathbf{e}_j^{'} \\ \mathbf{e}_i^{'} \end{bmatrix} \tag{17}$$

where $\mathbf{e}_i^{'}, \mathbf{e}_j^{'}$ are the $r \times 1$ elementary vectors. From (16) and (17) it can be verified by matrix-vector manipulations that $(\mathbf{e}_i - \mathbf{e}_{r+j})^T \mathbf{L}_G^+ (\mathbf{e}_i - \mathbf{e}_{r+j}) = (\mathbf{e}_{r+i} - \mathbf{e}_j)^T \mathbf{L}_G^+ (\mathbf{e}_{r+i} - \mathbf{e}_j)$, i.e. $R_{(i,r+j)}^{eff} = R_{(r+i,j)}^{eff}$.

If $a_{ij} < 0$ then it is contained in $\mathbf{A}_{dn}$ and corresponds to edges (i, j) and $(r+i, r+j)$ in $\mathbf{L}_G$. The $2r \times 1$ vectors $(\mathbf{e}_i - \mathbf{e}_j), (\mathbf{e}_{r+i} - \mathbf{e}_{r+j})$ have the forms:

$$\mathbf{e}_i - \mathbf{e}_j = \begin{bmatrix} \mathbf{e}_i^{'} - \mathbf{e}_j^{'} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{e}_{r+i} - \mathbf{e}_{r+j} = \begin{bmatrix} \mathbf{0} \\ \mathbf{e}_i^{'} - \mathbf{e}_j^{'} \end{bmatrix} \tag{18}$$

and from (16), (18) it can be verified by matrix-vector manipulations that $(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{L}_G^+ (\mathbf{e}_i - \mathbf{e}_j) = (\mathbf{e}_{r+i} - \mathbf{e}_{r+j})^T \mathbf{L}_G^+ (\mathbf{e}_{r+i} - \mathbf{e}_{r+j})$, i.e. $R_{(i,j)}^{eff} = R_{(r+i,r+j)}^{eff}$.

[3]If Algorithm 4 is selected to start with the maximum likelihood spanning tree, then it can potentially include an edge of fairly low probability (not normally selected in q trials) whose matching edge will not be among the remaining selected edges. Although the inclusion of such low-probability edges will be very scarce, we can always include the matching edges to the sparse graph to still ensure the validity of (15).

REFERENCES

[1] A. Odabasioglu, M. Celik and L.T. Pileggi, *PRIMA: passive reduced-order interconnect macromodeling algorithm*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 17, pp. 645-654, 1998.

[2] J. Phillips, L. Daniel and L. M. Silveira, *Guaranteed passive balancing transformations for model order reduction*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, pp. 1027-1041, 2003.

[3] J. M. S. Silva and L. M. Silveira, *On the Compressibility of Power Grid Models*, IEEE Computer Society Annual Symposium on VLSI, 2007.

[4] Z. Ye, D. Vasilyev, Z. Zhu and J. R. Phillips, *Sparse Implicit Projection (SIP) for Reduction of General Many-Terminal Networks*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2008.

[5] R. Ionutiu, J. Rommes and W. H. A. Schilders, *SparseRC: Sparsity Preserving Model Reduction for RC Circuits With Many Terminals*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, pp. 1828-1841, 2011.

[6] P. Miettinen, M. Honkala, J. Roos and M. Valtonen, *PartMOR: Partitioning-Based Realizable Model-Order Reduction Method for RLC Circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 30, pp. 374-387, 2011.

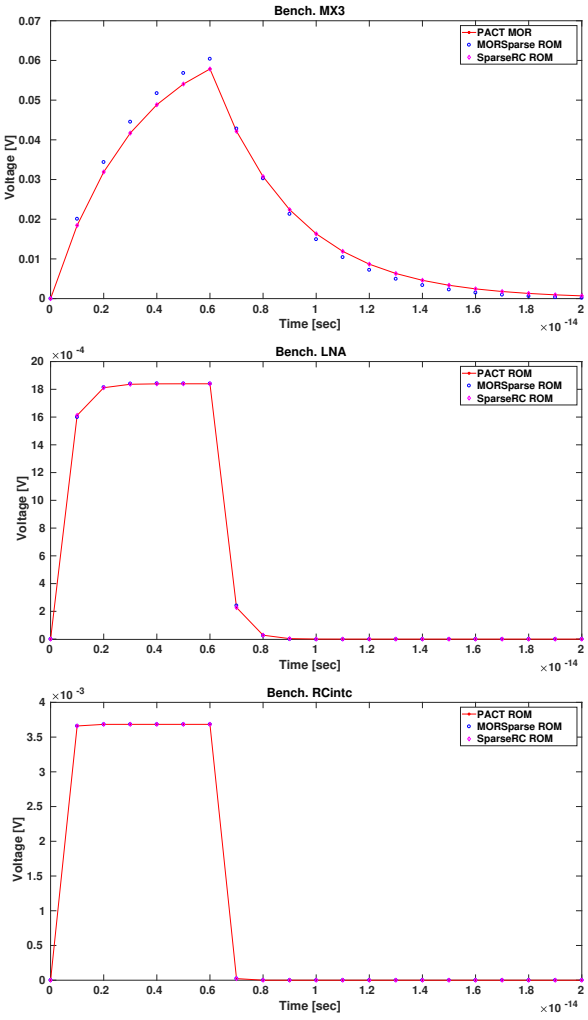

Fig. 5: Voltage response of PACT, MORSparse, and SparseRC ROMs at the port with the worst-case rms error for benchmarks MX3, LNA, and RCintc.

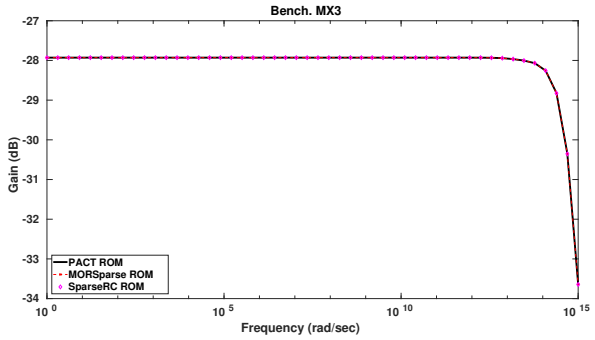

Fig. 6: Frequency Response of PACT, MORSparse and SparseRC ROMs at the port with the worst-case rms error for benchmark MX3.

[7] H. Yu, C. Chu, Y. Shi, D. Smart L. He and S. X.-D. Tan *Fast Analysis of Large Scale Inductive Interconnect by Block Structure Preserved Macromodeling*, IEEE Transactions on Very Large Scale Integration Systems, vol. 18, pp. 1399-1411, 2009.

[8] B. N. Sheehan, *TICER: Realizable reduction of extracted RC circuits*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 1999.

[9] C. S. Amin, M. H. Chowdhury and Y. I. Ismail, *Realizable reduction of interconnect circuits including self and mutual inductances*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 24, pp. 271-277, 2005.

[10] C. Antoniadis, N. Evmorfopoulos and G. Stamoulis, *Efficient Sparsification of Dense Circuit Matrices in Model Order Reduction*, IEEE/ACM Asia and South Pacific Design Automation Conference (ASP-DAC), 2019.

[11] C. Antoniadis, N. Evmorfopoulos and G. Stamoulis, *A Rigorous Approach for the Sparsification of Dense Matrices in Model Order Reduction of RLC Circuits*, IEEE/ACM Design Automation Conference (DAC), 2019.

[12] A. E. Ruehli, *Circuit Analysis, Simulation and Design, Part 1*, New York: North-Holland, 1986.

[13] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[14] A. Devgan, H. Ji and W. Dai, *How to Efficiently Capture On-Chip Inductance Effects: Introducing a New Circuit Element K*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2000.

[15] R. Merris, *Laplacian Matrices of a Graph: A Survey*, Linear Algebra and its Applications, vol. 197, pp. 143-176, 1994.

[16] R. W. Freund, *SPRIM: Structure-Preserving Reduced-Order Interconnect Macromodeling*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2004.

[17] 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*, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2004.

[18] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, Society for Industrial and Applied Mathematics, 2005.

[19] M. Mendoza, M. Raydan and P. Tarazaga, *Computing the Nearest Diagonally Dominant Matrix*, Numerical Linear Algebra with Applications, vol. 5, pp. 461-474, 1998.

[20] M. Monsalve, J. Moreno, R. Escalante and M. Raydan, *Selective Alternating Projections to Find the Nearest SDD+ Matrix*, Applied Mathematics and Computation, vol. 145, pp. 205-220, 2003.

[21] F. Yang, X. Zeng, Y. Su, D. Zhou, *RLCSYN: RLC Equivalent Circuit Synthesis for Structure-Preserved Reduced-order Model of Interconnect*, IEEE International Symposium on Circuits and Systems, 2007.

[22] K. J. Kerns and A. T. Yang, *Stable and Efficient Reduction of Large, Multiport RC Networks by Pole Analysis via Congruence Transformations*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 16, pp. 734-744, 1997.

[23] D. E. Crabtree, *Applications of M-Matrices to Non-Negative Matrices*, Duke Mathematical Journal, vol. 33, pp. 197-208, 1966.

[24] D. Spielman and S.H. Teng, *Spectral Sparsification of Graphs*, SIAM Journal on Computing, vol. 40, pp. 981âĂŞ1025, 2011.

[25] D. Spielman and N. Srivastava, *Graph Sparsification by Effective Resistances*, SIAM Journal on Computing, vol. 40, pp. 1913-1926, 2011.

[26] Z. Feng, *Spectral graph sparsification in nearly-linear time leveraging efficient spectral perturbation analysis*, IEEE/ACM Design Automation Conference (DAC), 2016.

[27] M. Kamon, M. J. Tsuk and J. K. White, *FASTHENRY: A Multipole-Accelerated 3-D Inductance Extraction Program*, IEEE Transactions on Microwave Theory and Techniques, vol. 42, pp. 1750-1758, 1994.