Xu Lei

Bordered Block-Diagonal Preserved Model-Order Reduction for RLC Circuits

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 25.5.2011

Thesis supervisor:

Prof. Martti Valtonen

Thesis instructor:

Lic.Sc. (Tech.) Mikko Honkala

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Xu Lei

Title: Bordered Block-Diagonal Preserved Model-Order Reduction for RLC Circuits

Date: 25.5.2011 Language: English Number of pages:8+40

Department of Radio Science and Technology

Professorship: Circuit theory Code: S-55

Supervisor: Prof. Martti Valtonen

Instructor: Lic.Sc. (Tech.) Mikko Honkala

This thesis details the research of the bordered block-diagonal preserved model-order reduction (BVOR) method and implementation of the corresponding tool designed for facilitating the simulation of industrial, very large sized linear circuits or linear subcircuits of a nonlinear circuit. The BVOR tool is able to extract the linear RLC parts of the circuit from any given typical SPICE netlist and perform reduction using an appropriate algorithm for optimum efficiency.

The implemented algorithms in this tool are bordered block-diagonal matrix solver and bordered block-diagonal matrix based block Arnoldi method.

Keywords: bordered block-diagonal preserved model-order reduction, RLC circuits, circuit simulation

Preface

This thesis was written in the research group of Circuit Theory of Department of Radio Science and Engineering of Aalto University.

I wish to express my gratitude to Prof. Martti Valtonen for giving me the opportunity to work in the research group of Circuit Theory and to write this thesis. Especially, I would like to thank my instructor Lic.Sc.(Tech.) Mikko Honkala and Lic.Sc.(Tech.) Pekka Miettinen for their help and guidance in writing this thesis.

Otaniemi, 25.5.2011

Xu Lei

Contents

Symbols and abbreviations

Symbols

$\mathbf{\Lambda}$	a diagonal matrix containing the eigenvalues of $\widetilde{\mathbf{G}}^{-1}\widetilde{\mathbf{C}}$ as its diagonal elements
$\mathcal{A}$	moment generation matrix
$\mathcal{B}_{\text{MNA}}$	MNA selector matrix consisting of ones, minus ones, and zeros
$\mathcal{B}$	VNA selector matrix consisting of ones, minus ones, and zeros
$\widetilde{\mathbf{B}}$	reduced port matrix
$\mathbf{B}_{\text{main}}$	main block port matrix
$\mathbf{B}_{\text{sub}}$	sub block port matrix
b_i	one circuit branch
$\mathcal{C}_{\text{MNA}}$	MNA capacitance matrix
$\mathcal{C}$	VNA capacitance matrix
$\widetilde{\mathbf{C}}$	reduced capacitance matrix
$\mathbf{C}_{\text{main}}$	main block capacitance matrix
$\mathbf{C}_{\text{sub}}$	sub block capacitance matrix
$\mathbf{C}_1$	matrix containing the stamps for capacitances
C	the number of capacitances
c_x	capacitor
c_0	self-capacitor
$Cond.$	condition number of $\widetilde{\mathbf{G}} + s_0\widetilde{\mathbf{C}}$
$\mathbf{E}_l$	additional incidence stamps for branch currents generated for the MNA equations
$\mathbf{E}_i$	additional incidence stamps for nodal voltages generated for the MNA equations
$E_{y_{ii}}$	average error of total y paramerters ($\frac{1}{n}\sum_{i=0}^{n}\triangle y_i$)
$\mathcal{G}_{\text{MNA}}$	MNA conductance matrix
$\mathcal{G}$	VNA conductance matrix
$\widetilde{\mathbf{G}}$	reduced conductance matrix
$\mathbf{G}_{\text{main}}$	main partitioned block
$\mathbf{G}_{\text{sub}}$	sub block
$\mathbf{G}_1$	matrix containing the stamps for inductances
$\mathcal{G}_{i,i} + s\mathcal{C}_{i,i}$	diagonal block
G	the number of resistances
g_x	conductor
g_0	self-conductor
$\mathbf{H}(s)$	transfer function
$\widetilde{H}(s)$	reduced system
$\mathbf{i}$	port currents
$\mathbf{i}_l$	inductive-branch current
$\mathbf{i}_i$	branch current for voltage sources
$\mathbf{I}_0$	identity matrix
k_0	self susceptor
k_x	mutual susceptor
$\mathbf{L}$	inductance matrix

$\mathbf{L}^{-1}$	susceptance matrix
L	the number of inductances
L_m	the number of mutual inductances
$\mathbf{M}$	diagonal matrix in BBD-structured state matrix
$\mathbf{M}_i$	the block moments of $\mathbf{H}(s)$
$\mathbf{M}_i$	partitioning block
n_i	number of internal nodes
n_e	number of external nodes
n_l	number of inductances
n_p	number of partitions
N_{nz}	number of non-zero elements in $\widetilde{\mathbf{G}} + s_0\widetilde{\mathbf{C}}$
N_{pp}	positive poles
N	number of ports
N_q	the number of iterations of Block Arnoldi method
N_{red}	size of the reduced matrix $\widetilde{\mathbf{G}} + s_0\widetilde{\mathbf{C}}$
n	the number of all nodes
$\mathbf{Q}$	projection matrix
$\mathbf{Q}_i$	projection block matrix
Q_{col}	order of reduction
q	refers to the number of iterations of Block Arnoldi method, the number equals to $\lfloor q/n_e \rfloor$
$\mathcal{R}$	moment generation matrix
$\mathbf{S}$	transformation matrix
s_0	expansion point
t	computational time of calculating certain y parameter
$\mathbf{u}$	port voltages
$\mathbf{v}_n$	nodal voltages
v_i	nodal voltage
$\mathbf{X}$	off-diagonal matrix in BBD-structured state matrix
$\mathbf{X}_{i,0}$	connection matrix
$\mathbf{x}_{\text{MNA}}$	MNA variables (nodal voltages, branch currents of inductances and voltage sources)
$\mathbf{x}$	VNA variables (nodal voltages, magnetic fluxes and voltage sources)
y_{11}	element of y-parameter matrix
Φ	magnetic flux
ϕ_i	branch flux

Abbreviations

APLAC	an object-oriented analog circuit simulator and design tool (originally Analysis Program for Linear Active Circuits)
AC	Alternating Current
BBD	Bordered Block-Diagonal form
BVOR	Bordered Block-Diagonal Preserved Model-Order Reduction
DC	Direct Current
hMETIS	a software package for partitioning hypergraphs
MATLAB	a programmable mathematics toolbox (MATrix LABoratory)
METIS	a software package for partitioning graphs
MNA	Modified Nodal Analysis
MOR	Model-Order Reduction
PRIMA	Passive Reduced-Order Interconnect Macromodeling Algorithm
SPICE	a circuit simulator software
VNA	vector-potential based nodal analysis

1 Introduction

1.1 Model-Order Reduction in general

The rapidly-increasing size and complexity of industrial circuits results in the substantial need for faster simulation methods. One way to speed up transistor-level simulations is to apply model-order reduction (MOR). As a discipline, MOR utilizes the properties of the dynamical systems, reduces the complexity of the systems, while preserves the original functionality of the systems within an acceptable margin of error.

There exists numerous MOR tools for reducing the large-scale circuits, for example: Reduced-Order Interconnect Macromodeling Algorithm (PRIMA) [7], SPRIM [3], PACT [2], Liao-Dai [20], Time constant Equilibration Reduction (TICER) [21] [22], PartMOR [12], BVOR [6], HiPRIME [16], and hiePrimor [17]. Liao-Dai and TICER are used for circuit containing only RC elements, PRIMA, SPRIM, and PACT can be used for RLC circuit. PartMOR, BVOR, HiPRIME, and hiePrimor are partitioning-based MOR method and they can also be used for RLC circuits.

1.2 Partitioning a circuit

Circuit partitioning plays an important role in partitioning-based MOR method. In order to analyze large circuit more easily, the large circuit is partitioned into smaller subcircuits. Different partitioning algorithms are available, such as METIS and hMETIS [8]. These two methods operate on graphs and hypergraphs in general.

As introduced previously, the existing partitioning-based MOR methods are: BVOR, PartMOR, HiPRIME, hiePrimor etc. The first two methods are utilized to deal with the mutual inductance feature in/between the partitioned subcircuits. BVOR reduces mutual inductances by decoupling the inductive branch, while PartMOR in Ref. [19] and its extension in Ref. [12] utilizes the physical characteristics of a transmission line to add the mutual inductance information. HiPRIME is used to analyze RLCK power delivery systems. The last method performs the projection-based reduction on partitioned subcircuits. BVOR, PartMOR and hiePrimor use hMETIS partitioning algorithm.

1.3 Bordered Block-Diagonal Preserved Model-Order Reduction (BVOR)

Bordered Block-Diagonal Preserved Model-Order Reduction (BVOR) is a partitioning-based MOR method.

Large-scale interconnect dominant RLC circuits are usually analyzed using modified nodal analysis (MNA) [4]. The MNA formulation uses nodal voltage variables and currents of inductance. The inductance part of MNA matrix may become dense with a large number of elements. An efficient MOR needs to first sparsify the dense inductance matrix. Since matrix $\mathbf{L}$ is not diagonal-dominant, sparsifying $\mathbf{L}$ directly

thereby leads to an instable state matrix. $\mathbf{L}^{-1}$ matrix is however more diagonal-dominant than $\mathbf{L}$, and hence, $\mathbf{L}^{-1}$ can be more effectively sparsified while preserving stability. The new circuit stamping for such RLC circuits is introduced and the corresponding new analysis method is called vector-potential based nodal analysis (VNA) [5]. In addition, a bordered block-diagonal form (BBD) is presented for the flat VNA matrix in order to preserve the strucutre of the state matrices such as sparsity and hierarchy as well as efficiently build and simulate the circuit. Corresponding model-order reduction is called BBD-structure preserving model order reduction (BVOR).

Another remarkable advantage of BVOR method is the possibility to apply parallel processing on the BBD-based matrix.

1.4 Background

The thesis is description, implementation, testing and evaluation of paper "Fast Analysis of a Large-Scale Inductive Interconnect by Block-Structure-Preserved Macromodeling " of IEEE transactions on very large scale integration (VLSI) systems [5].

In the course of the project, circuit theory reserch group of Aalto University developed and implemented the circuit partition tool, MNA matrix formulation tool and linear MOR tool prototype using MATLAB and C programming language. My implementation started from the formulation of the MNA matrix.

1.5 Overview of the thesis

Section 2 describes the circuit formulation using MNA method. Based on the state matrices, Krylov subspace model-order reduction and one specific method PRIMA are introduced in detail in Section 3. Section 4 discusses BVOR method. Specifically it presents the VNA method and BBD structure with several examples and comparisons. After this, BBD solver, Block Arnoldi method, the combination of the two methods are studied separately.

Section 5 discusses macromodel realization applied on the reduced blocks. Section 6 describes the total BVOR flow and the implementation of the BVOR algorithms. Section 7 documents the extensive simulations and bench mark tests performed and compares the different BVOR algorithms in the light of these results. Finally, conclusions of the work are given in Section 8.

1.6 Author's contribution

The author's main contribution to this project was the implementation of VNA matrix formulation, implementation of BBD solver, implementation of Block Arnoldi Algorithm, utilization of existing macromodel realization and test simulations in the final phase of the project.

The performance of BVOR is also compared with PRIMA method. Projection matrices produced from individual diagonal block is discussed briefly and corre-

sponding simulation results are compared with the projection matrices generated from original BBD system. Macromodel realization is also studied.

2 Formulation of circuit equation with MNA

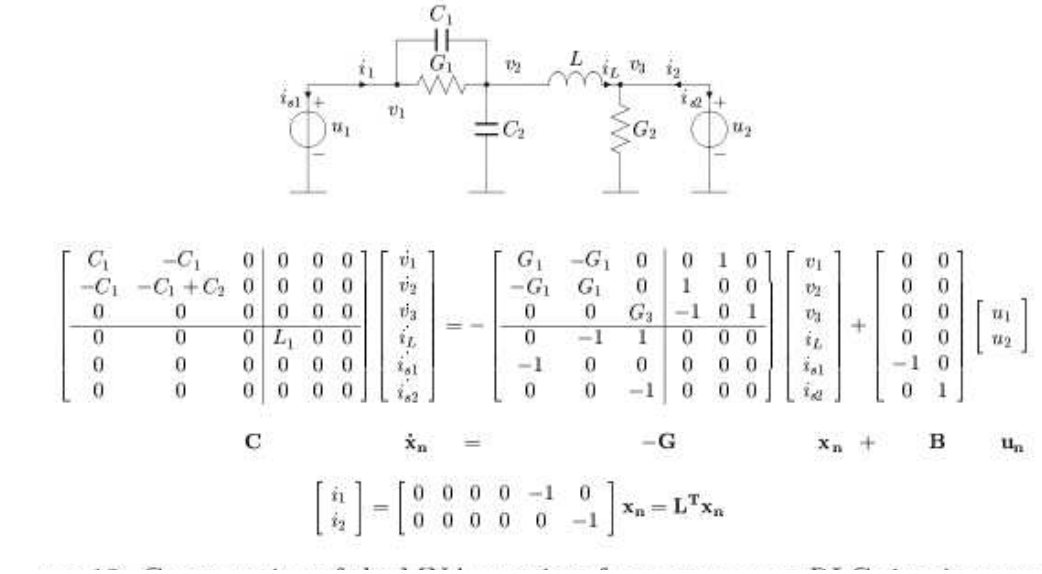

Figure 1: Construction of the MNA matrices for y-parameters from a two-port RLC circuit example [10]

One way to describe N-port circuit is to utilize the y-parameter matrix. Here, voltage sources are used at the input and output ports to excite the circuit, as shown in Fig. 1. The time-domain modified nodal analysis (MNA) circuit equations for a linear N-port RLC circuit can be expressed as

$$\begin{cases} \mathcal{C}_{\mathrm{MNA}} \dfrac{\mathrm{d}\mathbf{x}_{\mathrm{MNA}}(t)}{\mathrm{d}t} = -\mathcal{G}_{\mathrm{MNA}}\mathbf{x}_{\mathrm{MNA}}(t) + \mathcal{B}_{\mathrm{MNA}}\mathbf{u}(t), \\ \mathbf{y}(t) = \mathcal{B}_{\mathrm{MNA}}^{\mathrm{T}}\mathbf{x}_{\mathrm{MNA}}(t), \end{cases} \tag{1}$$

the corresponding first-order state equation in frequency domain is

$$\begin{cases} s\mathcal{C}_{\mathrm{MNA}}\mathbf{x}_{\mathrm{MNA}}(s) = -\mathcal{G}_{\mathrm{MNA}}\mathbf{x}_{\mathrm{MNA}}(s) + \mathcal{B}_{\mathrm{MNA}}\mathbf{u}(s), \\ \mathbf{y}(s) = \mathcal{B}_{\mathrm{MNA}}^{\mathrm{T}}\mathbf{x}_{\mathrm{MNA}}(s), \end{cases} \tag{2}$$

where $\mathcal{C}_{\mathrm{MNA}}$ and $\mathcal{G}_{\mathrm{MNA}}$ are susceptance and conductance matrices, respectively, and $\mathbf{y}$, $\mathbf{x}_{\mathrm{MNA}}$, $\mathbf{u}$ and $\mathbf{i}$ denote the port current, the MNA variables (nodal voltages and branch currents of inductances and voltage sources), port voltages, and port currents, respectively. Here, $\mathcal{B}_{\mathrm{MNA}}$ is a selector matrix consisting of ones, minus ones, and zeros. The matrices $\mathcal{C}_{\mathrm{MNA}}$ and $\mathcal{G}_{\mathrm{MNA}}$ are defined as

$$\mathcal{C}_{\mathrm{MNA}} \equiv \begin{bmatrix} \mathbf{C}_1 & 0 & 0 \\ 0 & \mathbf{L} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{G}_{\mathrm{MNA}} \equiv \begin{bmatrix} \mathbf{G}_1 & \mathbf{E}_l & \mathbf{E}_i \\ -\mathbf{E}_l^{\mathrm{T}} & 0 & 0 \\ -\mathbf{E}_i^{\mathrm{T}} & 0 & 0 \end{bmatrix}, \quad \mathbf{x}_{\mathrm{MNA}} \equiv \begin{bmatrix} \mathbf{v}_n \\ \mathbf{i}_l \\ \mathbf{i}_i \end{bmatrix}$$

where $\mathbf{C}_1$, $\mathbf{L}$ and $\mathbf{G}_1$ are the matrices containing the stamps for capacitances, inductances, and resistances respectively; $\mathbf{E}_l$ and $\mathbf{E}_i$ consist of ones, minus ones and zeros, which represent the additional incidence stamps for branch currents and nodal voltages generated for the MNA equations; and the vector $\mathbf{x}_{\text{MNA}}$ contains the nodal voltages $\mathbf{v}_n$ and branch currents $\mathbf{i}_l$, $\mathbf{i}_i$ for inductances and voltage sources. For RLC circuits, the dimension of the $\mathcal{C}_{\text{MNA}}$ and $\mathcal{G}_{\text{MNA}}$ matrix is thus $n \times n$, with $n = n_i + n_l + N$, where n_i, n_l and N are the number of internal nodes, inductances and ports respectively. One important property of the MNA formulation is that the transposed summations of state matrices are symmetric and semipositive-definite, i.e., $\mathcal{G}_{\text{MNA}} + \mathcal{G}_{\text{MNA}}^T > 0$ and $\mathcal{C}_{\text{MNA}} + \mathcal{C}_{\text{MNA}}^T > 0$.This is one of the sufficient conditions for a system to be passive. [5]

3 Krylov subspace Model-Order Reduction

There are other types of MOR methods than Krylov subspace method, but the thesis focused on Krylov subspace method.

Solving Eq. (2) following transfer function is obtained:

$$\frac{y(s)}{u(s)} = \mathbf{H}(s) = \mathcal{B}_{\text{MNA}}^{T}(\mathcal{G}_{\text{MNA}} + s\mathcal{C}_{\text{MNA}})^{-1}\mathcal{B}_{\text{MNA}} \tag{3}$$

Expanding $\mathbf{H}(s)$ at some frequency point s_0, state variable is contained in a block Krylov subspace

$$span\{\mathcal{R}, \mathcal{A}\mathcal{R}, ..., \mathcal{A}^{N_q-1}\mathcal{R}, ...\} \tag{4}$$

with two moment generation matrices

$$\mathcal{A} = (\mathcal{G}_{\text{MNA}} + s_0\mathcal{C}_{\text{MNA}})^{-1}\mathcal{C}_{\text{MNA}},\ \mathcal{R} = (\mathcal{G}_{\text{MNA}} + s_0\mathcal{C}_{\text{MNA}})^{-1}\mathcal{B}_{\text{MNA}} \tag{5}$$

First lemma [5] related to Krylov subspace and MOR is: *if a small-dimensioned matrix* $\mathbf{Q}$ *that spans the* N_q*th-order* ($N_q = \lfloor q/N \rfloor$) *block Krylov subspace is*

$$\mathcal{K}(\mathcal{A}, \mathcal{R}, q) = span\{\mathcal{R}, \mathcal{A}\mathcal{R}, ..., \mathcal{A}^{N_q-1}\mathcal{R}\} \subseteq span\{\mathbf{Q}\} \tag{6}$$

then applying $\mathbf{Q}$ *to project the original system*

$$\widetilde{\mathcal{G}}_{\text{MNA}} = \mathbf{Q}^T\mathcal{G}_{\text{MNA}}\mathbf{Q},\ \widetilde{\mathcal{C}}_{\text{MNA}} = \mathbf{Q}^T\mathcal{C}_{\text{MNA}}\mathbf{Q},\ \widetilde{\mathcal{B}}_{\text{MNA}} = \mathbf{Q}^T\mathcal{B}_{\text{MNA}} \tag{7}$$

the first q *block moments of the reduced system* $\widetilde{\mathbf{H}}(s)$

$$\widetilde{\mathbf{H}}(s) = \widetilde{\mathcal{B}}_{\text{MNA}}^{T}(\widetilde{\mathcal{G}}_{\text{MNA}} + s\widetilde{\mathcal{C}}_{\text{MNA}})^{-1}\widetilde{\mathcal{B}}_{\text{MNA}} \tag{8}$$

expanded at s_0 *are identical to the original one* $\mathbf{H}(s)$.

The projection-based model-order reduction is essentially to construct an invariant subspace that can approximate the dominant system response in terms of the first few moments expanded at s_0. The N_q determines the accuracy of the reduced model and depends on both the number of ports (N) and the order of reduction (q). A detailed analysis on how to select q can be found in [13]. Note that PRIMA applies a block Arnoldi method to construct an orthonormalized projection matrix $\mathbf{Q}$. Section 3.1 introduces PRIMA in brief.

Second lemma [5] is: *when the input and the output are symmetric, the reduced model* $\widetilde{\mathbf{H}}(s)$ *is passive when projected by an orthonormalized* $\mathbf{Q}$ *in a fashion similar to that of the congruence transformation Eq. (7).*

Let us consider the special case $s_0 = 0$, Eq. (5) becomes:

$$\mathcal{A} = \mathcal{G}_{\text{MNA}}^{-1}\mathcal{C}_{\text{MNA}},\ \mathcal{R} = \mathcal{G}_{\text{MNA}}^{-1}\mathcal{B}_{\text{MNA}} \tag{9}$$

the corresponding transfer function at $s_0 = 0$ is written as

$$\mathbf{H}(s) = \mathcal{B}_{\text{MNA}}^{\text{T}}(\mathbf{I} + s\mathcal{A})^{-1}\mathcal{R}, \tag{10}$$

where $\mathbf{I}$ is the $n \times n$ identity matrix. The block moments of $\mathbf{H}(s)$ are defined as the coefficients of the Taylor expansion of $\mathbf{H}$ around $s_0 = 0$:

$$\mathbf{H}(s) = \mathbf{M}_0 + \mathbf{M}_1 s + \mathbf{M}_2 s^2 + \cdots. \tag{11}$$

Here, $(\mathbf{I} + s\mathcal{A})^{-1}$ is expanded into a Neumann series. For analysis purposes, the zeroth block moment may be perceived as the DC properties of the circuit, with the higher moments corresponding to the frequency behavior of the circuit. The block moments can be computed using the relation

$$\mathbf{M}_i = \mathcal{B}_{\mathrm{MNA}}^{\mathrm{T}} \mathcal{A}^i \mathcal{R}. \tag{12}$$

Note that the dimension of the block moments $\mathbf{M}_i$ is the same as the number of ports in the circuit.

3.1 Passive reduced-order macromodeling algorithm (PRIMA)

Based on the block Arnoldi algorithm, a passive reduced-order macromodeling algorithm (PRIMA) is presented below. This algorithm is a general method for the passive reduction of the RLC circuits.

Algorithm 1
Connect voltage sources to the multi-port & obtain the MNA matrices $\mathcal{G}_{\mathrm{MNA}}$, $\mathcal{C}_{\mathrm{MNA}}$, $\mathcal{B}_{\mathrm{MNA}}$ from Eq. (2).

Solve $\mathcal{G}_{\mathrm{MNA}}\mathcal{R} = \mathcal{B}_{\mathrm{MNA}}$ for $\mathcal{R}$

$(\mathbf{Q}_0, \mathbf{T}) = qr(\mathcal{R})$; qr factorization of $\mathcal{R}$

If q/N is not an integer, set $N_q = \lfloor q/N \rfloor + 1$, else set $N_q = q/N$

For $k = 1, 2, ..., N_q$

Set $\mathbf{V} = \mathcal{C}_{\mathrm{MNA}}\mathbf{Q}_{k-1}$
Solve $\mathcal{G}_{\mathrm{MNA}}\mathbf{Q}_k^{(0)} = \mathbf{V}$ for $\mathbf{Q}_k^{(0)}$
For $j = 1, ..., k$
$\mathbf{H} = \mathbf{Q}_{k-j}^{\mathbf{T}}\mathbf{Q}_k^{(j-1)}$
$\mathbf{Q}_k^{(j)} = \mathbf{Q}_k^{(j-1)} - \mathbf{Q}_{k-j}\mathbf{H}$
$(\mathbf{Q}_k, \mathbf{T}) = qr(\mathbf{Q}_k^{(k)})$; qr factorization of $\mathbf{Q}_k^{(k)}$

Set $\mathbf{Q} = [\mathbf{Q}_0, \mathbf{Q}_1, ..., \mathbf{Q}_k]$ and truncate $\mathbf{Q}$ so that it has N_q columns only.

---

4 BVOR method

BBD-structure (bordered block-diagonal form) preserving model order reduction, shortly BVOR, is a partitioning-based MOR method which is utilized to reduce large-scale RLC circuit containing mutual inductances.

BVOR method uses hMETIS algorithm to partition the circuit into subcircuit. Then form VNA (vector-potential based nodal analysis) matrices for each subcircuit. Afterwards, BVOR method applies node splitting and branch tearing procedures to the couplings and connections between subcircuits such that state matrix (VNA) in BBD structure is generated for the large-scale RLC network. The state matrix is further reduced using a set of orthonormalized projection vectors $\mathbf{Q}$ by BVOR method. The projection is performed by the block Arnoldi procedure.

Summary of BVOR method:

1. Use hMETIS algorithm to partition the circuit
2. Form VNA matrices for each subcircuit
3. Apply node splitting and branch tearing on VNA matrices to produce VNA matrices in BBD structure
4. Perform BBD-based block Arnoldi method that utilizes BBD matrix solver to reduce the BBD matrices
5. Generated structured projection matrix $\mathbf{Q}$ is further used to reduce the original BBD matrices

This section discusses BVOR method in detail. Firstly VNA formation and BBD structure are introduced, then corresponding matrix-based model-order reduction (BVOR) is studied. At last alternative way to construct orthonormalized projection vectors $\mathbf{Q}$ using individual diagonal block is discussed briefly.

The reduced matrices are realized by macromodel realization which will be discussed in next section.

4.1 Vector-potential based nodal analysis (VNA)

The relation between a magnetic flux Φ, a branch inductance $\mathbf{L}$, and an inductive-branch current $\mathbf{i}_l$ is:

$$\Phi = \mathbf{L}\mathbf{i}_l \tag{13}$$

Note that the branch voltage drop at an inductor can be calculated as $\mathbf{v}_l = s\mathbf{L}\mathbf{i}_l$, or can be selected from the nodal voltage $\mathbf{v}_n$ by $\mathbf{v}_l = \mathbf{E}_l^T\mathbf{v}_n$. Thus the following relation between the magnetic flux Φ (branch variable) and the nodal voltage $\mathbf{v}_n$ can be derived by

$$s\Phi = \mathbf{E}_l^T\mathbf{v}_n \tag{14}$$

Furthermore, first equation of Eq. (2) leads to

$$\left[\begin{bmatrix} \mathbf{G}_1 & \mathbf{E}_l & \mathbf{E}_i \\ -\mathbf{E}_l^{\mathrm{T}} & 0 & 0 \\ -\mathbf{E}_i^{\mathrm{T}} & 0 & 0 \end{bmatrix} + s \begin{bmatrix} \mathbf{C}_1 & 0 & 0 \\ 0 & \mathbf{L} & 0 \\ 0 & 0 & 0 \end{bmatrix}\right] \begin{bmatrix} \mathbf{v}_n \\ \mathbf{i}_l \\ \mathbf{i}_i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{B} \end{bmatrix} u(s)$$

then the following two independent equations can be led to:

$$s\mathbf{i}_l = \mathbf{L}^{-1}\mathbf{E}_l^T\mathbf{v}_n \tag{15}$$

and

$$(\mathbf{G}_1 + s\mathbf{C}_1)\mathbf{v}_n + \mathbf{E}_l\mathbf{i}_l + \mathbf{E}_i\mathbf{i}_i = 0 \tag{16}$$

According to Eq. (13), Eq. (15) becomes

$$\mathbf{i}_l = \mathbf{L}^{-1}\Phi \tag{17}$$

and Eq. (16) becomes

$$(\mathbf{G}_1 + s\mathbf{C}_1)\mathbf{v}_n + \mathbf{E}_l(\mathbf{L}^{-1}\Phi) + \mathbf{E}_i\mathbf{i}_i = 0 \tag{18}$$

As a result, based on Eq. (14) and Eq. (18), a new MNA equation with a first-order admittance can be obtained

$$\begin{cases} s\mathcal{C}\mathbf{x}(s) = -\mathcal{G}\mathbf{x}(s) + \mathcal{B}\mathbf{u}(s), \\ \mathbf{y}(s) = \mathcal{B}^{\mathrm{T}}\mathbf{x}(s), \end{cases} \tag{19}$$

where

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_n \\ \Phi \\ \mathbf{i}_i \end{bmatrix} \tag{20}$$

is a new vector of state variables composed by the nodal voltage, flux, and branch current for the external voltage source. The new state matrices $\mathcal{G}$, $\mathcal{C}$ and $\mathcal{B}$ become

$$\mathcal{G} = \begin{bmatrix} \mathbf{G}_1 & (\mathbf{E}_l\mathbf{L}^{-1}) & \mathbf{E}_i \\ -(\mathbf{L}^{-1}\mathbf{E}_l^T) & 0 & 0 \\ -\mathbf{E}_i^T & 0 & 0 \end{bmatrix}$$

$$\mathcal{C} = \begin{bmatrix} \mathbf{C}_1 & 0 & 0 \\ 0 & \mathbf{L}^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathcal{B} = \begin{bmatrix} 0 \\ 0 \\ -\mathbf{B} \end{bmatrix}$$

We call such a new MNA the VNA.

Since the aforementioned derivation is just to replace the state variable of $\mathbf{i}_l$ by Φ, the resulting MNA has the same structure as the original MNA. The new VNA stamping results in a passive formulation of $\mathbf{L}^{-1}$ because both

$$\mathcal{G}+\mathcal{G}^T=\begin{bmatrix} 2\mathbf{G}_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \mathcal{C}+\mathcal{C}^T=\begin{bmatrix} 2\mathbf{C}_1 & 0 & 0 \\ 0 & 2\mathbf{L}^{-1} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{21}$$

are symmetric and semipositive-definite [5].

4.2 BBD structure

One efficient solution for a large-scale network is to apply hMETIS to partition the network into sub-networks and then utilize network decomposition to further decompose the couplings of node and mutual inductances between sub-networks. Network decomposition is achieved by node splitting and branch tearing for nodal voltage and branch current variables, respectively. In order to save memory, one large matrix is divided into several smaller blocks, each smaller block is solved at a time: the BBD solver is designed to efficiently solve divided blocks, which can be solved in parallel.

In addition, the network decomposition usually results in a state matrix with a bordered block diagonal (BBD) structure. The BBD matrix stretching basically introduces new columns/rows for decoupled nodes or branches. It provides a way to tear the state matrix with sparse inverse inductance in the frame work of VNA.

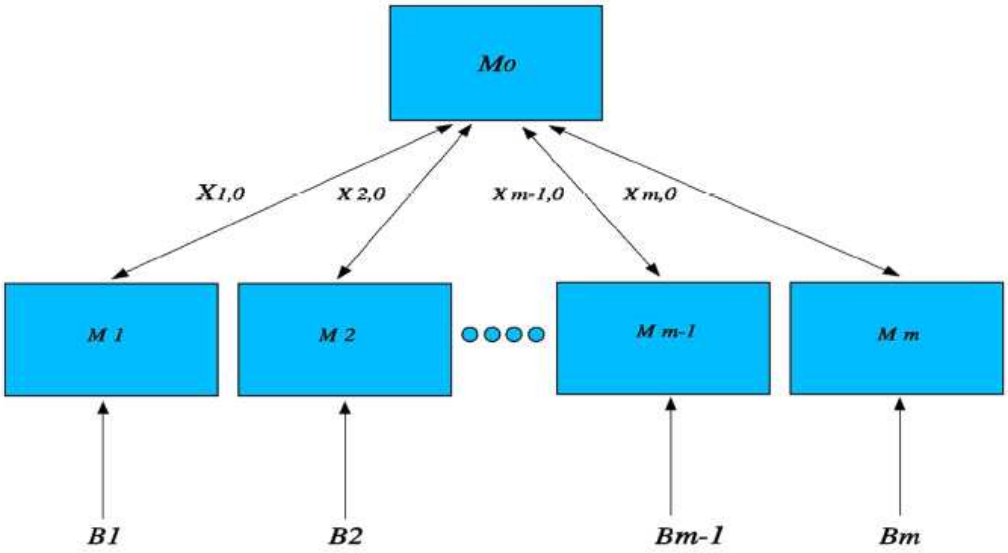

Figure 2: Two-level BBD representation of a flat VNA circuit. There is no coupling in bottom blocks, but each bottom block is connected to a centric interconnection block. (From [5])

4.2.1 Stretching of nodal voltages

A matrix stretching of nodal voltages can be described by the following rule.

Rule 1: Assume that two resistive (or capacitive) branches b_i and b_j are coupled at a common node v_2 with a conductor g_x (or a capacitor c_x). Branch b_i has nodal voltages (v_1,v_2)and b_j has nodal voltages (v_2,v_3). They can be decoupled by introducing: 1) a duplicated state variable v_2' with v_2=v_2' and 2) an auxiliary state variable $i_{2,2'}$ for a new branch current between nodes v_2 and v_2'

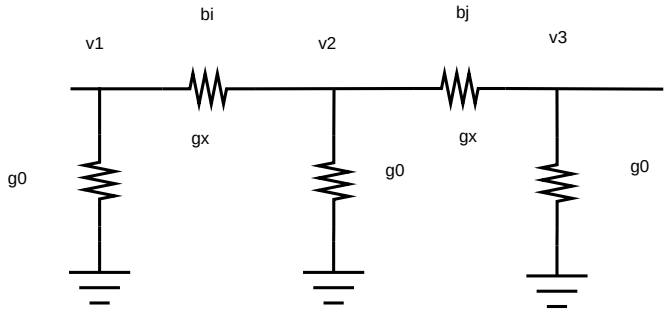

Figure 3: circuit before node splitting

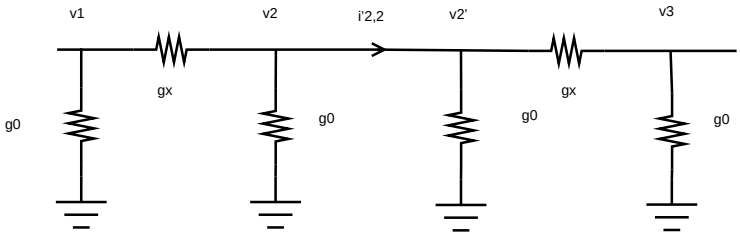

Figure 4: circuit after node splitting

The corresponding transformations from the old $\mathcal{G}$ and $\mathcal{C}$ to new $\mathbf{G}$ and $\mathbf{C}$ are

$$\mathcal{G}: \left[\begin{array}{c|ccc} & v_1 & v_2 & v_3 \\ \hline v_1 & g_0+g_x & -g_x & \\ v_2 & -g_x & g_0+2g_x & -g_x \\ v_3 & & -g_x & g_0+g_x \end{array}\right] \longrightarrow$$

$$\mathbf{G}: \left[\begin{array}{c|ccccc} & v_1 & v_2 & v_2' & v_3 & i_{2,2}' \\ \hline v_1 & g_0+g_x & -g_x & & & \\ v_2 & -g_x & 2g_0+g_x & & & +1 \\ v_2' & & & 2g_0+g_x & -g_x & -1 \\ v_3 & & & -g_x & g_0+g_x & \\ i_{2,2}' & & -1 & +1 & & \end{array}\right]$$

and

$$\mathcal{C}: \left[\begin{array}{c|ccc} & v_1 & v_2 & v_3 \\ \hline v_1 & c_0+c_x & -c_x & \\ v_2 & -c_x & c_0+2c_x & -c_x \\ v_3 & & -c_x & c_0+c_x \end{array}\right] \longrightarrow$$

$$\mathbf{C}: \left[\begin{array}{c|ccccc} & v_1 & v_2 & v_2' & v_3 & i_{2,2}' \\ \hline v_1 & c_0+c_x & -c_x & & & \\ v_2 & -c_x & 2c_0+c_x & & & +1 \\ v_2' & & & 2c_0+c_x & -c_x & -1 \\ v_3 & & & -c_x & c_0+c_x & \\ i_{2,2}' & & -1 & +1 & & \end{array}\right]$$

respectively, where g_0 and c_0 are the self-conductance and self-capacitance respectively at each node. Since the resistor and the capacitor are represented by nodal voltage, such a node splitting can be efficiently applied to decouple the RC network. The node splitting, however, cannot handle inductance or its inverse element because inductance is described by branch current. It is quite possible that two branch currents at two partitioned blocks are still coupled by mutual inductance.

4.2.2 Stretching of branch currents

We call the entries of $\mathbf{L}^{-1}$ the susceptor, which includes the self-susceptor k_0 and the mutual susceptor k_x. To cleanly decouple the inductive couplings between two partitioned blocks, a matrix stretching of inductive-branch currents can be described by the following rule.

Rule 2: Assume that two inductive branches b_i and b_j are coupled by a mutual susceptor k_x. Branch b_i has nodal voltages (v_1,v_2) and b_j has nodal voltages (v_3,v_4). Moreover, b_i has a branch flux ϕ_i and b_j has a branch flux ϕ_j. These can be decoupled by introducing an auxiliary state variable ϕ_{ij} that describes the flux difference by $\phi_{ij} = \phi_i - \phi_j$.

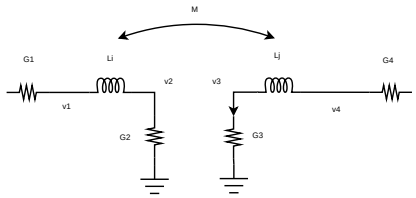

Figure 5: circuit with one mutual inductance

The following transformations from the old $\mathcal{G}$ and $\mathcal{C}$ to the new $\mathbf{G}$ and $\mathbf{C}$ illustrates how to separate two blocks which are coupled with mutual inductance:

$$\mathcal{G}:\left[\begin{array}{c|cccccc} & v_1 & v_2 & \phi_i & v_3 & v_4 & \phi_j \\ \hline v_1 & G_1 & 0 & k_0 & & & k_x \\ v_2 & 0 & G_2 & -k_0 & & & -k_x \\ \phi_i & -k_0 & k_0 & 0 & -k_x & k_x & 0 \\ v_3 & & & k_x & G_3 & 0 & k_0 \\ v_4 & & & -k_x & 0 & G_4 & -k_0 \\ \phi_j & -k_x & k_x & 0 & -k_0 & k_0 & 0 \end{array}\right] \longrightarrow$$

$$\mathbf{G}:\left[\begin{array}{c|ccccccc} & v_1 & v_2 & \phi_i & v_3 & v_4 & \phi_j & \phi_{ij} \\ \hline v_1 & G_1 & 0 & k_0' & & & & -k_x \\ v_2 & 0 & G_2 & -k_0' & & & & k_x \\ \phi_i & -k_0' & k_0' & 0 & & & & 0 \\ v_3 & & & & G_3 & 0 & k_0' & k_x \\ v_4 & & & & 0 & G_4 & -k_0' & -k_x \\ \phi_j & & & & -k_0' & k_0' & 0 & 0 \\ \phi_{ij} & k_x & -k_x & 0 & -k_x & k_x & 0 & 0 \end{array}\right]$$

and

$$\mathcal{C}:\left[\begin{array}{c|cccccc} & v_1 & v_2 & \phi_i & v_3 & v_4 & \phi_j \\ \hline v_1 & 0 & 0 & 0 & & & 0 \\ v_2 & 0 & 0 & 0 & & & 0 \\ \phi_i & 0 & 0 & k_0 & 0 & 0 & k_x \\ v_3 & & & 0 & 0 & 0 & 0 \\ v_4 & & & 0 & 0 & 0 & 0 \\ \phi_j & 0 & 0 & k_x & 0 & 0 & k_0 \end{array}\right] \longrightarrow$$

$$\mathbf{C}:\left[\begin{array}{c|ccccccc} & v_1 & v_2 & \phi_i & v_3 & v_4 & \phi_j & \phi_{ij} \\ \hline v_1 & 0 & 0 & 0 & & & & 0 \\ v_2 & 0 & 0 & 0 & & & & 0 \\ \phi_i & 0 & 0 & k_0' & & & & 0 \\ v_3 & & & & 0 & 0 & 0 & 0 \\ v_4 & & & & 0 & 0 & 0 & 0 \\ \phi_j & & & & 0 & 0 & k_0' & 0 \\ \phi_{ij} & 0 & 0 & 0 & 0 & 0 & 0 & -k_x \end{array}\right]$$

respectively, where $k_0' = k_0 + k_x$. Note that Rule 2 obtains an equivalent solution by finding a summed equivalent state matrix

$$\mathcal{G} + s\mathcal{C} \rightarrow \mathbf{G} + s\mathbf{C}. \tag{22}$$

This is due to the following node-branch relations in Eq. (14):

$$v_1{}^i - v_2{}^i = s\Phi_i, \quad v_1{}^j - v_2{}^j = s\Phi_j. \tag{23}$$

4.3 VNA matrix with BBD structure

The node splitting is applied to split the connected resistive or capacitive branches between two coupled blocks, while the branch tearing is applied to tear the coupled inductive branches between two coupled blocks. The resulting m blocks with no coupling in between are at the bottom level of the BBD, and are represented by $\mathbf{M}_i (i = 1, ..., m)$. The top level is the global interconnection block represented by $\mathbf{M}_0$, which is connected with one diagonal block $\mathbf{M}_i (i \neq 0)$ by the corresponding connection matrix $\mathbf{X}_{i,0}$. The interconnection block has size n_0 and contains all coupling branches between any pair of blocks at the bottom level.

Precisely, the resulting system equation is

$$(\mathbf{G} + s\mathbf{C})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s) \tag{24}$$

where

$$\mathbf{G}: \begin{bmatrix} \mathcal{G}_{1,1} & 0 & \cdots & 0 & \mathbf{X}^g_{1,0} \\ 0 & \mathcal{G}_{2,2} & \cdots & 0 & \mathbf{X}^g_{2,0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathcal{G}_{m,m} & \mathbf{X}^g_{m,0} \\ -(\mathbf{X}^g_{1,0})^T & -(\mathbf{X}^g_{2,0})^T & \cdots & -(\mathbf{X}^g_{m,0})^T & \mathcal{G}_{0,0} \end{bmatrix}$$

$$\mathbf{C}: \begin{bmatrix} \mathcal{C}_{1,1} & 0 & \cdots & 0 & \mathbf{X}^c_{1,0} \\ 0 & \mathcal{C}_{2,2} & \cdots & 0 & \mathbf{X}^c_{2,0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathcal{C}_{m,m} & \mathbf{X}^c_{m,0} \\ -(\mathbf{X}^c_{1,0})^T & -(\mathbf{X}^c_{2,0})^T & \cdots & -(\mathbf{X}^c_{m,0})^T & \mathcal{C}_{0,0} \end{bmatrix}$$

$$\mathbf{B}: \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \mathcal{B}_0 \end{bmatrix}$$

and $\mathbf{x} = [x_1, x_2, ..., x_m, x_0]^T$, $\mathbf{u} = [u_1, u_2, ..., u_m, 0]^T$, where $\mathbf{G}, \mathbf{C} \in R^{N \times N}$, and $\mathbf{B} \in R^{N \times n_p}$. For each block $\mathbf{M}_i$, the state variable $\mathbf{x}_i$ includes the nodal voltage v_n for the block conductance and capacitance, and the magnetic flux Φ. Its first-order VNA admittance is $\mathcal{G}_{ii} + s\mathcal{C}_{ii} (\in R^{n_i \times n_i})$, and $\mathcal{B}_i$ is 0.

The diagonal blocks $\mathbf{M}_i$ are interconnected with $\mathbf{M}_0$ by the torn branches $\mathbf{X}^{g,c}_{i0}$ $(\in R^{n_i \times n_0})$ in the border. For the global interconnection block $\mathbf{M}_0$ at the bottom, the state variable includes the interfacing current variables for resistive or capacitive node splitting and the new state variable describing the flux difference for inductive branch tearing. Its first-order VNA admittance is $(\mathbf{G})_{0,0} + s(\mathbf{C})_{0,0}$ $(\in R^{n_0 \times n_0})$. In addition, all external sources are counted by $\mathbf{B}_0$ for $\mathbf{M}_0$.

Note that $\mathcal{B}_i$ in original method presented in Ref. [5] is different than the one described here. In original method, $\mathcal{B}_i$ $(i \neq 0)$ are not 0, but contains port information of corresponding sub-circuit i and $\mathcal{B}_0$ is 0 because all external sources are

counted by $\mathcal{B}_i$ for block $\mathbf{M}_i$ (i=1,...,m) and there are no external sources in $\mathbf{M}_0$. The $\mathbf{B}$ in original form is

$$\mathbf{B}: \begin{bmatrix} \mathcal{B}_1 & 0 & \cdots & 0 & 0 \\ 0 & \mathcal{B}_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathcal{B}_m & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Let us take the circuit of Fig. 1 as an example. In the beginning, partition the circuit at dc into user-specified m blocks (here, for simplicity, m=2) using hypergraph and hmetis algorithm (See Section 5.2). State matrices of two blocks are generated. The node-split procedure is applied to the two blocks. Node v_1 is splitted into node v_1 and duplicate node v_1', node v_3 is splitted into node v_3 and duplicate node v_3'. The main block contains nodes and external ports, denoted by $\mathbf{G}_{\text{main}}, \mathbf{C}_{\text{main}}$ and $\mathbf{B}_{\text{main}}$; sub block contains duplicate nodes, original nodes in the sub block and external ports, denoted by $\mathbf{G}_{\text{sub}}, \mathbf{C}_{\text{sub}}$ and $\mathbf{B}_{\text{sub}}$. The main block is shown below. State matrix before arrow is the block after partition at dc, state matrix after arrow is the block applied by node split. Circuit after node splitting is illustrated in Fig. 6.

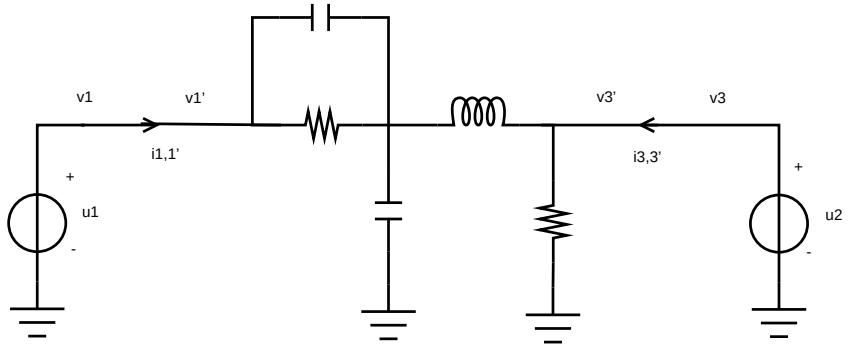

Figure 6: Two-port RLC circuit after node splitting

$$\mathbf{G}_{\text{main}}: \left[\begin{array}{c|cccc} & v_1 & v_3 & ext_1 & ext_2 \\ \hline v_1 & 0 & 0 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 1 \\ ext_1 & -1 & 0 & 0 & 0 \\ ext_2 & 0 & -1 & 0 & 0 \end{array}\right] \longrightarrow$$

$$\mathbf{G}_{0,0}: \left[\begin{array}{c|cccccc} & v_1 & v_3 & ext_1 & ext_2 & i_{1,1'} & i_{3,3'} \\ \hline v_1 & 0 & 0 & 1 & 0 & -1 & 0 \\ v_3 & 0 & 0 & 0 & 1 & 0 & -1 \\ ext_1 & -1 & 0 & 0 & 0 & 0 & 0 \\ ext_2 & 0 & -1 & 0 & 0 & 0 & 0 \\ i_{1,1'} & 1 & 0 & 0 & 0 & 0 & 0 \\ i_{3,3'} & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{C}_{\text{main}}: \left[\begin{array}{c|cccc} & v_1 & v_3 & ext_1 & ext_2 \\ \hline v_1 & 0 & 0 & 0 & 0 \\ v_3 & 0 & 0 & 0 & 0 \\ ext_1 & 0 & 0 & 0 & 0 \\ ext_2 & 0 & 0 & 0 & 0 \end{array}\right] \longrightarrow$$

$$\mathbf{C}_{0,0}: \left[\begin{array}{c|cccccc} & v_1 & v_3 & ext_1 & ext_2 & i_{1,1'} & i_{3,3'} \\ \hline v_1 & 0 & 0 & 0 & 0 & -1 & 0 \\ v_3 & 0 & 0 & 0 & 0 & 0 & -1 \\ ext_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ ext_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ i_{1,1'} & 1 & 0 & 0 & 0 & 0 & 0 \\ i_{3,3'} & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{B}_{\text{main}}: \left[\begin{array}{cc} v_1 & v_3 \\ \hline 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{array}\right] \longrightarrow \mathbf{B}_0: \left[\begin{array}{cc} v_1 & v_3 \\ \hline 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{array}\right]$$

The block matrix of the subcircuit is:

$$\mathbf{G}_{\text{sub}}: \left[\begin{array}{c|cccccc} & v_1' & v_2 & v_3' & i_l & ext_1 & ext_2 \\ \hline v_1' & G_1 & -G_1 & 0 & 0 & 1 & 0 \\ v_2 & -G_1 & G_1 & 0 & L^{-1} & 0 & 0 \\ v_3' & 0 & 0 & G_2 & -L^{-1} & 0 & 1 \\ i_l & 0 & -L^{-1} & L^{-1} & 0 & 0 & 0 \\ ext_1 & -1 & 0 & 0 & 0 & 0 & 0 \\ ext_1 & 0 & 0 & -1 & 0 & 0 & 0 \end{array}\right] \longrightarrow$$

$$\mathbf{G}_{1,1}: \left[\begin{array}{c|cccc} & v_1' & v_2 & v_3' & i_l \\ \hline v_1' & G_1 & -G_1 & 0 & 0 \\ v_2 & -G_1 & G_1 & 0 & L^{-1} \\ v_3' & 0 & 0 & G_2 & -L^{-1} \\ i_l & 0 & -L^{-1} & L^{-1} & 0 \end{array}\right] \quad \mathbf{X}_{1,0}^{g}: \left[\begin{array}{cccccc} & & & & i_{1,1'} & i_{3,3'} \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{C}_{\text{sub}}: \left[\begin{array}{c|cccccc} & v_1' & v_2 & v_3' & i_l & ext_1 & ext_2 \\ \hline v_1' & C_1 & -C_1 & 0 & 0 & 0 & 0 \\ v_2 & -C_1 & C_1+C_2 & 0 & 0 & 0 & 0 \\ v_3' & 0 & 0 & 0 & 0 & 0 & 0 \\ i_l & 0 & 0 & 0 & L^{-1} & 0 & 0 \\ ext_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ ext_2 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \longrightarrow$$

$$\mathbf{C}_{1,1}: \left[\begin{array}{c|ccc|c} & v_1' & v_2 & v_3' & i_l \\ \hline v_1' & C_1 & -C_1 & 0 & 0 \\ v_2 & -C_1 & C_1+C_2 & 0 & 0 \\ v_3' & 0 & 0 & 0 & 0 \\ i_l & 0 & 0 & 0 & L^{-1} \end{array}\right] \quad \mathbf{X}_{1,0}^{c}: \left[\begin{array}{cccc|cc} & & & & i_{1,1'} & i_{3,3'} \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{B}_{\text{sub}}: \left[\begin{array}{cc} v_1' & v_3' \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{array}\right] \longrightarrow \mathbf{B}_1: \left[\begin{array}{cc} v_1' & v_3' \\ \hline 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{array}\right]$$

Then arrange the main and sub block matrices in diagonal form:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{X}_{1,0}^{g} \\ -(\mathbf{X}_{1,0}^{g})^T & \mathbf{G}_{0,0} \end{bmatrix}$$

$$= \left[\begin{array}{cccc|cccccc} G_1 & -G_1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -G_1 & G_1 & 0 & L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & G_2 & -L^{-1} & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & -L^{-1} & L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{1,1} & \mathbf{X}_{1,0}^{c} \\ -(\mathbf{X}_{1,0}^{c})^T & \mathbf{C}_{0,0} \end{bmatrix}$$

$$= \left[\begin{array}{cccc|cccccc} C_1 & -C_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -C_1 & C_1+C_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_0 \end{bmatrix}$$

$$= \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline 0 & 0 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \\ 0 & 0 \end{array}\right]$$

4.4 BBD Solver

The size of the state matrix can be reduced by the model-order reduction by finding a set of orthonormalized projection vectors $\mathbf{Q}$. The projection is usually performed by the block Arnoldi procedure.

Directly using the block Arnoldi algorithm consumes a heavy computation to factorize the state matrix. BBD solver described below can be applied to reduce the computational cost during the block Arnoldi procedure.

Before using BBD solver, the BBD-structured state matrix $\mathbf{G} + s\mathbf{C}$ should be decomposed into the diagonal $\mathbf{M}$ and off-diagonal $\mathbf{X}$ parts, where the diagonal part $\mathbf{M}$ is

$$\mathbf{M} = \left[\begin{array}{cccc} \mathbf{M}_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \mathbf{M}_m & 0 \\ 0 & \cdots & 0 & \mathbf{M}_0 \end{array}\right]$$

$$\mathbf{M}_i = \mathcal{G}_{i,i} + s\mathcal{C}_{i,i} \ \ (i = 1, ..., m)$$

and the off-diagnonal part $\mathbf{X}$ is

$$\mathbf{X} = \left[\begin{array}{cccc} 0 & 0 & \cdots & -\mathbf{X}_1 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & -\mathbf{X}_m \\ -\mathbf{X}_1 & \cdots & -\mathbf{X}_m & 0 \end{array}\right]$$

$$\mathbf{X}_i = \mathbf{X}^g_{i,0} + s\mathbf{X}^c_{i,0} \ \ (i = 1, ..., m)$$

Algorithm 2

1. Block LU-factorization to calculate x_k

for every k in m $(k++)$ do

(1.1) input: $\mathbf{M}_k, \mathbf{X}_k, \mathcal{B}_k$;

(1.2) factor: $\mathbf{M}_k = \mathbf{L}_k\mathbf{U}_k$;

(1.3) solve: $\mathbf{L}_k\Phi_k = \mathbf{X}_k$ for Φ_k, $\Psi_k\mathbf{U}_k = (\mathbf{X}_k)^T$ for Ψ_k, and $\mathbf{L}_k\xi_k = \mathcal{B}_k$ for ξ_k;

(1.4) form: $\mathbf{F}_k = \Psi_k\Phi_k$, and $\mathbf{G}_k = \Psi_k\xi_k$

(1.5) output: $\mathbf{F}_k, \mathbf{G}_k$.

end for

2. Update interconnection $\mathbf{x}_0$

(2.1) input: $\mathbf{M}_0, \mathbf{F}_k, \mathbf{G}_k$;

(2.2) form: $\mathbf{F} = \mathbf{M}_0 + \sum_{k=1}^{m} \mathbf{F}_k, \mathbf{G} = \sum_{k=1}^{m} \mathbf{G}_k$

(2.3) solve: $\mathbf{F}\mathbf{x}_0 = \mathbf{G}$ for $\mathbf{x}_0$;

(2.4) output: $\mathbf{x}_0$.

3. Block-backward substitution of x_k

for every k in $m(k--)$ do

(3.1) input: $\mathbf{x}_0, \Phi_k, \xi_k, \mathbf{U}_k$;

(3.2) form: $\xi_k = \xi_k - \Phi_k\mathbf{x}_0$

(3.3) solve: $\mathbf{U}_k\mathbf{x}_k = \xi_k$ for $\mathbf{x}_k$

(3.4) output: $\mathbf{x}_k$

end for

The overall procedure is outlined above. Each block matrix $\mathbf{M}_i(i = 1, ..., m)$ is first solved individually with LU factorization and substitution (1.1-1.5). The results $\mathbf{x}_i(i = 1, ..., m)$ from each reduced block are then further used to solve the coupling block $\mathbf{M}_0$ for $\mathbf{x}_0$ (2.1-2.4). The final $\mathbf{x}_i$ of each reduced block is updated (3.1-3.4) with the result from the coupling current.

4.5 Block Arnoldi procedure

The new block-Arnoldi orthonormalization enhanced by the BBD solver is presented below, this is also considered as another way to implement Block Arnoldi procedure comparing with PRIMA shown in Section 3.1 :

Algorithm 3

(1.1) input: **G**, **C**, **B**;

(1.2) BBD-solve: $(\mathbf{G}+s_0\mathbf{C})\mathbf{Q}^{(0)}=\mathbf{B}$ for $\mathbf{Q}^{(0)}$

(1.3) orthonormalize: each column in $\mathbf{Q}^{(0)}$;

for every i in N_q-1 do

(1.4) BBD-solve: $(\mathbf{G}+s_0\mathbf{C})\mathbf{Q}^{(i)}=\mathbf{C}\mathbf{Q}^{(i-1)}$ for $\mathbf{Q}^{(i)}$;
(1.5) orthogonalize: $\mathbf{Q}^{(i)}$ to all $\mathbf{Q}^{(j)}$ $(j=0,...,i-1)$;
(1.6) orthonormalize: each column in $\mathbf{Q}^{(i)}$;

end for

(1.7) compose: $\mathbf{Q}=[\mathbf{Q}^{(0)},\mathbf{Q}^{(1)},...,\mathbf{Q}^{(N_q-1)}]$;

(2.1) partition: $\mathbf{Q}:\longrightarrow \mathbf{Q}_{0_{n_0\times n_{p0}}},...,\mathbf{Q}_{m_{n_m\times n_{pm}}}$

for every j in m do

(2.2) merge: $\mathbf{Q}_i$ and $\mathbf{Q}_{i+1}$ till a new $\mathbf{Q}_i'$ is nonsingular;
(2.2) orthonormalize: each column in $\mathbf{Q}_i$;

end for

(2.3) compose: $\mathbf{Q}=diag[\mathbf{Q}_1,...,\mathbf{Q}_m,\mathbf{Q}_0]$;

The order-reduced state matrices by $\mathbf{Q}$ become

$$\widetilde{\mathbf{G}}=\mathbf{Q}^T\mathbf{G}\mathbf{Q},\ \ \widetilde{\mathbf{C}}=\mathbf{Q}^T\mathbf{C}\mathbf{Q},\ \ \widetilde{\mathbf{B}}=\mathbf{Q}^T\mathbf{B} \tag{25}$$

where

$$\widetilde{\mathbf{G}}_{i,j}=\mathbf{Q}_i^T\mathbf{G}_{i,j}\mathbf{Q}_j,\ \ \widetilde{\mathbf{C}}_{i,j}=\mathbf{Q}_i^T\mathbf{C}_{i,j}\mathbf{Q}_j,\ \ \widetilde{\mathbf{B}}_i=\mathbf{Q}_i^T\mathbf{B}_i$$

As a result, the transfer function is

$$\widetilde{\mathbf{H}}(s)=\widetilde{\mathbf{B}}^T[\widetilde{\mathbf{G}}+s\widetilde{\mathbf{C}}]^{-1}\widetilde{\mathbf{B}}.$$

We call this reduction the BVOR (BBD-based VOR) method.

4.6 Q constructed with individual diagonal block

The structured projection matrix $\mathbf{Q}$ discussed in previous section is constructed with full BBD matrix as input. This section will, however, discuss another way to generate this projection matrix $\mathbf{Q}$, presented in Ref [9].

The structure of the $\mathbf{Q}$ is

$$\mathbf{Q} = diag[\mathbf{Q}_1, \mathbf{Q}_2, ..., \mathbf{Q}_0] \tag{26}$$

where $\mathbf{Q}_i \in R^{n_{b_i} \times N_q}$ (1≤ i≤ m).

Each projection block $\mathbf{Q}_i$ is constructed from each diagonal block in $\mathcal{G}_{i,i} + s\mathcal{C}_{i,i}$ independently, i.e. $\mathcal{G}_{i,i} + s\mathcal{C}_{i,i} = \mathcal{B}_i$. Note that $\mathbf{Q}_0$ is replaced with an identity matrix $\mathbf{I}_0$ ($\in R^{n_0 \times n_0}$) to avoid the zero diagonal entries in original $\mathbf{Q}_0$; and $\mathbf{B}_i$ ($i \neq 0$) is different as the one in full BBD structure. In full BBD matrix, $\mathbf{B}_i (i \neq 0)$ is empty, corresponding port information is deleted; here $\mathbf{B}_i (i \neq 0)$ contains original port information of the corresponding sub-circuit i.

The main benefit of using this type of $\mathbf{Q}$ is that only the diagonal blocks of the original BBD matrix are utilized, thus computational time is saved.

Algorithm to generate the $\mathbf{Q}$ with individual diagonal block is presented below algorithm

---

Algorithm 4

for every i in m do

(1.1) input: $\mathbf{G}_{i,i}$, $\mathbf{C}_{i,i}$, $\mathbf{B}_i$ ($i \neq 0$);

(1.2) solve: $(\mathbf{G}_{i,i} + s_0\mathbf{C}_{i,i})\mathbf{Q}^{(0)} = \mathbf{B}_i$ for $\mathbf{Q}^{(0)}$

(1.3) orthonormalize: each column in $\mathbf{Q}^{(i)}$;

for every k in $N_q - 1$ do

(1.4) solve: $(\mathbf{G}_{i,i} + s_0\mathbf{C}_{i,i})\mathbf{Q}^{(k)} = \mathbf{C}_{i,i}\mathbf{Q}^{(k-1)}$ for $\mathbf{Q}^{(k)}$;

(1.5) orthogonalize: $\mathbf{Q}^{(k)}$ to all $\mathbf{Q}^{(j)}$ ($j = 0, ..., k-1$);

(1.6) orthonormalize: each column in $\mathbf{Q}^{(k)}$;

end for

end for

(1.7) compose: $\mathbf{Q}_i = [\mathbf{Q}^{(0)}, \mathbf{Q}^{(1)}, ..., \mathbf{Q}^{(N_q-1)}] (i \neq 0)$;

(2.1) compose: $\mathbf{Q} = diag[\mathbf{Q}_1, ..., \mathbf{Q}_m, \mathbf{Q}_0]$

---

5 Macromodel realization

The reduced state matrices are used to describe the linear RLC network in frequency domain. For the purpose of transient analysis of the whole nonlinear circuit, the reduced linear part should be linked to the whole circuit. The linkage can be done by using the macromodels realization for the reduced state matrices.

There are several macromodel realization methods: Matsumoto [6], direct stamping I [7] & II [14], transfer-function realization [1], differential-equation macromodel [1] etc. In Ref. [1] it was also found out that Matsumoto's method was the fastest equivalent circuit realization of these reduced-order interconnect macromodels studied for time-domain simulation.

At first, let us replace $\mathcal{G}$, $\mathcal{C}$ and $\mathcal{B}$ with reduced $\widetilde{\mathbf{G}}$, $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{B}}$ in Eq. (1). Thus new equation is:

$$\begin{cases} \widetilde{\mathbf{C}}\dfrac{\mathrm{d}\tilde{\mathbf{x}}(t)}{\mathrm{d}t} = -\widetilde{\mathbf{G}}\tilde{\mathbf{x}}(t) + \widetilde{\mathbf{B}}\mathbf{u}(t), \\ \mathbf{i}(t) = \widetilde{\mathbf{B}}^{\mathrm{T}}\tilde{\mathbf{x}}(t), \end{cases} \tag{27}$$

Before performing the actual macromodel realization, some preprocessing is needed, namely eigenvalue decomposition.

Next, the first equation of Eq. (27) is premultiplied with $\widetilde{\mathbf{G}}^{-1}$. Assuming that a basis of eigenvectors exists for the matrix $\widetilde{\mathbf{G}}^{-1}\widetilde{\mathbf{C}}$, it can be written as $\widetilde{\mathbf{G}}^{-1}\widetilde{\mathbf{C}} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$, where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of $\widetilde{\mathbf{G}}^{-1}\widetilde{\mathbf{C}}$ as its diagonal elements and $\mathbf{S}$ has the corresponding q eigenvectors as its columns. After premultiplying with $\mathbf{S}^{-1}$, Eq. (27) can be written as

$$\begin{cases} \mathbf{S}^{-1}\mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}\dfrac{\mathrm{d}\tilde{\mathbf{x}}(t)}{\mathrm{d}t} = -\mathbf{S}^{-1}\tilde{\mathbf{x}}(t) + -\mathbf{S}^{-1}\widetilde{\mathbf{G}}^{-1}\widetilde{\mathbf{B}}\mathbf{u}(t), \\ \mathbf{i}(t) = \widetilde{\mathbf{B}}^{\mathrm{T}}\mathbf{S}\mathbf{S}^{-1}\tilde{\mathbf{x}}(t), \end{cases} \tag{28}$$

or, if a change of variables $\mathbf{S}^{-1}\tilde{\mathbf{x}} \longrightarrow \tilde{\mathbf{x}}$ (using $\tilde{\mathbf{x}}$ again for ease of notation) is assumeed, as

$$\begin{cases} \mathbf{\Lambda}\dfrac{\mathrm{d}\tilde{\mathbf{x}}(t)}{\mathrm{d}t} = -\mathbf{I}\tilde{\mathbf{x}}(t) + \mathbf{H}\mathbf{u}(t), \\ \mathbf{i}(t) = \mathbf{E}^{\mathrm{T}}\tilde{\mathbf{x}}(t), \end{cases} \tag{29}$$

where $\mathbf{H} = \mathbf{S}^{-1}\widetilde{\mathbf{G}}^{-1}\widetilde{\mathbf{B}}$, $\mathbf{E} = \mathbf{S}^{\mathbf{T}}\widetilde{\mathbf{B}}$, and $\mathbf{I}$ is the $q \times q$ unity matrix. Note that Eq. (29) has the same dimensions as Eq. (27), but the coefficient matrices $\mathbf{\Lambda}$ and $\mathbf{I}$ are now diagonal. For the forthcoming treatment of complex eigenvalues, the m:th row in the first equation of Eq. (29) is written as

$$\Lambda_m\frac{\mathrm{d}\tilde{x}_m(t)}{\mathrm{d}t} = -\tilde{x}_m(t) + \sum_{j=1}^{N} H_{mj}u_j(t), \tag{30}$$

Some of the eigenvalues of the real matrix $\mathbf{G}^{-1}\widetilde{\mathbf{C}}$ may be complex numbers, in which case they appear in complex-conjugate pairs. Assuming that q_r of the eigenvalues are real and the rest appear in q_c conjugate pairs such that $q = q_r + 2q_c$.

Consider one such pair, $\Lambda_m^r \pm j\Lambda_m^i$. The corresponding eigenvectors, and therefore, also the corresponding rows of matrices $\mathbf{H}$ and $\mathbf{E}$ in Eq. (29) are complex

conjugate. Let the corresponding elements of vector $\tilde{\mathbf{x}}$ be $\tilde{x}_m^r \pm j\tilde{x}_m^r$. Inserting these into Eq. (30), and requiring the real and imaginary parts of the equation to hold independently, yields (the same pair of equations is obtained twice).

$$\begin{cases} \Lambda_m^r \dfrac{\mathrm{d}\tilde{x}_m^r(t)}{\mathrm{d}t} = -\tilde{x}_m^r(t) + \Lambda_m^i \dfrac{\mathrm{d}\tilde{x}_m^i(t)}{\mathrm{d}t} + \displaystyle\sum_{j=1}^{N} H_{mj}^r u_j(t), \\ \Lambda_m^r \dfrac{\mathrm{d}\tilde{x}_m^i(t)}{\mathrm{d}t} = -\tilde{x}_m^i(t) - \Lambda_m^i \dfrac{\mathrm{d}\tilde{x}_m^r(t)}{\mathrm{d}t} + \displaystyle\sum_{j=1}^{N} H_{mj}^i u_j(t), \end{cases} \tag{31}$$

As discussed before, the fastest equivalent circuit realization is Matsumoto's method. Thus, it was the implementation choice for the BVOR macromodel. Matsumoto's method is a realization of Eq. (30) and (31), the realization is presented in Fig. 8. The other equivalent circuit is presented in Fig. 7, but it utilizes controlled charge sources not availabe in SPICE. The nodal equations, e.g. for the circuit in Fig. 7 can be expressed as

$$\sum_{j=1}^{N} H_{mj} U_j = (s\Lambda_m + 1)\widetilde{X}_m \tag{32}$$

which is the same as Eq. (30) when $s\tilde{X}_m$ is replaced with $d\tilde{x}_m/dt$. [10]

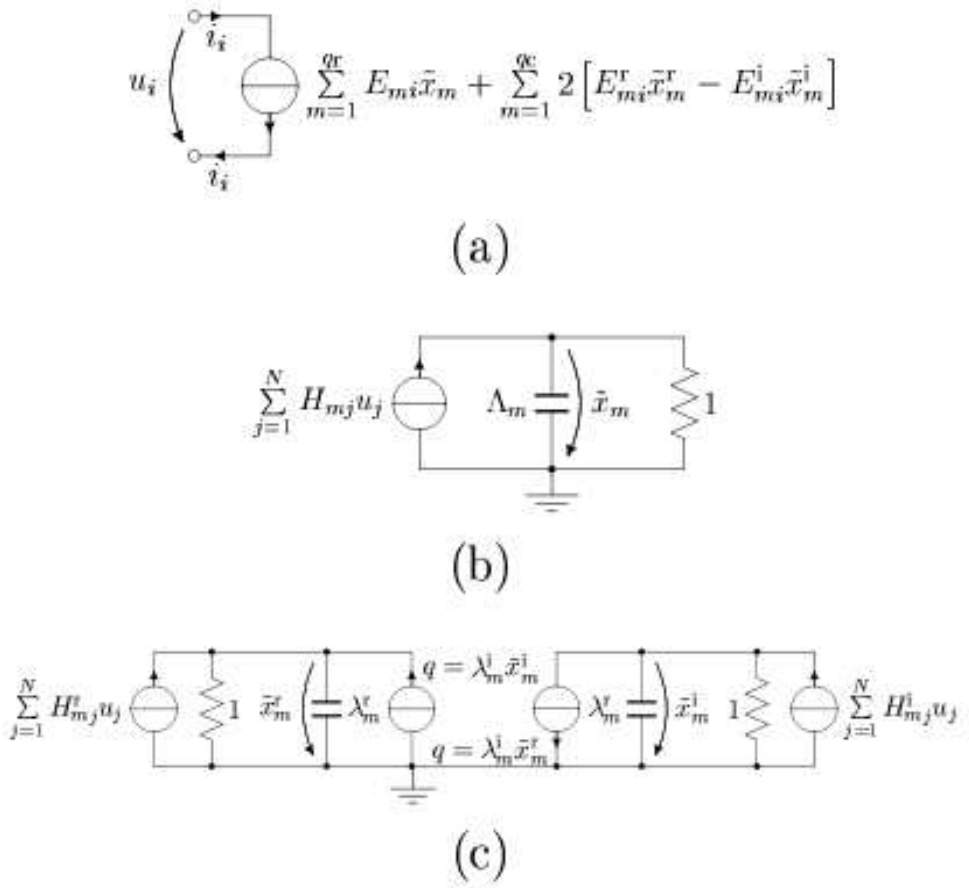

Figure 7: The Matsumoto's equivalent-circuit realization (From [1]). (a) A port VCCS, (b) realization of a real eigenvalue, and (c) realization of a complex eigenvalue pair.

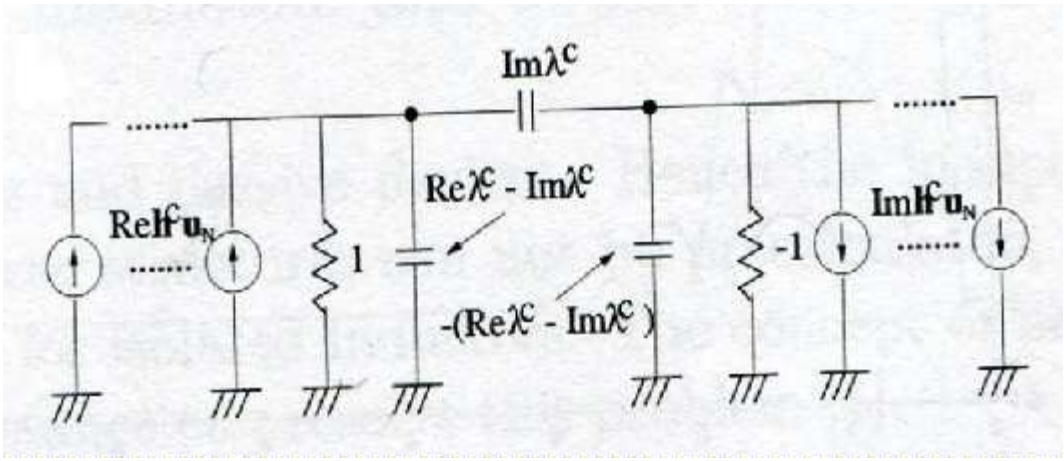

Figure 8: equivalent π circuit of Eq. (31) (From [6])

6 BVOR Implementation

This section describes the implementation of forming BBD structured VNA matrices and algorithms related BVOR method detailed in the previous sections as a netlist-in-netlist-out tool. Formation of BBD matrix and BVOR method are implemented in MATLAB code. The tool is written for SPICE netlists, which will possibly consist of non-linear elements, subcircuits and mutual inductances.

6.1 Total BVOR flow

The implemented RLC BVOR flow can be divided into the five steps listed below.

1. Circuit partitioning and netlist parsing : partition.run implemented by C partitions the linear part of the netlist into user-defined m subcircuits. Then C-tool matrix.run generates supplement files based on the partitioned subcircuits for constructing BBD-structured VNA matrices.

2. Matrix construction: MATLAB-tool constructs BBD-structured VNA matrices based on the supplement files describing the node and RLC component information of the linear part of the original netlist.

3. BVOR method: using full BBD matrix $\mathbf{G} + s_0\mathbf{C}$ to generate projection structured matrix $\mathbf{Q}$; or using diagonal blocks of the BBD matrix $\mathbf{G}_{i,i} + s_0\mathbf{C}_{i,i}$ $(i = 1, ..., m)$ to generate corresponding $\mathbf{Q}_i$,construct $\mathbf{Q}_i$ and $\mathbf{I}_0$ to $\mathbf{Q}$ in diagonal form. Generate transformed $\widetilde{\mathbf{G}}$, $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{B}}$.

4. Macromodel realization: synthesize each reduced subcircuit as a VCCS and RC macromodel.

5. Netlist reconstruction: include each macromodel in the proper position in the total netlist to achieve the original structure.

6.2 Netlist parsing and circuit partitioning

The netlist is parsed to construct graphs. The graphs and matrices are constructed for each subciruit. Circuit partitioning is based on "Using METIS and hMETIS in circuit partitioning" [8]. The partitioning algorithm constructs hypergraphs from the linear circuit netlist, partitions the graph into the specified number of partitions, maps the partitions back to netlists. The original netlist should follow a required structure, such that file structure of the mapped-back netlists can be easily realized by following program and generated supplement files can be easily utilized for further processing.

6.3 Matrix construction

Supplement files contain information of all the partitioned subcircuit. Note that individual subcircuit has common node with the main circuit only, there is no interconnection between two or more subcircuits; the main circuit has external ports with

non-linear components and sources, main circuit and individual subcircuit contain only resistor, capacitor and inductor with or without mutual inductances.

MNA matrices are generated using MATLAB code based on the information of the supplement files. Supplement files also provide information of common ports between main circuit and individual subcircuits, external ports between main circuit and non-linear components and sources, mutual inductances (if exist) between elements of subcircuits.

Then MNA matrices are converted to VNA matrices. With the information of common ports, external ports and mutual inductances (if exist), BBD-structure VNA matrices are formed.

The whole process of forming BBD-structured VNA matrix is described below:

1. After circuit partition is done, start
2. convert MNA to VNA
3. find common ports (node) between main circuit and every subcircuit
4. find external ports between main circuit and non-linear components and sources
5. find the number of mutual inductances and corresponding values in/between subcircuits, if the existence of mutual inductances is detected
6. form $\mathbf{M}_0$, $\mathbf{B}_0$
7. form $\mathbf{M}_i$, $\mathbf{X}_i$ with the feature of mutual inductances
8. form BBD-structure VNA matrix
9. end

6.4 BVOR method

Implementation follows the BVOR method enhanced with BBD solver described in previous section. Algorithms for calculating two types of $\mathbf{Q}$ (one is generated by inputing full BBD matrix and another by individual diagonal block) are implemented separately. The simulation results are illustrated in next section.

6.5 Macromodel realization and netlist reconstruction

Matsumoto's method is used for the RLC macromodel realization. The partitions of the original netlist have been reduced and a separate macromodel has been generated for each partition. The final part of the BVOR flow reconstructs a single netlist with the same hierarchical structure as in the original netlist, i.e. the macromodels generated for each subcircuit are combined together. Finally, multiple parallel resistances and capacitances to the ground are added together to further reduce the number of elements in the final BVOR netlist.

7 Simulation examples

In this section, the operation of the BVOR methods implemented and discussed in the previous sections is studied. Section 7.1 describes the test netlists used and the simulations performed. Section 7.2, 7.3, 7.4 present the results utilizing **Q** generated with full BBD matrix as input, **Q** generated with individual diagonal block as input, **Q** generated by PRIMA with MNA matrix as input and without partition, respectively.

7.1 Simulation setup

The netlists used in the simulations are shown in Table 1, where n, n_e, G, C, L and L_m mean the number of all nodes, external nodes, resistances, capacitances, inductances and mutual inductances respectively.

Table 1: Netlists used in the simulations

Name	n	n_e	G	C	L	L_m
lt.cir	1083	3	369	363	360	-
RL_chain.net	2002	3	1998	-	999	-
RCchain2.net	1002	2	1000	999	-	-
RCLKbuses3.net	4511	2	1507	1507	1502	400
sub_large_tran.net	1083	3	363	369	360	1

The netlist lt.cir is RLC netlist, RL_chain.net is RL-only netlist and RCchain2.net is RC-only netlist. The inductances in lt.cir and RL_chain.net do not contain mutual inductances. On the other hand, both RCLKbuses3.net [1] and sub_large_tran.net are RLC netlist with mutual inductance feature.

It should be noticed that BVOR tool performs the reduction only on the RLC blocks of the netlist. The non-linear components of the netlist should be removed before the actual reduction procedure starts.

The simulations were performed in a way that the circuit was firstly partitioned and then reduced with the user-defined BVOR method. All the simulations were done on a RAM1.5G/2.01GHz computer.

Some of the simulations are compared with those obtained with existing MOR tool, specifically, PRIMA method. The algorithm of PRIMA has been already introduced in previous sections.

In order to test the partitioning with Algorithm 3 by utilizing **Q** generated with full BBD matrix as input, the target netlist was divided with several partitionings and several orders of reduction. The original/reduced circuit was run using AC analysis with specified frequency sweeps. In addition, y parameters are calculated using following equation:

$$Y = \mathbf{B}^T(\mathbf{G} + j2\pi f\mathbf{C})^{-1}\mathbf{B}; \tag{33}$$

Simulation results in this section are obtained using Eq. (33) implemented by Matlab *inv*() function. If *LU* factorization is used, the simulation results are more realistic and faster; however speed up of the reduction is worse.

7.2 Q generated with full BBD matrix as input

In this section, circuits are reduced by using **Q** generated with full BBD matrix. This alternative is efficient since BBD solver (Algorithm 2) is designed especially for BBD-structured state matrix to reduce the computational cost. In the following, RLC, RC and RL netlists are reduced with BVOR method.

7.2.1 RC netlist

The obtained results are listed in Table 2 and 3, where n_p, Q_{col}, s_0, N_{red}, $Cond.$, N_{nz}, N_{pp}, $E_{y_{ii}}$ and t mean number of partitions, order of reduction, expansion point, size of $\widetilde{\mathbf{G}}+s_0\widetilde{\mathbf{C}}$, condition number of $\widetilde{\mathbf{G}}$, number of non-zero elements in $\widetilde{\mathbf{G}}+s_0\widetilde{\mathbf{C}}$, positive poles, average error of total y paramerters ($\frac{1}{n}\sum_{i=0}^{n}\triangle y_i$, $\triangle y_i$ is the subtraction between original y parameter and reduced y parameter), and computational time of calculating a certain y parameter respectively. q here refers to the order of reduction. q and port number N determines the number of iterations of Block Arnoldi method: $N_q = \lfloor q/N \rfloor$.

The reduced system should be eigenvalue decomposed before the Matsumoto macromodel realization procedure and decomposition may generate positive poles which lead to unstable system, therefore the number of positive poles is recorded in the testing phase.

Table 2: Testing result of RCchain2.net, frequency range: 100k∼1G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
4	ori.	-	-	1018	-	-	3042	-	56.83
	6	40	100e6	40	4.9669e14	2	831	2.6028e-4	0.03
	16	87	100e6	87	1.4338e5	3	3679	3.1255e-16	1.39

Table 3: Testing result of RCchain2.net, frequency range: 10∼1000G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
4	ori.	-	-	1018	-	-	3042	-	56.14
	6	40	100e6	40	4.9669e14	2	831	1.4254e-6	0.03

Figure 9 a, b and c indicate the y_{11} parameter as a function of frequency. Plain curves represent the reduced circuits and diamond curves represent original circuits. According to the figure, even with small value of q which results in small reduction order, reduced and original systems can be overlapped well, therefore it is not necessary to continue the test with larger q and we could conclude that BVOR tool succeeded in RC-only circuit.

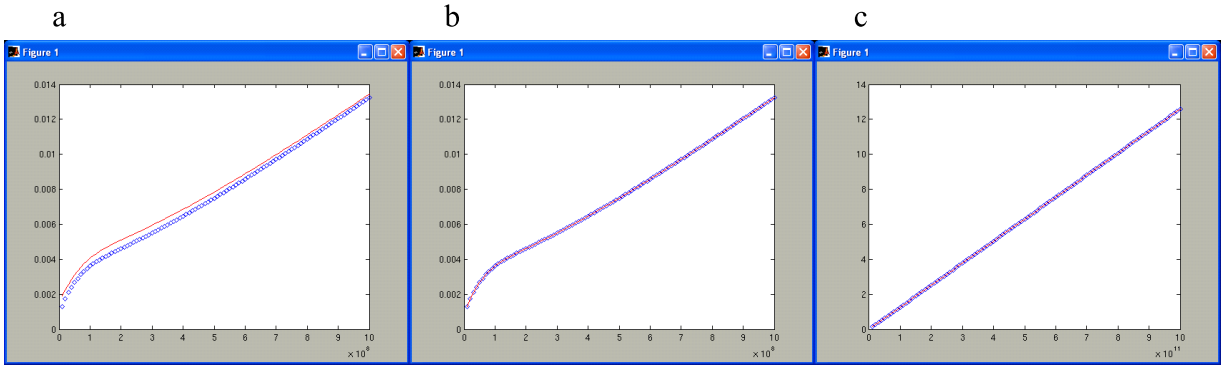

Figure 9: y_{11} parameter as a function of frequency of netlist RCchain2.net. a, b are at n_p=3 and q=6, q=16 with frequency range 0-40G. c is at n_p=3 and q=6 with frequency range 10-1000G

Table 4 shows that when expansion point s_0 is 100e6, system error dereases dramatically. Therefore expansion point at 100e6 is suitable for the circuit RC-chain2.net.

Table 4: Expansion point: s_0 at n_p=4 and q=16, testing result of RCchain2.net, frequency range: 100k∼1G, number of sampling points: 100

s_0	$Cond.$	$E_{y_{ii}}$
0	1.6181e5	0.2637
1000	3.8327e5	0.2151
100e6	1.4338e5	5.1293e-15
100e8	2.6619e5	5.1439e-15

7.2.2 RL netlist

Further tests with RL-only netlist RL_chain.net is illustrated in Table 5.

With original un-reduced BBD matrix, the computational time for a certain y parameter is relatively large as we can see from Table 5. Note that the conditional number and positive poles are also large even we tried to fix the problem by adding very small conductances to the diagonal. This mechanism should help with the numerical instablilty in most cases, but based on the result, it is not as we expected for RL-netlist.

Table 5: Testing result of RL_chain.net, frequency range: 100∼40G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
9	ori.	-	-	2040	-	-	8094	-	170.96
	6	90	1000	90	2.4866e20	36	2268	0.0232	1.51
	15	180	1000	180	1.1272e27	59	9072	0.0191	5.78
	25	300	1000	300	6.2391e28	67	25199	0.0238	18.07
	30	328	1000	328	2.4866e20	85	29175	0.0178	21.81
	40	436	1000	436	2.4866e20	93	44295	0.0177	43.02
4	ori.	-	-	2018	-	-	8039	-	161.9
	6	45	1000	45	5.2651e30	18	1052	0.0116	0.37
	16	100	1000	100	6.5050e27	28	4705	0.0111	1.81
	26	136	1000	136	1.5114e27	35	7693	0.013	3.48
	36	172	1000	172	4.2716e27	47	11329	0.0133	5.56
	46	220	1000	220	1.5406e27	62	17185	0.0133	10.98
	96	412	1000	412	1.9748e27	98	52129	0.0136	47.88
7	ori.	-	-	2032	-	-	8074	-	164.14
	6	72	1000	72	2.4019e20	27	1781	1.5046	0.95
	15	144	1000	144	2.4296e28	43	7127	0.9797	3.76
	25	237	1000	237	2.4019e20	61	18368	1.0151	11.09
	35	300	1000	300	2.4019e20	85	26117	1.0027	18.340
	40	342	1000	342	2.4019e20	80	31913	0.9928	24.73

The results also indicate that errors and computational time of certain y parameters of reduced system are not significantly relevant with partitioning number, contrarily computational time of original system decreases when partitioning number becomes larger. It is apparent to find out that the suitable order of reduction for the system is about 140. For instance, when circuit is partitioned into 4 subcircuits and the order reduction is 136, the average error of total y parameters is at minimum and simultaneously computational time for a certain y parameter is relatively small.

Table 6 indicates that when expansion point s_0 is 1e8, system error is at minimum level. Therefore expansion point 1e8 is suitable for the circuit RL_chain.net.

7.2.3 RLC netlist

This section presents two sets of testing results. First set (Table 7) is obtained by performing BVOR method on RLC netlist which does not contain mutual inductances. Second set (Table 9 and 10) is achieved from RLC netlists with mutual inductance feature.

Results in Table 7 show that BVOR tool works fine on lt.cir with different partitioning numbers. The increasing value of order reduction results in decreasing error. It is clear that suitable order reduction is around 180. Relative small error and

Table 6: Expansion point: s_0 at n_p=4 and q=36, testing result of RL_chain.net, frequency range: 100∼40G, number of sampling points: 100

s_0	$Cond.$	$E_{y_{ii}}$
0	4.3040e27	151.9560
10	1.4012e20	46.5541
1000	4.2716e27	0.0133
1e5	1.4011e20	3.4036e-04
1e8	1.0592e21	2.2570e-08

computational time are achieved when order reduction is near this value. It should be noted that large order reduction (546 in Table 7 when $n_p = 5$ and $q = 100$) causes even more complicated reduced system, the computational time is heavier than the original system, hence it is not necessary to use large order reduction.

Table 8 shows that different expansion point s_0 does not affect system errors significantly. Since smaller expansion points lead to smaller conditional number, smaller expansion points are suitable for the circuit lt.cir.

Figure 10 shows the entire reduction time when n_p=3 (q=40), n_p=4 (q=30), n_p=5 (q=20) respectively. The result depicts that the increase of partition size results in the decrease of the reduction time.

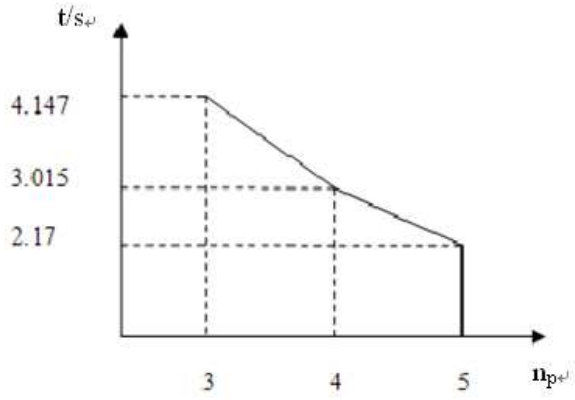

Figure 10: the entire reduction time when n_p=3 (q=40), n_p=4 (q=30), n_p=5 (q=20) respectively

Example circuit lt.cir does not contain mutual inductance feature. In reality, complicated situation such as large conditional number and instablility are resulted from mutual inductance(s). Following two RLC netlists contain mutual inductance(s).

After performing BVOR method on RLCKbuses3.net, results are in Table 9. The increasing of order reduction results in the increasing of computational time. The original computational time is relatively large probably because the circuit contains

Table 7: Testing result of lt.cir, frequency range: 0∼40G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s	$tredu/s$
3	ori.	-	-	1097	-	-	3279	-	79.76	-
	10	60	1000	60	9.0714e6	4	2250	0.0486	0.34	2.927
	20	87	1000	87	1.8046e7	3	4113	0.0333	2.86	2.727
	30	114	1000	114	7.5574e6	2	6462	0.0342	2.15	3.607
	40	150	1000	150	9.3906e6	2	10350	0.0222	3.89	4.147
	50	177	1000	177	6.4004e7	3	13833	0.0157	7.21	5.347
	80	267	1000	267	4.0658e7	1	28953	0.0134	16.18	6.657
	100	330	1000	330	8.9743e7	3	42750	0.0088	26.52	9.727
4	ori.	-	-	1101	-	-	3289	-	80.44	-
	10	75	1000	75	9.1309e18	1	2925	0.0462	1.09	2.745
	20	114	1000	114	3.6465e6	1	6084	0.0274	2.22	2.935
	30	150	1000	150	4.0343e6	3	9432	0.0207	3.96	3.015
	40	198	1000	198	2.1165e7	1	14904	0.0150	7.28	2.915
	50	234	1000	234	3.0136e7	3	19764	0.0134	10.27	4.305
	80	354	1000	354	7.3517e7	4	40644	0.0065	26.2	5.655
	100	438	1000	438	1.3887e8	5	59544	0.0034	42.83	6.415
5	ori.	-	-	1105	-	-	3299	-	80.94	-
	10	90	1000	90	1.1737e15	5	3600	0.0415	1.56	2.839
	20	141	1000	141	8.1578e6	5	8361	0.0266	3.76	2.179
	30	186	1000	186	2.5754e7	3	12816	0.021	8.65	2.729
	40	246	1000	246	8.2467e6	2	20016	0.0147	13.46	5.039
	50	291	1000	291	3.9266e7	5	26361	0.0125	19.41	6.249
	80	441	1000	441	7.2477e9	2	53361	0.0066	64.130	7.859
	100	546	1000	546	8.9266e13	18	77616	0.0038	95.98	9.529

Table 8: Expansion point: s_0 at n_p=5 and q=50, testing result of lt.cir, frequency range: 0∼40G, number of sampling points: 100

s_0	$Cond.$	$E_{y_{ii}}$
0	3.8941e7	0.0128
10	9.1629e7	0.0121
1000	3.9266e7	0.0125
1e5	1.5647e8	0.0135
1e8	9.5237e11	0.0122

large set of mutual inductances. BVOR method significantly reduces the time as we can see from the results of the table. Therefore BVOR method is suitable for circuit

containing large set of mutual inductances.

Table 9: Testing result of RLCKbuses3.net, frequency range: 0∼40G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
5	ori.	-	-	4927	-	-	17165	-	2325.83
	10	72	1000	72	2.9955e18	21	2304	0.0246	1.67
	40	252	1000	252	5.0433e19	84	28224	0.0096	15.23
	80	492	1000	492	8.2582e19	190	107584	0.011	66.84

RLC circuit sub_large_tran.net is a simpler case since it has only one mutual inductance. Table 10 shows the result after performing BVOR tool on this circuit, the computational time of original system is much less than the one in previous case.

Table 10: Testing result of sub_large_tran.net, frequency range: 0∼40G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
3	ori.	-	-	1098	-	-	3288	-	83.48
	10	60	1000	60	3.2146e24	20	2250	0.0482	0.08
	40	151	1000	151	4.5835e16	37	10651	0.0381	4.31
	80	268	1000	268	1.1998e17	68	29488	0.034	14.82

Figure 11 depicts the the curves of parameter y_{11} as a function of frequency when $q = 80$ and frequency range is from 0 to 10G Hz. Note that diamond curve represents the original system, and plain curver represents the reduced system.

With relatively narrow frequency ragne, from 0 to 2GHz, as shown in Fig. 12, better results can be achieved.

Table 11 shows when expansion point s_0 increases to 1e8, system error decreases dramatically. Therefore s_0=1e8 is suitable expansion point for circuit sub_large_tran.net.

Table 11: Expansion point: s_0 at n_p=5 and q=80, testing result of sub_large.net, frequency range: 0∼10G, number of sampling points: 100

s_0	$Cond.$	$E_{y_{ii}}$
0	2.3242e20	0.0362
10	1.3990e15	0.0461
1000	1.5234e17	0.0254
1e5	3.9727e19	0.1391
1e8	5.9358e21	2.8929e-04

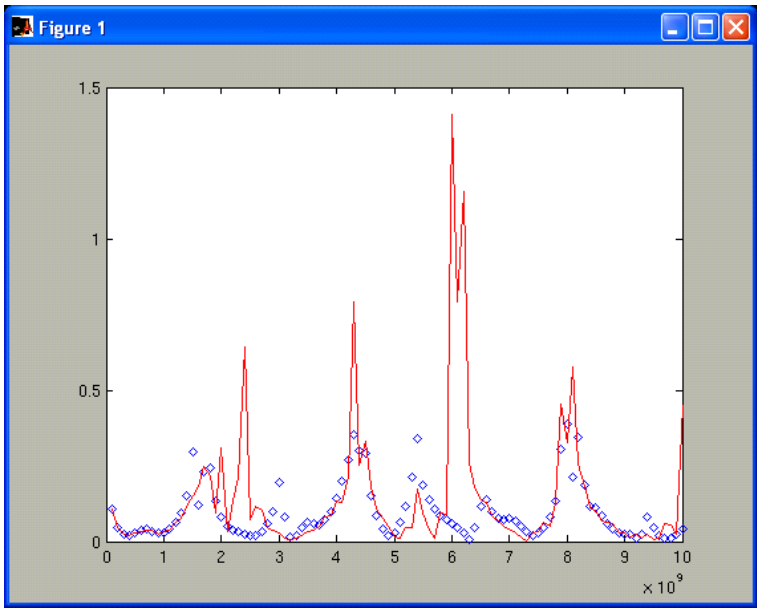

Figure 11: y_{11} parameter as a function of frequency of netlist sub_large_tran.net with frequency range 0-10G at n_p=3 and q=80

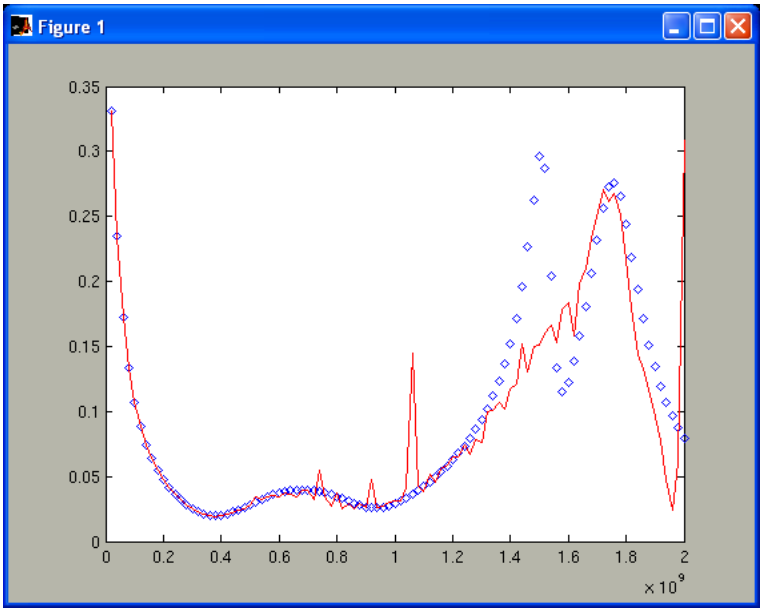

Figure 12: y_{11} parameter as a function of frequency of netlist sub_large_tran.net with frequency range 0-2G at n_p=3 and q=80

7.3 Q generated with individual diagonal block as input

In this section, RC netlist RCchain2.net, RL netlist RL_chain.net, and RLC netlist lt.cir are used for testing. Instead of using full BBD system to generate **Q**, individual diagonal block is used.

Testing result of RLC netlist lt.cir is presented in this section. Table 12 indicates

that since conditional number of $\widetilde{\mathbf{G}}$ is infinite when q is 20, it is not possible to get negative poles, additionally, it is meaningless to continue the testing with larger q. Other types of circuits indicate similar simulation results but not shown here. Therefore it is concluded that $\mathbf{Q}$ generated with individual diagonal block is worse than from full BBD system.

Table 12: Q generated with individual diagonal block, testing result of lt.cir, frequency range: 0∼40G, number of sampling points: 100

n_p	q	Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
3	ori.	-	-	1097	-	-	3279	-	79.76
	10	40	1000	40	8.9492e18	2	355	3.0352	0.33
	20	80	1000	87	Inf	error	1505	NaN	0.68
4	ori.	-	-	1101	-	-	3289	-	80.44
	10	50	1000	50	1.0684e19	1	434	7.9884	0.46
	20	100	1000	100	Inf	error	1990	NaN	1.04

7.4 Q generated by PRIMA with MNA matrix as input and without partition

It should be noted that tables in this section does not include n_p and q because PRIMA tool uses full MNA matrix as input, i.e. partitioning was not carried out .

By comparing Table 13, 14 with Table 2, 3 of Section 7.2, it is convinceable that for RC-only circuit PRIMA method with full MNA matrix produces more reliable result than BVOR method and the computing time of both methods is approximately the same.

Table 13: PRIMA Testing result of RCchain2.net, frequency range: 100k∼1G, number of sampling points: 100

Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
ori.	-	1002	-	-	-	-	47.68
40	100e6	40	8.8104e4	3	1600	1.8690e-14	0.03
87	100e6	87	9.9975e4	1	7569	2.2262e-14	3.66

Base on the outcome in Table 15 of this section and Table 5 of Section 7.2, we can see that PRIMA with full MNA produces smaller conditional number and less positive poles resulting in much smaller errors than BVOR method; at the same time, it consumes almost the same amount of time as BVOR does. Therefore it is apparent that PRIMA with full MNA is more suitale for RL circuit.

According to the result of Table 16 of this section and Table 7 of Section 7.2, PRIMA with full MNA is better choice than BVOR method for RLC circuit since

Table 14: PRIMA testing result of RCchain2.net, frequency range: 10∼1000G, number of sampling points: 100

Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
ori.	-	1002	-	-	-	-	46.91
40	100e6	40	8.8104e4	3	1600	5.3847e-12	0.03

Table 15: PRIMA testing result of RL_chain.net, frequency range: 100∼40G, number of sampling points: 100

Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
ori.	-	2002	-	-	-	-	218.16
72	1000	72	5.4557e8	7	5184	1.6212e-10	0.92
144	1000	144	1.0251e9	20	20736	1.0970e-10	3.16
237	1000	237	1.9330e9	25	56169	4.6570e-11	12.73
300	1000	300	2.3367e9	38	90000	5.0901e-11	28.03
342	1000	342	2.5263e9	48	116964	1.2559e-10	24.62

the former causes less errors and comsumes less time than latter at the same level of order reduction.

Table 16: PRIMA testing result of lt.cir, frequency range: 0∼40G, number of sampling points: 100

Q_{col}	s_0	N_{red}	$Cond.$	N_{pp}	N_{nz}	$E_{y_{ii}}$	t/s
ori.	-	1083	-	-	-	-	60.76
60	1000	60	7.3392e4	0	3600	0.0227	0.07
87	1000	87	8.1696e8	0	7569	0.0219	1.29
114	1000	114	6.9434e7	0	12996	0.0164	2.2
150	1000	150	8.2439e9	0	22500	0.0133	4.22
177	1000	177	6.9278e14	58	31329	0.0088	5.93
267	1000	267	2.3777e18	1	71289	0.001	15.43
330	1000	330	6.014e10	4	108900	5.9925e-4	23.95

Based on previous comparisons, it is concluded that PRIMA method with full MNA provides more reliable model-order reduction and faster speed than BVOR method for RL(C) circuits. However, it is still necessary to use BVOR method for large-scale circuits, and it is BVOR method that can decompose circuits into sub-circuits which contains mutual inductances.

7.5 Discussion of simulation results

The main results are listed below:

- BVOR was successful; BVOR tool is applied on RC, RL, and RLC circuit, the simulation results indicate that with the appropriate selection of order reduction, computing time of reduced system is faster;

- PRIMA shows better results than BVOR if partitioning is not needed;

- $\mathbf{Q}$ generated with individual diagonal block is worse than from full BBD system;

- macromodel realization using "Matsumoto" on BBD matrix was not successful. The reason is: branch tearing method results in new line which is linearly dependent due to Eq. (23) and the auxiliary state variable ϕ_{ij}, hence the BBD-structured VNA matrix $\mathbf{G}$ is singular; $\mathbf{Q}^T\mathbf{G}\mathbf{Q}$ does not change the conditional property of $\mathbf{G}$, therefore the transformed $\widetilde{\mathbf{G}}$ is also singular.

8 Summary

A stand-alone netlist-in-netlist-out MATLAB/C tool was implemented for BVOR method. The performance of the BVOR method was evaluated with extensive simulations. It was found that BBD-based Arnoldi method with full BBD matrix as input is the better choice for most of the circuits than the corresponding method with individual diagonal blocks as input since the latter method causes errors and interruption to order reduction process.

Benefit of BBD structure is obvious. Each of the divided small block is solved at a time, which saves memory. In addition, solving these blocks in parallel is efficient.

The algorithm of BBD solver further improves the reduction results in many cases and in general speeds up the process.

Macromodel realization "Matsumoto" on the system reduced by BVOR method failed, the final reconstructed netlist does not have the same functionality as the original netlist. In further investigation, another method RLCSYN [23] (RLC Equivalent Circuit Synthesis for Structure-Preserved Reduced-order Model of Interconnect) may present convincible outcome.

In a word, the implemented BVOR tool provides the user with BVOR algorithm enhanced by BBD solver suitable for most types of circuits. The BVOR achieved excellent results with reduction of over 80% in the number of elements and simulation error well below 3%. With the appropriate selection of order reduction, computing time of reduced system is 10 to 100 times faster; on the other hand, inappropriate order reduction could cause even lower computing time.

References

[1] T.Palenius, and J. Roos, "Comparison of Reduced-Order Interconnect Macromodels for Time-Domain Simulation",*IEEE Transactions on Microwave Thoery and Techniques*, vol. 52, no. 9, 09.2004, pp. 2240-2250.

[2] K.J.Kerns and A. T. Yang, "Preservation of passivity during RLC network reduction via split congruence transformations," *IEEE Trans. Comput.-Aided Des.Integr.Circuits Syst.*, vol.17,no.7,pp.582-591,Jul.1998.

[3] R. W. Freund, "SPRIM: Structure-preserving reduced-order inter-connect macro-modeling," in *Proc. Int. Conf. Comput. Aided Des. (ICCAD)*, 2004,pp.80-87.

[4] C. W. Ho, A. E. Ruehli, and P. A. Brennan, "The modified nodal approach to network analysis," *IEEE Trans. Circuits Syst.*, vol. 22, no. CAS-6, pp. 504-509, Jun. 1975.

[5] Hao Yu, Chunta Chu, Yiyu Shi, "Fast Analysis of a Large-Scale Inductive Interconnect by Block-Structure-Preserved Macromodeling" *IEEE Trans. On Very Large Scale Integration (VLSI) Syst.*, vol. 18, no. 10, pp. 1399-1411, Oct. 2010.

[6] Yuya Matsumoto, Yuichi Tanji and Mamoru Tanaka, "Efficient SPICE-Netlist Representation of Reduced-Order Interconnect Model" *ECCTD'01*, Espoo, August 28-31, 2001, pp. II-145 - II-148.

[7] Altan Odabasioglu, Mustafa Celik, "PRIMA:Passive Reduced-Order Interconnect Macromodeling Algorithm", *IEEE transactions on Computer-Aided design of Integrated Circuits and Systems*, vol. 17, no. 88, pp. 645-654, August, 1998.

[8] P. Miettinen, M. Honkala, J. Roos, "Using METIS and hMETIS Algorithms in Circuit Partitioning",*Circuit Theory Laboratory Report Series*, CT-49, 2006, Espoo.

[9] Hao Yu, Chunta Chu, Yiyu Shi, "A Fast Block Structure Preserving Model Order Reduction for Inverse Inductance Circuits", *Proceedings of the 2006 IEEE/ACM International Conference on Computer-Aided Design*, ICCAD'06, pp.7-12, 5-9 Nov. 2006.

[10] P. Miettinen, "Hierarchical Model-Order Reduction Tool for RLC Circuits", *Master Thesis*, 49p, Helsinki University of Technology, Espoo, 2007.

[11] P. Miettinen, "Partitioning and Macromodel-Based Model-Order Reduction for RLC Circuits", *Licentiate Thesis*, 29p, Aalto University, Espoo, 2010.

[12] P. Miettinen, M. Honkala, J. Roos, and M. Valtonen, "Partitioning-Based Reduced of Circuits with Mutual Inductances", in *Proceedings of Scientific Computing in Electrical Engineering 2010*, 8 p. accepted to be published.

[13] G. Shi and C.J. R. Shi, "Model order reduction by dominant subspace projection: Error bounds, subspace approximation and circuit applications", in *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 52, no. 5, pp. 975-993, May 2005.

[14] T. Palenius and J. Roos, "Development and comparison of reduced-order interconnect macromodels for time-domain simulation", in *Proc. ICECS'02*, vol. 2, Dubrovnik, Croatia, 2002, pp. 757-760.

[15] *APLAC – Circuit Simulation and Design Tool* , Version 8.3 Manuals, AWR-APLAC Corporation, Finland, 2007.

[16] Yu-Min Lee, Yahong Cao, Tsung-Hao Chen and Janet Meiling Wang, "HiPRIME: Hierarchical and Passivity Preserved Interconnect Macromodeling Engine for RLKC Power Delivery" *IEEE Trans. Comput.-Aided Des.Integr.Circuits Syst.*, vol.24, no.6,pp.797-806,June,2005.

[17] Dou Li, Sheldon X.-D. Tan, Lifeng Wu, "Hierarchical Krylov subspace based reduction of large interconnects" *INTEGRATION, the VLSI journal*, no.42, pp.193-202, 2009.

[18] P. Miettinen, M. Honkala, J. Roos, and M. Valtonen, "PartMOR: Partitioning-Based Realizable Model-Order Reduction Method for RLC Circuits", in *IEEE Trans. Comput.-Aided Des.Integr.Circuits Syst.*, vol.30, no.3, pp.374-387,March,2011.

[19] P. Miettinen, M. Honkala, J. Roos, and M. Valtonen, "PartMOR: Partitioning-Based Realizable Model-Order Reduction Method for RLC Circuits", in *IEEE Trans. Comput.-Aided Des.Integr.Circuits Syst.*, vol.30, no.3, pp.374-387,March,2011.

[20] H. Liao and W. W.-M. Dai, "Partitioning and reduction of RC interconnect networks based on scattering parameter macromodels", in *Digest of Technical Papers of IEEE/ACM International Conference on Computer Aided Design*, pp.704-709, 1995.

[21] B. N. Sheehan, "TICER: Realizable Reduction of Extracted RC Circuits", in *Proceedings of the 1999 IEEE/ACM International Conference on Computer-Aided Design*,San Hose, CA, 1999, pp. 200-203.

[22] S. Lee and K. Cho, "RCC: An RC Compression System for Interconnect Circuits", in *Journal of the Korean Physical Sociaty*, vol. 41, no. 6, pp. 982-987, Dec. 2002.

[23] Fan Yang, Xuan Zeng, Yangfeng Su, and Dian Zhou, "RLCSYN: RLC Equivalent Circuit Synthesis for Structure-Preserved Reduced-order Model of Interconnect", in *Proc. ISCAS*, pp. 2710-2713, May, 2007.