UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Model Order Reduction of Electromagnetic Models for Large Integrated Circuits

Diploma Thesis

Giamouzis Christos

Panagiotou Dimitra

Supervisor: Evmorfopoulos Nestor

June 2022

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Model Order Reduction of Electromagnetic Models for Large Integrated Circuits

Diploma Thesis

Giamouzis Christos

Panagiotou Dimitra

Supervisor: Evmorfopoulos Nestor

June 2022

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μείωση Τάξης Μεγέθους Ηλεκτρομαγνητικών Μοντέλων για Μεγάλα Ολοκληρωμένα Κυκλώματα

Διπλωματική Εργασία

Γιαμούζης Χρήστος

Παναγιώτου Δήμητρα

Επιβλέπων/πουσα: Ευμορφόπουλος Νέστωρ

Ιούνιος 2022

Approved by the Examination Committee:

Supervisor **Evmorfopoulos Nestor**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Tsompanopoulou Panagiota**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Stamoulis George**
Professor, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

First and foremost, we would like to express our appreciation to our supervisor, Prof. Nestor Evmorfopoulos for the support throughout the development of the thesis and the chance to participate in a remarkable research project in collaboration with ANSYS, Inc.

Besides our supervisor, we would like to express our gratitude and thanks to our team leader Dr. Dimitrios Garyfallou, who guided us along the journey of our research study.

We also want to to thank Dr. George Floros and Ph.D. candidate Olympia Axelou who were ready and willing to offer their help whenever needed.

Finally, our deepest gratitude goes to our friends and family for their love and support in order to make this thesis possible.

This research has been co-financed by the European Regional Development Fund and Greek national funds via the Operational Program ”Competitiveness, Entrepreneurship and Innovation”, under the call ”RESEARCH-CREATE-INNOVATE” (project code: T2EDK-00609).

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarants

Giamouzis Christos and Panagiotou Dimitra

Diploma Thesis

Model Order Reduction of Electromagnetic Models for Large Integrated Circuits

Giamouzis Christos

Panagiotou Dimitra

Abstract

The increasing demand for accurate and inexpensive simulation of modern IC subsystems constitutes a great challenge for the EDA industry. The simulation of such systems requires solving large systems of equations with several millions or, in some cases, billions of units. Model Order Reduction (MOR) techniques form a solution to the aforementioned problem since they reduce the computational complexity of large mathematical models by replacing the original models with reduced models. The new produced reduced models approximate the behavior of the original models at the input/output ports but with significantly smaller internal dimensions.

MOR methods are divided in two large categories, the Moment Matching (MM) techniques and the system theoretic techniques. MM techniques are known for their computational efficiency, however they do not provide an a-priori error concerning the accuracy of the produced reduced model. System theoretic techniques, like the Balanced Truncation (BT) method, on the other hand, offer a reliable bound for the approximation error of the method before the computation of the reduced model. Nevertheless, the BT algorithm involves expensive computations for the solution of the Lyapunov equation and considerable storage demands for dense matrices of large models produced by solving the Lyapunov equations.

In this thesis, we present a new BT approach that deals with the above computational and storaging limitations. The proposed approach implements an efficient low-rank solver for the Lyapunov equation that uses the Extended Krylov Subspace (EKS) method, which handles large electromagnetic models with greater accuracy compared to the default BT algorithm.

Keywords:

Model Order Reduction, Balanced Truncation, Krylov Subspace

Διπλωματική Εργασία

Μείωση Τάξης Μεγέθους Ηλεκτρομαγνητικών Μοντέλων για Μεγάλα Ολοκληρωμένα Κυκλώματα

Γιαμούζης Χρήστος

Παναγιώτου Δήμητρα

Περίληψη

Η αυξανόμενη ζήτηση για υπολογιστικά φθηνές προσομοιώσεις με ακριβή αποτελέσματα σε σύγχρονα ολοκληρωμένα κυκλώματα αποτελεί μεγάλη πρόκληση για τη βιομηχανία ημιαγωγών. Η προσομοίωση τέτοιων κυκλωμάτων απαιτεί την επίλυση μεγάλων συστημάτων εξισώσεων με πολλά εκατομμύρια ή, σε ορισμένες περιπτώσεις, δισεκατομμύρια μονάδες. Οι τεχνικές υποβιβασμού τάξης μοντέλου (Model Order Reduction - MOR) αποτελούν μια λύση στο προαναφερθέν πρόβλημα, αφού μειώνουν την υπολογιστική πολυπλοκότητα μεγάλων μαθηματικών μοντέλων, ενώ παράλληλα τα νέα μοντέλα διατηρούν παρόμοια συμπεριφορά με τα αρχικά μοντέλα στις θύρες εισόδου/εξόδου.

Οι τεχνικές MOR διακρίνονται σε δύο μεγάλες κατηγορίες, τις τεχνικές Moment Matching (MM) και τις θεωρητικές τεχνικές συστήματος. Οι τεχνικές MM είναι γνωστές για την υπολογιστική τους απόδοση, ωστόσο δεν παρέχουν την δυνατότητα υπολογισμού σφάλματος, αναφορικά με την ακρίβεια του παραγόμενου μειωμένου μοντέλου, πριν από τον υπολογισμό του. Οι θεωρητικές τεχνικές συστημάτων, από την άλλη πλευρά, όπως η μέθοδος εξισορρόπησης και αποκοπής (Balanced Truncation - BT) προσφέρουν αξιόπιστο εύρος διακύμανσης του σφάλματος της μεθόδου πριν από τον υπολογισμό του μειωμένου μοντέλου. Παρόλα αυτά, η μέθοδος BT απαιτεί ακριβούς υπολογισμούς για τη λύση των Lyapunov εξισώσεων ενώ έχει σημαντικές απαιτήσεις μνήμης αφού αποθηκεύει πυκνούς πίνακες μεγάλων ηλεκτρομαγνητικών μοντέλων που προκύπτουν από την επίλυση αυτών των εξισώσεων.

Σε αυτή τη διπλωματική εργασία, παρουσιάζουμε μία προσέγγιση BT που διαχειρίζεται τις παραπάνω υπολογιστικές και αποθηκευτικές ανεπάρκειες. Η προτεινόμενη προσέγγιση υλοποιεί έναν αποτελεσματικό μηχανισμό για την επίλυση των Lyapunov εξισώσεων που χρησιμοποιεί τον υπόχωρο Krylov (Extended Krylov Subspace - EKS) με σκοπό να χειριστεί μεγάλα ηλεκτρομαγνητικά μοντέλα ταχύτερα και με μεγαλύτερη ακρίβεια σε σύγκριση με

τον αρχικό αλγόριθμο BT.

Λέξεις-κλειδιά:

Υποβιβασμός Τάξης Μοντέλου, Μέθοδος Εξισορρόπησης και Αποκοπής

Table of contents

List of figures

List of tables

Abbreviations

EDA	Electronic Design Automation
IC	Integrated Circuit
ROM	Reduced Order Model
CAD	Computer Aided Design
MOR	Model Order Reduction
MM	Moment Matching
BT	Balanced Truncation
MNA	Modified Nodal Analysis
IVP	Initial Value Problem
BE	Backward Euler
TR	Trapezoidal
HSV	Hankel Singular Value
SVD	Singular Value Decomposition
ADI	Alternating Direction Implicit
EKS	Extended Krylov Subspace
SPD	Symmetric Positive Definite
BiCG	Bi-Conjugate Gradient
CG	Conjugate Gradient
MRE	Mean Relative Error
MAX_RE	Maximum Relative Error

Chapter 1

Introduction

1.1 Motivation

The ever-expanding need for accurate simulation of large, and sometimes complex, integrated circuits (ICs) forms a great challenge for the semiconductor industry. Moore's law, especially, emphasizes this problem, while it states that the number of transistors on a single microchip doubles every two years. In Figure 1.1, Moore's law presented by [1] shows

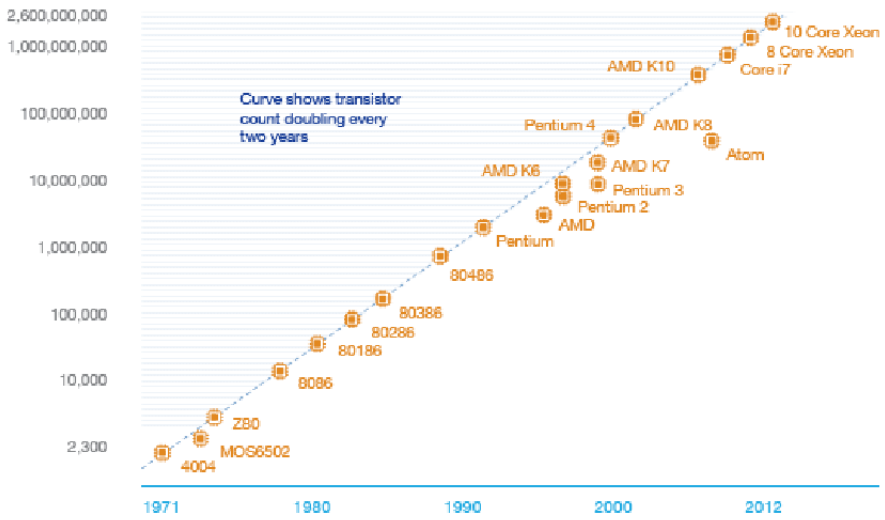

Figure 1.1: Moore's law.

a graphical representation of the increasing number of transistors in a single microchip over time.

The dramatic outburst in circuit complexity was seen by the steadily shrinking of the transistors' size over the course of time. In the late 40s, the dimensions of transistors were

expressed at the scale of millimeter while in the early 2010s were commonly measured in nanometers, a reduction at a scale of 100,000 units.

The simulation process is one of the most important tools for designing and understanding large-scale ICs. In order for such ICs to be simulated, long-lasting and computationally expensive simulations are needed. The semiconductor industry focus its efforts on simulating large electromagnetic models in a short term of time with the maximum possible accuracy. Research has shown that there is a lot of superfluous information in the model before the simulation process that could be excluded without compromising the accuracy of the process. Hence, it seems that one of the most efficient moves is to reduce the size of large electromagnetic models and eliminate the unnecessary details, while preserving the accuracy and realism in the simulation process results.

Model Order Reduction (MOR) aims to reduce the computational complexity in various mathematical models addressing numerical simulations. It is related with the idea of meta-modeling in order to produce fast and real-time simulations for large-scale systems. MOR methods are usually applied in the area of control systems, however, several definitions of MOR can be distinguished by the context of the method. Figure 1.2 shows the modeling complexity of the physical systems based on several analysis areas.

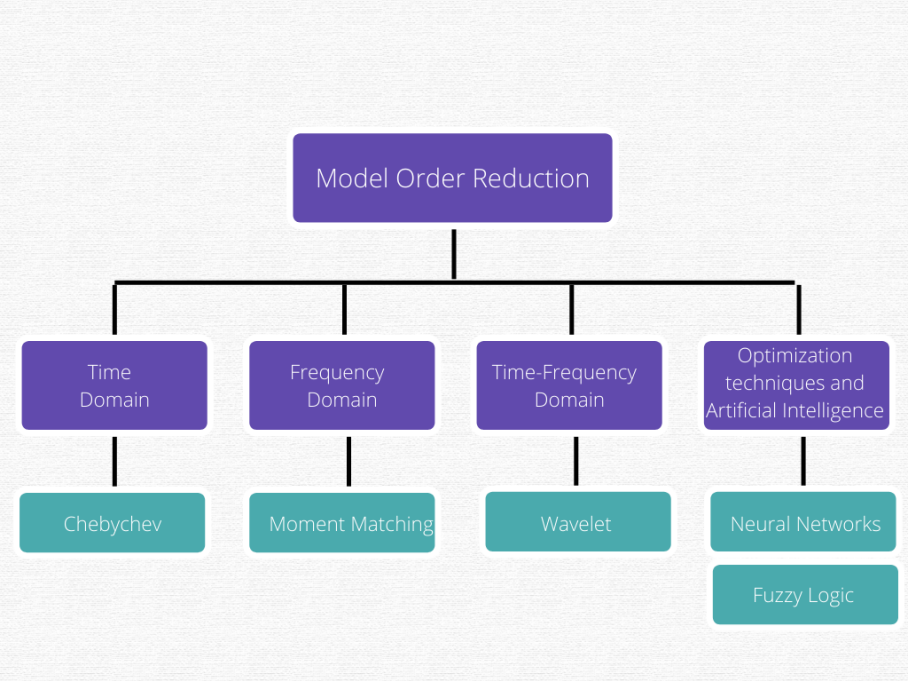

Figure 1.2: Flow chart of model order reduction.

MOR is divided in two large categories, the Moment Matching (MM) techniques first

presented in [2] and techniques like Balanced Truncation (BT) presented in [3]. The first category is well-established by dint of the computational efficiency of the reduced order models (ROMs) that it produces. Notwithstanding, MM-produced ROMs are not based on an a-priori error (which easily can be specified by the engineer) but are mainly based on the number of matching moments. As a result, the error of the overall process is only available after the ROM generation. On the other hand, BT-like methods offer competent results. Unfortunately, these methods deal with expensive computations and storaging due to the solution of the Lyapunov matrix equations.

Overall, MOR methods have proven their effectiveness in transforming large and complex mathematical models into smaller and simpler ones. BT especially is a method with great contribution in the research of the MOR area with noteworthy benefits. The most important of them, presented in [4], is that they provide a global bound error, so as to determine the reduced order before the computation of the reduced model, while they handle expensive tasks (e.g., solution of the Lyapunov equations) with low-rank approaches.

1.2 Contribution

To mitigate the computationally expensive task, we propose a low-rank solution of the Lyapunov equation. The approach is based on Krylov subspaces approaches, where iteratively large dimensional subspaces are projected into lower ones in order to obtain the above mentioned low-rank solution of the Lyapunov equations. In this thesis, we present a ROM generation tool that uses BT techniques. Particularly, we present the default implementation of BT and the implementation of BT that exploits Krylov subspace methods, with main focus to reduce the scale of very large electromagnetic models and achieve low execution times and great accuracy. [1] The contributions of this thesis are summarized below:

- We create a ROM tool that handles large-scale electromagnetic models, comprising of several million units. The reduced-order matrices produced by our ROM tool display great accuracy and very small dimension, according to user specifications.

- State-of-the-art C++ solvers for sparse and dense matrices were used to obtain exceptionally fast execution times.

[1] This research was conducted under the auspices of the Electronics Research Lab of University of Thessaly.

- Experiments have proved that our tool achieves remarkably fast reduction of large electromagnetic models consisting of millions of units, while providing great accuracy compared to the original models.

1.3 Outline

The rest of the thesis is organised as follows. In Chapter 2, we present in detail the modeling of electrical circuits and their simulation process. In Chapter 3, we provide the theoretical background of MOR, focusing on the BT method. In Chapter 4, we introduce computational improvements of the BT method along with detailed notes concerning the implementation of both the default and the proposed version of BT. Chapter 5 presents the experimental evaluation of the proposed low-rank BT approach. Finally, in Chapter 6, we conclude this thesis.

Chapter 2

Modeling and Simulation

2.1 Model Equations for Electrical Circuits

All lumped electrical circuits consist of electrical elements such as resistors, inductors, capacitors, current sources, and voltage sources. The modeling base of the dynamical behavior for such electrical topologies, derives from Kirchoff's laws along with the constitutive relations of the electrical system's elements. In this thesis, we consider the Modified Nodal Analysis (MNA) modeling setup.

2.1.1 Modified Nodal Analysis

The MNA is one of the most common ways to model an electrical circuit. This modeling technique considers a graph whose branches represent the circuit elements and the nodes represent the interconnections of these elements. An incidence matrix A_0 describes this structure, a graph with n nodes and b branches, with entries:

$$a_{ij} = \begin{cases} -1 & \text{if branch } j \text{ leaves node } i \\ 1 & \text{if branch } j \text{ enters node } i \\ 0 & \text{if branch } j \text{ is not incident with node } i \end{cases} \tag{2.1}$$

Note: The dimensions of matrix A_0 is $n \times b$.

In case the network graph is connected, the rows of matrix A_0 are linearly dependent and we can randomly choose one node and handle it as reference. By eliminating the corresponding row for this node in matrix A_0, a new reduced incidence matrix A is produced (the

dimensions of the new matrix are $((n-1) \times b)$. The new reduced matrix A now has full row rank.

Given that $u(t)$ = [u_1(t) u_2(t) ... u_b(t)]^T is the vector of branch voltages, $v(t)$ = [v_1(t) v_2(t) ... v_n(t)] is the vector of all node potentials (except from the reference node) and $i(t)$ = [i_1(t) i_2(t) ... i_b(t)]^T is the current vector of b branches, then the topology's equations are described by Kirchoff's law as follows:

Kirchhoff's Voltage Law, KVL

$$u(t) = A^T v(t) \tag{2.2}$$

Kirchhoff's Current Law, KCL

$$Ai(t) = 0 \tag{2.3}$$

Suppose that the circuit elements are divided in two categories.

- Elements whose equations can be written as:

$$i_k(t) = g_k u_k(t) + c_k \frac{du_k(t)}{dt} + s_k(t) \tag{2.4}$$

  , for circuits consisting of resistors, capacitors, and current sources.

- Elements whose equations cannot be written using the above form. They are addressed to circuits consisting of inductors and voltage sources.

Let b_1 be the number of elements belonging to the first category (G_1) and b_2 the elements of G_2, b = b_1 + b_2. If we seperate the incidence matrix A and the vectors u(t) and i(t) in sub-matrices and sub-vectors belonging to the groups presented above, we have:

Kirchhoff's Current Law, KCL

$$Ai(t) = 0 \Leftrightarrow A_1 i_1 + A_2 i_2 = 0 \tag{2.5}$$

Kirchhoff's Voltage Law, KVL

$$u(t) = A^T v(t) \Leftrightarrow \begin{cases} u_1 = A_1^T v(t) \\ u_2 = A_2^T v(t) \end{cases} \tag{2.6}$$

Considering the above, the equations for the elements of the first category G1 can be written as:

$$i_1(t) = Gu_1(t) + C\frac{du_1(t)}{dt} + s_1(t) \tag{2.7}$$

, where:

- G is a diagonal matrix $b_1 \times b_1$ with non−zero diagonal values where there are conductances,
- C is a diagonal matrix $b_1 \times b_1$ with non−zero diagonal values where there are capacitors,
- s_1(t) is the a vector $b_1 \times 1$ with non−zero values where there are current sources.

The equations for the elements of the second category G2 can be written as:

$$u_2(t) = L\frac{di_2}{dt} + s_2(t) \tag{2.8}$$

, where:

- L is a diagonal matrix $b_2 \times b_2$ with non−zero diagonal values where there are inductors,
- $s_2(t)$ is the a vector $b_2 \times 1$ with non−zero values where there are voltage sources.

By replacing the first equation of Eq. (2.6) to Eq. (2.7), and then to Eq. (2.5), we have

$$A_1 G A_1^T v(t) + A_1 C A_1^T \frac{dv(t)}{dt} + A_2 i_2(t) = -A_1 s_1(t) \tag{2.9}$$

Furthermore, if we replace the second equation of Eq. (2.6) to Eq. (2.8), we have

$$A_2 v(t) - L\frac{di_2(t)}{dt} = s_2(t) \tag{2.10}$$

Eq. (2.9) forms a system of $(n-1)$ equations and $(n-1)+b_2$ unknown variables, while Eq. (2.10) forms a system of b_2 equations and $(n-1)+b_2$ unknown variables. The combination of the two forms presented above, gives a new $[(n-1)+b_2] \times [(n-1)+b_2]$ system:

$$\begin{bmatrix} A_1 G A_1^T & A_2 \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ i_2(t) \end{bmatrix} + \begin{bmatrix} A_1 C A_1^T & 0 \\ 0 & -L \end{bmatrix} \begin{bmatrix} \frac{dv(t)}{dt} \\ \frac{di_2(t)}{dt} \end{bmatrix} = \begin{bmatrix} -A_1 s_1(t) \\ s_2(t) \end{bmatrix} \tag{2.11}$$

which constitutes the MNA system.

In the case of DC analysis, we exclude the time factor and the MNA system takes the form:

$$\begin{bmatrix} A_1 G A_1^T & A_2 \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} v \\ i_2 \end{bmatrix} = \begin{bmatrix} -A_1 s_1 \\ s_2 \end{bmatrix} \tag{2.12}$$

2.2 Simulation

The term circuit simulation describes the process of predicting and verifying the behavior and the performance of the circuit. Since the ever-expanding growth of the semiconductor industry, a great need for faster and cost-effective simulators has risen. The fabrication demands accurate simulations of the ICs' behavior, before the phase of fabrication, in order for possible problems to be spotted and fixed.

Generally, there are two different approaches to the simulation process. At one end of the spectrum, there are analog simulators offering accurate representations of the electrical circuit, but they are usually used only for small circuits. At the other end of the spectrum, digital simulators make use of functional representations (described by hardware languages) of the electrical circuit. Analog simulators offer higher accuracy for small circuits, but digital simulators offer the highest capacity and performance. For large-scale electrical circuits, digital simulators are preferred.

2.2.1 Transient Analysis

In transient analysis, a circuit's behavior is simulated over a period of time (which is defined by the user) [5]. The accuracy of this process depends on the simulation time and the number of the internal time steps. In the case of transient analysis or response-time analysis in an MNA system (presented in Chapter 2.1) with circuit elements such as capacitors, resistors, and inductors, the system has the below form:

$$\begin{bmatrix} A_1 G A_1^T & A_2 \\ A_2^T & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ i_2(t) \end{bmatrix} + \begin{bmatrix} A_1 C A_1^T & 0 \\ 0 & -L \end{bmatrix} \begin{bmatrix} \frac{dv(t)}{dt} \\ \frac{di_2(t)}{dt} \end{bmatrix} = \begin{bmatrix} -A_1 s_1(t) \\ s_2(t) \end{bmatrix} \tag{2.13}$$

The above system is a first-order system of linear equations with constant coefficients:

$$\tilde{G}x(t) + \tilde{C}\frac{dx(t)}{dt} = e(t) \tag{2.14}$$

If we define a start time for the x(t) factor t_0 (x(t_0) = x_0), then the problem can be described as:

$$\begin{cases} \tilde{G}x(t) + \tilde{C}\frac{dx(t)}{dt} = e(t) \\ x(t_0) = x_0 \end{cases}$$

The problem is defined as an initial value problem (IVP) and under constraints has a unique solution x(t) in a time interval [t_0 , t_f].

The solution for an IVP problem is usually computed with arithmetic approaches in a time interval [t_0 , t_f], for discrete times $t_0 < t_1 < t_2 < ... < t_m \equiv t_f$. The solution can be found by computing an estimate x(t_k) of x(t) for every discrete time t_k (k = 1,2,..,m), starting from the initial condition x(t_0) = x_0. The value $h_k = t_k - t_{k-1}$ is called time step or sampling step at the t_k time. If the time points were selected to be spaced equally, then the time step is constant and the computation of x(t_k) for every time point t_k (k = 1,2,..,m) can be performed by one of the below two approaches, for $\frac{dx(t_k)}{dt}$ in the system of Eq. (2.14).

- Backward Euler (BE) or Implicit Euler approach

$$\frac{dx(t_k)}{dt} \approx \frac{1}{h}[x(t_k) - x(t_{k-1})]$$

The above MNA system of Eq. (2.14) takes the form:

$$(\tilde{G} + \frac{1}{h}\tilde{C})x(t_k) = e(t_k) + \frac{1}{h}\tilde{C}x(t_{k-1}), k = 1, 2, ..., m$$

- Trapezoidal (TR) approach

$$\frac{1}{2}\left[\frac{dx(t_k)}{dt} + \frac{dx(t_{k-1})}{dt}\right] \approx \frac{1}{h}\left[x(t_k) - x(t_{k-1})\right]$$

The system is now transformed to the below linear system:

$$\tilde{G}[x(t_k) - x(t_{k-1})] + \tilde{C} \underbrace{\left[\frac{dx(t_k)}{dt} + \frac{dx(t_{k-1})}{dt}\right]}_{2[x(t_k)-x(t_{k-1})]/h} = e(t_k) + e(t_{k-1}) \Leftrightarrow$$

$$\left(\tilde{G} + \tfrac{2}{h}\tilde{C}\right) x(t_k) = e(t_k) + e(t_{k-1}) - \left(\tilde{G} - \tfrac{2}{h}\tilde{C}\right) x(t_{k-1}), k = 1, 2, ..., m$$

The TR approach is more accurate for a given step (or allows bigger steps to achieve the same accuracy) and is usually preferred as the default method. However, in certain cases, it presents an undesirable phenomenon, called ”ringing”, making it less accurate in non-smooth transitions, in which cases the backward Euler is preferred.

Chapter 3

Model Order Reduction

The simulation of complex and large-scale systems is a challenge for the semiconductor industry, especially nowadays, where the majority of the systems present those characteristics. The functional simulation of such systems demands solving equations with dimension scale that exceeds millions or even billions of units. Different MOR techniques were introduced, to overcome this problem by downscaling the original models. Generally, MOR

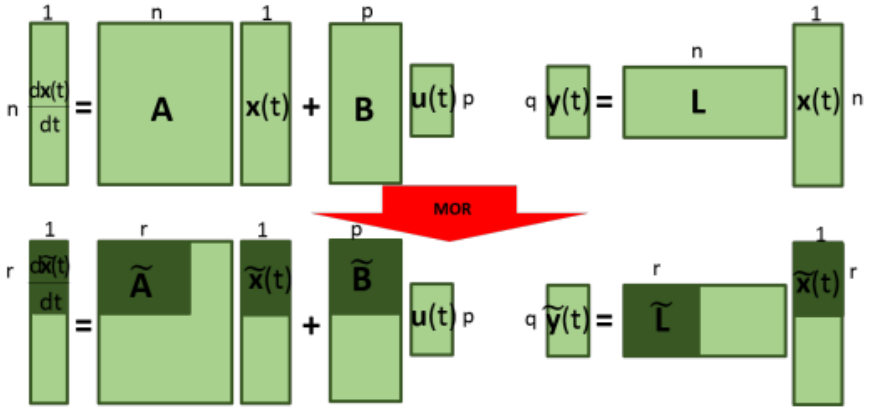

Figure 3.1: Moder order reduction on LTI systems.

methods are divided into two categories, the MM techniques ([6], [7], [8], [9]) and the system theoretic techniques ([4], [10]).

MM methods (i.e., Krylov subspace approaches) handle the downscaling procedure by creating a subspace (Krylov subspace) of much smaller dimensions and then project the original system onto the new subspace. To obtain the Krylov subspace, this method uses moments of the original transfer function to approximate the reduced transfer function. The MM methods are well-established due to their contribution in efficient computational production of

new reduced-order models. Except from the computational performance, these techniques lead to stable reduced models and ensure an acceptable accuracy.

Despite the above advantages, MM techniques, in contrast to system theoretic methods, do not offer an a-priori error and the efficiency of the algorithm can only be estimated after the generation of the ROM. Furthermore, the quality of the reduced system depends solely on the quality of the produced Krylov subspace. MM techniques, also fail to preserve significant properties of the system, such as the passivity and the stability of the system. System theoretic methods, and especially the BT method [3], were proposed to overcome some of the inadequacies of the MM techniques.

BT method offers greater accuracy by preserving important properties, such as the system's stability [11], while providing an a-priori error between the transfer function of the original and the reduced model [12]. The main focus of the BT method is to discard-truncate states that contribute less in terms of observability and controllability. In order to achieve this, it truncates the smallest Hankel singular values (HSVs). However, the BT algorithm consists of computationally expensive methods, such as the solution of the Lyapunov equations, while dealing with storage issues because of the dense matrices produced from their solution.

Different approaches have been proposed to handle the memory requirements and the computational overhead of the BT method. These approaches address the problem either by limiting the frequency reduction window of the method [13] or by solving the Lyapunov equations in a low-rank factorized form. The last approach has two alternatives. The first one is the Alternating Direction Implicit (ADI) and the second alternative is the Extended Krylov Subspace (EKS) approach. The ADI method presents fast convergence but in order to achieve that, certain input shift parameters are needed. In addition, these input parameters rely on unclear heuristics and their selection may affect the convergence of the entire algorithm. Projection-type methods, on the other side, do not rely on specific parameters and also they form a well-studied and straightforward implementation. The EKS approach uses two complementary subspaces to achieve fast convergence. For this thesis, we consider the EKS approach as the low-rank solution of the Lyapunov equations.

3.1 Balanced Truncation

Consider an MNA system with m (inductive) branches, n nodes, p inputs, and q outputs, which translates into an RLC circuit in the time domain:

$$\begin{aligned} \begin{pmatrix} \mathbf{G_n} & \mathbf{E} \\ -\mathbf{E}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{i}(t) \end{pmatrix} + \begin{pmatrix} \mathbf{C_n} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{v}}(t) \\ \dot{\mathbf{i}}(t) \end{pmatrix} &= \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{pmatrix} \mathbf{u}(t) \\ \mathbf{y}(t) = \begin{pmatrix} \mathbf{L}_1 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{v}(t) \\ \mathbf{i}(t) \end{pmatrix} &+ \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{3.1}$$

, where:

- $G_n \in \mathbb{R}^{n \times n}$ is the conductance matrix,
- $C_n \in \mathbb{R}^{n \times n}$ is the node capacitance matrix,
- $M \in \mathbb{R}^{m \times m}$ is the branch inductance matrix,
- $E \in \mathbb{R}^{n \times m}$ is the node-to-branch incidence matrix,
- $v \in \mathbb{R}^n$ is the vector of node voltages,
- $i \in \mathbb{R}^m$ is the vector of inductive current sources,
- $B_1 \in \mathbb{R}^{n \times p}$ is the input-to-state connectivity matrix,
- $u \in \mathbb{R}^p$ is the vector of the input excitations from the current sources,
- $y \in \mathbb{R}^q$ is the vector of the output measurements,
- $L_1 \in \mathbb{R}^{q \times n}$ is the state-to-output connectivity matrix,
- $D \in \mathbb{R}^{q \times p}$ is the input-to-output connectivity matrix.

Note that below we denote $\dot{v} \equiv \frac{dv(t)}{dt}$ and $\dot{i} \equiv \frac{di(t)}{dt}$.

Without loss of generality, we make the assumption that all voltage sources were transformed to Norton-equivalent current sources. In addition, we suppose that all outputs are obtained as node voltages at the nodes.

The model order is denoted as N ≡ m + n, the state vector as x(t) ≡ $\begin{pmatrix} v(t) \\ i(t) \end{pmatrix}$, and also

$$G \equiv -\begin{pmatrix} G_n & E \\ -E^T & 0 \end{pmatrix}, C \equiv \begin{pmatrix} C_n & 0 \\ 0 & M \end{pmatrix}, B \equiv \begin{pmatrix} B_1 \\ 0 \end{pmatrix}, L \equiv \begin{pmatrix} L_1 & 0 \end{pmatrix}$$

Then, the above MNA system of Eq. (3.1) can be written in the following form (descriptor form):

$$\begin{aligned} \mathbf{C}\frac{d\mathbf{x}(t)}{dt} &= \mathbf{G}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{L}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{3.2}$$

The main focus of a MOR procedure is to produce a reduced-order model:

$$\begin{aligned} \tilde{\mathbf{C}}\frac{d\tilde{\mathbf{x}}(t)}{dt} &= \tilde{\mathbf{G}}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(\mathbf{t}) \\ \tilde{\mathbf{y}}(t) &= \tilde{\mathbf{L}}\tilde{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{3.3}$$

, where:

- $\tilde{G}, \tilde{C} \in \mathbb{R}^{r \times r}$,
- $\tilde{B} \in \mathbb{R}^{r \times p}$,
- $\tilde{L} \in \mathbb{R}^{q \times r}$.

Consider that the order r ≪ N and the output error is bounded $\|\tilde{y}(t) - y(t)\|_2 < \epsilon\|u(t)\| - 2$, for a given vector u(t) and supposing that ϵ is a small number. The above bounded error can be written equivalently in the frequency domain as $\|\tilde{y}(s) - y(s)\|_2 < \epsilon\|u(s)\| - 2$ via the Plancherel's theorem [14]. If the transfer functions of the original and the reduced model in the frequency domain are:

$$H(s) = L(sC - G)^{-1}B + D\tilde{H(s)} = L(s\tilde{C} - \tilde{G})^{-1}\tilde{B} + D$$

Then the error in the frequency domain is:

$$\|\tilde{y}(s) - y(s)\|_2 = \|\tilde{H}(s)u(s) - H(s)u(s)\|_2 \leq \|\tilde{H}(s) - H(s)\|_\infty \|u(s)\|_2 \tag{3.4}$$

where the $\|.\|_\infty$ is the $\mathcal{H}_\infty$ norm of the rational transfer function, or the $\mathcal{L}_2$ matrix norm. Hence, in order for the error to be bounded, it is essential to bound the distance between the transfer functions $||\tilde{\mathbf{H}}(s) - \mathbf{H}(s)||_\infty < \epsilon$.

Related MOR methods, and particularly the BT method, use the observability and controllability Gramian matrices P, Q such that:

$$\begin{aligned} \mathbf{P} &= \int_0^\infty exp(\mathbf{C}^{-1}\mathbf{G}t)\mathbf{C}^{-1}\mathbf{B}\mathbf{B}^T\mathbf{C}^{-T}exp(\mathbf{C}^{-1}\mathbf{G}t)^T dt \\ \mathbf{Q} &= \int_0^\infty exp(\mathbf{C}^{-1}\mathbf{G}t)^T\mathbf{L}^T\mathbf{L}exp(\mathbf{C}^{-1}\mathbf{G}t)dt \end{aligned} \tag{3.5}$$

which are equivanlently obtained by the solution of the Lyapunov equations [12]

$$\begin{aligned} (\mathbf{C}^{-1}\mathbf{G})\mathbf{P} + \mathbf{P}(\mathbf{C}^{-1}\mathbf{G})^T &= -(\mathbf{C}^{-1}\mathbf{B})(\mathbf{C}^{-\mathbf{1}}\mathbf{B})^T \\ (\mathbf{C}^{-1}\mathbf{G})^T\mathbf{Q} + \mathbf{Q}(\mathbf{C}^{-1}\mathbf{G}) &= -\mathbf{L}^T\mathbf{L} \end{aligned} \tag{3.6}$$

taking into consideration that the $\mathbf{C}$ matrix is nonsingular.

Generally, the controllability Gramian matrix $\mathbf{P}$ presents the input-to-state behavior, expressly the degree to which the inputs can control the states, while the observability Gramian matrix $\mathbf{Q}$ presents the state-to-output behavior, that is the degree that the states are observable at the outputs. The main focus of BT is to truncate the states that are difficult to reach but easy to observe. Nevertheless, in the original system's model, there are states that are easy to observe and difficult to reach and backwards. So before truncating the original model, it is essential to transform it into a new coordinate system that every state has the same degree of difficulty to be reached and to be observed. To achieve the above, there is such transformation $\mathbf{T}x(t)$, which directs to a new transformed model:

$$\begin{aligned} \mathbf{T}\mathbf{C}\mathbf{T}^{-1}\frac{d(\mathbf{T}\mathbf{x}(t))}{dt} &= \mathbf{T}\mathbf{G}\mathbf{T}^{-\mathbf{1}}(\mathbf{T}\mathbf{x}(t)) + \mathbf{T}\mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{L}\mathbf{T}^{-\mathbf{1}}(\mathbf{T}\mathbf{x}(t)) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{3.7}$$

, therefore preserving the transfer function H(s) and leads to [12]:

$$\mathbf{P} = \mathbf{Q} = diag(\sigma_1, \sigma_2, ..., \sigma_N) \tag{3.8}$$

, where $\sigma_i, i = 1, ..., N$ are the HSVs of the origial model. HSVs stand for the square root of the eigenvalues of the product $\mathbf{PQ}$ ($\sigma_i = \sqrt{\lambda_i(\mathbf{PQ})}, i = 1, ..., N$) for all coordinate sytems of the state space. In the new transformed balanced model, the states are easier to observe and reach, and correspond to the greatest HSVs. The distance between the original and the reduced-order transfer function can be estimated by the truncated $\mathbf{N} - \mathbf{r}$ HSVs (corresponding to the smallest HSVs) and is bounded as:

$$||\mathbf{H}(s) - \tilde{\mathbf{H}}(s)||_\infty \leq 2(\sigma_{r+1} + \sigma_{r+2} + ... + \sigma_N) \tag{3.9}$$

Eq. (3.9) defines the a-priori criterion in order to select the reduced order $\mathbf{r}$ given a desired output error tolerance ϵ and constitutes one of the most significant advantages of BT compared to other MOR methods. The main steps of BT are presented in Algorithm 1.

Algorithm 1 MOR by Balanced Truncation

1: Compute the solution of the Lyapunov equations (3.6) in order to obtain the Gramian matrices $\mathbf{P}$ and $\mathbf{Q}$

2: Calculate the eigenvalue decomposition of $\mathbf{PQ}$, or compute the singular value decomposition (SVD) for the product of the Cholesky factors $\mathbf{P} = \mathbf{Z}_P\mathbf{Z}_P^T$ and $\mathbf{Q} = \mathbf{Z}_Q\mathbf{Z}_Q^T$, i.e., $\mathbf{Z}_P^T\mathbf{Z}_Q = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$

3: Compute the transformations matrices $\mathbf{T}_{(r\times N)} = \mathbf{\Sigma}_{(r\times r)}^{-1/2}\mathbf{V}_{(r\times N)}\mathbf{Z}_Q^T$ and $\mathbf{T}_{(N\times r)}^{-1} = \mathbf{Z}_P\mathbf{U}_{(N\times r)}\mathbf{\Sigma}_{(r\times r)}^{-1/2}$, and the corresponding truncated reduced-order matrices as $\tilde{\mathbf{C}} = \mathbf{T}_{(r\times N)}\mathbf{C}\mathbf{T}_{(N\times r)}^{-1}$, $\tilde{\mathbf{G}} = \mathbf{T}_{(r\times N)}\mathbf{G}\mathbf{T}_{(N\times r)}^{-1}$, $\tilde{\mathbf{B}} = \mathbf{T}_{(r\times N)}\mathbf{B}$, $\tilde{\mathbf{L}} = \mathbf{L}\mathbf{T}_{(r\times N)}^{-1}$

Despite the advantages of BT, this method has significant computational and memory cost, which hampers its applicability to large-scale models (where N exceeds a few thousand states). The solution of the Lyapunov equation, the SVD [15], and the Cholesky [16] factorization, as mentioned in the introduction of this chapter, are all computationally expensive procedures with complexity in the range of $\mathbf{O}(\mathbf{N}^3)$. In addition, the BT method involves the storage of dense matrices $\mathbf{P}$ and $\mathbf{Q}$, despite of the density of the original matrices $\mathbf{C}$, $\mathbf{G}$, $\mathbf{B}$, $\mathbf{L}$.

Notwithstanding, in most cases the number of inputs and outputs is much smaller than the number of states ($\mathbf{p}, \mathbf{q} \ll \mathbf{N}$), meaning that the products of $\mathbf{BB}^T$ and $\mathbf{L}^T\mathbf{L}$ will have lower than N rank (this also holds for the Gramian matrices $\mathbf{P}$ and $\mathbf{Q}$). Considering the previous observation, it follows that $\mathbf{P}$ and $\mathbf{Q}$ can be approximated by low-rank products (instead of the full Cholesky factorization of the default algorithm), such as $\mathbf{P} \approx \mathbf{Z}_P\mathbf{Z}_P^T$ and $\mathbf{Q} \approx \mathbf{Z}_Q\mathbf{Z}_Q^T$ where $\mathbf{Z}_P$, $\mathbf{Z}_Q \in \mathbb{R}^{N\times k}$ and $\mathbf{N} \ll \mathbf{k}$. This way, the memory requirements and the complexity of the SVD procedure are significantly reduced, leaving the solution of the Lyapunov equations as the main task that adds computational overhead.

Chapter 4

Computational Improvements in Balanced Truncation MOR

In this chapter, we present our approach on a BT implementation using an EKS low-rank iterative method. To this end, we discuss all steps in detail for an efficient implementation of the proposed algorithm, that we used in our ROM generation tool.

The main part of low-rank Krylov subspace methods for computing approximate solutions of large-scale Lyapunov equations like (3.6), is to iteratively project them onto a lower-dimensional subspace, and then solve the produced small-scale equations. Each iteration increases the dimension of the projection subspace, until convergence is attained.

Consider a subspace $\mathbf{K}$, where $\mathbf{K}^{(j)}$ is a projection whose columns span $\mathbf{K}$. The small-scale Lyapunov equation is derived by projecting the large-scale matrix onto the approximation subspace $\mathbf{K}$, i.e.,

$$\mathbf{M}\mathbf{X} + \mathbf{X}\mathbf{M}^T = -\mathbf{R}\mathbf{R}^T \tag{4.1}$$

, where $\mathbf{K}^{(j)} \in^{N\times k}$ ($k << N$), $\mathbf{M} = \mathbf{K}^{(j)T}\mathbf{G}_C\mathbf{K}^{(j)}$, $\mathbf{R} = \mathbf{K}^{(j)T}\mathbf{B}$, $\mathbf{G}_C \equiv \mathbf{C}^{-1}\mathbf{G}$.

After solving (4.1), an approximate solution $\mathbf{Y} = \mathbf{K}^{(j)}\mathbf{X}\mathbf{K}^{(j)T}$ is found on subspace $\mathbf{K}$. The residual $\mathbf{R} = \mathbf{G}_C\mathbf{Y} + \mathbf{Y}\mathbf{G}_C^T + \mathbf{B}\mathbf{B}^T$ is orthogonal to $\mathbf{K}$, which is also referred to as the Galerkin condition [17].

Note that the above procedure is independent of the chosen subspace, but its effectiveness and convergence are seriously influenced by the selection. In some studies, the standard Krylov subspace was used as the approximation subspace, but this method usually requires many iterations until a good approximation of the solution is obtained [18]. This leads to a much higher final rank on the solution (reduced size), as well as longer execution time. On

the other hand, the EKS proves to be more efficient, and also achieves faster convergence in comparison to that of the standard Krylov subspace method [19].

The standard Krylov subspace is defined as

$$\mathcal{K}_k(\mathbf{G}_C, \mathbf{B}_C) = span\{\mathbf{B}_C, \mathbf{G}_C\mathbf{B}_C, \mathbf{G}_C^2\mathbf{B}_C, \ldots, \mathbf{G}_C^{k-1}\mathbf{B}_C\} \tag{4.2}$$

, where:

$$\mathbf{G}_C \equiv \mathbf{C}^{-1}\mathbf{G}, \quad \mathbf{B}_C \equiv \mathbf{C}^{-1}\mathbf{B}$$

which can be enriched with information from the subspace $\mathcal{K}_k(\mathbf{G}_C^{-1}, \mathbf{B}_C)$, corresponding to the inverse matrix $\mathbf{G}_C^{-1}$ leading to:

$$\begin{aligned} &\mathcal{K}_k^C(\mathbf{G}_C, \mathbf{B}_C) = \mathcal{K}_k(\mathbf{G}_C, \mathbf{B}_C) + \mathcal{K}_k(\mathbf{G}_C^{-1}, \mathbf{B}_C) = \\ &span\{\mathbf{B}_C, \mathbf{G}_C^{-1}\mathbf{B}_C, \mathbf{G}_C\mathbf{B}_C, \mathbf{G}_C^{-2}\mathbf{B}_C, \mathbf{G}_C^2\mathbf{B}_C, \ldots, \\ &\qquad \mathbf{G}_C^{-(k-1)}\mathbf{B}_C, \mathbf{G}_C^{k-1}\mathbf{B}_C\} \end{aligned} \tag{4.3}$$

which is known as the EKS.

The only compromise is that the matrix $\mathbf{G}_C$ requires inversion in the means of the EKS method, which is not required in the standard Krylov subspace method. Nevertheless, despite this additional step, the EKS method still competes with the computational efficiency of the standard Krylov subspace method. In fact, during the iterative process, $\mathbf{G}_C^{-1}$ is not explicitly required.

4.1 Proposed Algorithm

The EKS method starts by the pair $\{\mathbf{B}_C, \mathbf{G}_C^{-1}\mathbf{B}_C\}$ and generates a sequence of extended subspaces $\mathcal{K}_k^C(\mathbf{G}_C, \mathbf{B}_C)$ of increasing dimensions, solving the projected Lyapunov equation in each iteration, until a sufficiently accurate approximation of the solution of Eq. (3.6) is obtained. The complete EKS method is given in Algorithm 2.

4.2 Implementation Details

In this subsection, we present the details concerning the efficient implementation of the default BT and the proposed low-rank BT methods. The dense and sparse matrix representation and the implemented procedures, utilize types and methods from the Eigen library (C++).

Algorithm 2 Extended Krylov Subspace method (EKSM) for low-rank solution of Lyapunov equations

Input: $\mathbf{G}_C \equiv \mathbf{C}^{-1}\mathbf{G}, \mathbf{B}_C \equiv \mathbf{C}^{-1}\mathbf{B}$ (or $\mathbf{G}_C^T$, $\mathbf{L}^T$)

Output: $\mathbf{Z}$ such that $\mathbf{P} \approx \mathbf{Z}\mathbf{Z}^T$

1: $j = 1; p = size\_col(\mathbf{B}_C)$

2: $\mathbf{K}^{(j)} = Orth([\mathbf{B}_C, \mathbf{G}_C^{-1}\mathbf{B}_C])$

3: **while** $j < maxiter$ **do**

4: $\quad \mathbf{M} = \mathbf{K}^{(j)T}\mathbf{G}_C\mathbf{K}^{(j)}; \quad \mathbf{R} = \mathbf{K}^{(j)T}\mathbf{B}_C$

5: $\quad$ Solve $\mathbf{M}\mathbf{X} + \mathbf{X}\mathbf{M}^T = -\mathbf{R}\mathbf{R}^T$ for $\mathbf{X} \in^{2pj \times 2pj}$

6: $\quad$ **if** converged **then**

7: $\quad\quad \mathbf{S} = Chol(\mathbf{X})$

8: $\quad\quad \mathbf{Z} = \mathbf{K}^{(j)}\mathbf{S}$

9: $\quad\quad$ **break**

10: $\quad$ **end if**

11: $\quad k_1 = 2p(j-1); k_2 = k_1 + p; k_3 = 2pj$

12: $\quad \mathbf{K}_1 = [\mathbf{G}_C\mathbf{K}^{(j)}(:, k_1 + 1 : k_2), \mathbf{G}_C^{-1}\mathbf{K}^{(j)}(:, k_2 + 1 : k_3)]$

13: $\quad \mathbf{K}_2 = Orth(\mathbf{K}_1) \quad$ w.r.t $\quad \mathbf{K}^{(j)}$

14: $\quad \mathbf{K}_3 = Orth(\mathbf{K}_2)$

15: $\quad \mathbf{K}^{(j+1)} = [\mathbf{K}^{(j)}, \mathbf{K}_3]$

16: $\quad j = j + 1$

17: **end while**

Furthermore, a collection of state-of-the-art solvers were employed, so as to achieve faster convergence and overall results.

4.2.1 Matrix products with inverse of sparse matrix

Algorithm 2 involves the inverse $\mathbf{G}_C^{-1}$ of the sparse system matrix $\mathbf{G}_C$. Regrettably, it should be noted that inverting a sparse matrix will produce a dense matrix, and is also a very expensive computational operation that should be avoided if it is not explicitly needed. In our case, however, the inverse matrix $\mathbf{G}_C$ is only used in products with the $\mathbf{N} \times \mathbf{p}$ matrix $\mathbf{B}$ (initially) and then with the $\mathbf{N} \times \mathbf{pj}$ matrix $\mathbf{K}^{(j)}$ in step 12 for each iteration j (where $\mathbf{p}, \mathbf{pj} \ll \mathbf{n}$, and the iteration count is typically very small). Therefore, the inputs to Algorithm

2 are not actually $\mathbf{G}_C \equiv \mathbf{C}^{-1}\mathbf{G}$, $\mathbf{B}_C \equiv \mathbf{C}^{-1}\mathbf{B}$, but the sparse system matrices $\mathbf{G}$, $\mathbf{C}$, $\mathbf{B}$ (or $\mathbf{G}^T$, $\mathbf{C}^T$, $\mathbf{L}^T$), since these products can be implemented by solving the linear systems $\mathbf{CY} = \mathbf{R}$ and $\mathbf{GY} = \mathbf{R}$ (or $\mathbf{C}^T\mathbf{Y} = \mathbf{R}$, $\mathbf{G}^T\mathbf{Y} = \mathbf{R}$), using any sparse solver.

4.2.2 Orthogonalization

For steps 2 and 14 of Algorithm 2, householder QR transformations [20] are employed, using the correspoding methods of the Eigen library. The orthogonalization in step 13, however, needs to be performed with respect to $\mathbf{K}^{(j)}$. For this purpose, a Gram-Schmidt procedure [20] is used, which is described in Algorithm 3.

Algorithm 3 Orthogonalization w.r.t. another matrix

Input: $\mathbf{K1}, \mathbf{K}^{(j)}$, #ports p

Output: $\mathbf{K2}$

1: **for** k_1 = 1, ..., j **do**

2: $k_2 = 2p(k_1 - 1); \quad k_3 = 2pk_1$

3: $\mathbf{K2} = \mathbf{K1} - \mathbf{K}^{(j)}(:, k_2 + 1 : k_3)\mathbf{K}^{(j)T}(:, k_2 + 1 : k_3)\mathbf{K1}$

4: **end for**

4.2.3 Lyapunov solver

The solution of the continuous-time Lyapunov equations (3.6), for the purpose of this thesis, was based on the Bartels-Stewart method [21] and is presented in Algorithm 4. Consider solving an equation in the form of $\mathbf{AX} + \mathbf{XA}^T + \mathbf{Q} = 0$. In the case of default BT, the $\mathbf{A}$ factor stands for the product $\mathbf{C}^{-1}\mathbf{G}$, the $\mathbf{X}$ factor stands for $\mathbf{P}$, and finally, the $\mathbf{Q}$ stands for the product $(\mathbf{C}^{-1}\mathbf{B})(\mathbf{C}^{-1}\mathbf{B})^T$, concerning the solution of matrix $\mathbf{P}$. For the solution of matrix $\mathbf{Q}$, the $\mathbf{A}$ factor stands for the product $(\mathbf{C}^{-1}\mathbf{G})^T$, the $\mathbf{X}$ factor stands for $\mathbf{Q}$, and finally, the $\mathbf{B}$ stands for the product $\mathbf{L}^T\mathbf{L}$. The Lyapunov solver, in any case, returns dense matrices and so the produced matrices $\mathbf{P}$ and $\mathbf{Q}$ are also dense, despite the density status of the input matrices $\mathbf{C}, \mathbf{G}, \mathbf{B}$, and $\mathbf{L}$.

4.2.4 Convergence criterion

The solution $\mathbf{X} \in^{k\times k}$ of Eq. (4.1) can be back-projected to the N-dimensional space to give an approximate solution $\mathbf{P} = \mathbf{K}^{(j)}\mathbf{X}\mathbf{K}^{(j)T}$ for the original large-scale equation (3.6).

Algorithm 4 Lyapunov solver

Input: $\mathbf{A}$, $\mathbf{Q}$

Output: $\mathbf{X}$

1: Apply Schur decomposition (presented in [22]) on $\mathbf{A}$, to obtain the Schur T triangular matrix $\mathbf{TA}$ and the Schur U matrix $\mathbf{ZA}$
2: Transform the right-hand side by computing $\mathbf{F} = \mathbf{ZA}^T \cdot \mathbf{Q} \cdot \mathbf{ZA}$
3: Initialize an identity matrix $\mathbf{idx}$ with dimensions equal to the dimensions of the original matrix $\mathbf{A}$ and a vector containing the diagonal elements of matrix $\mathbf{TA}$ (referred as $\mathbf{p}$)
4: Apply backward substitution to obtain the transformed solution $\mathbf{Y}$
5: **for** $k = n : -1 : 1$ **do**
6: $\quad rhs = \mathbf{F}(:, k) + \mathbf{Y} \cdot \mathbf{TA}^T(:, k)$
7: $\quad \mathbf{TA}(idx) = p + \mathbf{TA}^T(k, k)$
8: $\quad \mathbf{Y}(:, k) = \mathbf{TA} \backslash (-rhs)$
9: **end for**
10: Transform solution back by estimating $\mathbf{X} = \mathbf{ZA} \cdot \mathbf{Y} \cdot \mathbf{ZA}^T$

An appropriate stopping criterion is the residual of Eq. (3.6) with the approximate solution to reach a certain threshold in magnitude, i.e.,

$$\frac{||\mathbf{G}_C\mathbf{P} + \mathbf{P}\mathbf{G}_C + \mathbf{B}_C\mathbf{B}_C^T||}{||\mathbf{B}_C\mathbf{B}_C^T||} \leq tol \tag{4.4}$$

However, it has been proved [19] that the above criterion is actually equal to $||\mathbf{R}^T\mathbf{M}\mathbf{X}||$, which can be computed much more efficiently, and thus the stopping criterion is transformed to:

$$||\mathbf{R}^T\mathbf{M}\mathbf{X}|| \leq tol \tag{4.5}$$

A tolerance of $tol = 10^{-10}$ is typically sufficient in practice to acquire a good approximation of the solution.

4.2.5 Cholesky Factorization

Generally, the Cholesky factorization demands as inputs Symmetric Positive Definite (SPD) matrices, in order to produce an upper triangular matrix $\mathbf{U}$, such that the product $\mathbf{U}^T \cdot \mathbf{U}$ equals to the original matrix. In both Algorithms 1 and 2, the Cholesky factorization is performed on the final solutions of the Lyapunov equations. A necessary and sufficient condition

for the Lyapunov solver to produce unique and SPD matrices, when solving an equation in the form of $\mathbf{AX} + \mathbf{XA}^T + \mathbf{Q} = 0$, is that the matrix $\mathbf{A}$'s eigenvalues have positive real parts and the matrix $\mathbf{Q}$ is SPD as presented in [23]. However, in the case of $\mathbf{Q} = \mathbf{BB}^T$ (or $\mathbf{Q} = \mathbf{B}_C\mathbf{B}_C^T$), where $\mathbf{B}$ is $N \times p$ and $p \ll N$, the matrix is low-rank and does not satisfy the necessary conditions to be SPD. Thus, the Cholesky factorization is replaced by the LDLT decomposition [24] and the Cholesky factors are replaced by the appropriate computations with the produced LDLT factors ($Z_P = L_P \cdot D_P^{\frac{1}{2}}$ and $Z_Q = L_Q \cdot D_Q^{\frac{1}{2}}$, where L_P, L_Q are the permuted lower triangular matrices L for matrix P and Q respectively, and D_P, D_Q are the diagonal matrices, such that $P = L_P \cdot D_P \cdot L_P^T$ and $Q = L_Q \cdot D_Q \cdot L_Q^T$).

4.2.6 Lower-rank solution

The matrix $\mathbf{X}$ in the solution $\mathbf{P} = \mathbf{K}^{(j)}\mathbf{X}\mathbf{K}^{(j)T}$ has a final rank of $2pj$, where j is the final iteration count and is often numerically positive semi-definite [25]. If that is the case, it is possible to replace step 7 of Algorithm 2 to reduce the rank of the final solution even further. More precisely, let $\mathbf{X} = \mathbf{WDW}^T$ be the eigendecomposition of the $2m \times 2m$ matrix $\mathbf{X}$, with $\mathbf{D}$ having all the diagonal entries sorted in decreasing order. A new size k is determined ($k \ll 2pj$) by truncating all the values in $\mathbf{D}$, that are less than a specified threshold (in this case 10^{-12}). Furthermore, by only keeping the corresponding k columns of $\mathbf{W}$ and discarding the rest, the new more reduced approximation can be calculated as $\mathbf{Z} = \mathbf{K}^{(j)}\mathbf{W}\mathbf{D}^{\frac{1}{2}}$.

4.2.7 Solvers

As mentioned before, there exists a workaround, in calculating the inverse matrices $\mathbf{C}^{-1}$ and $\mathbf{G}_C^{-1}$, since they are only used in products with relatively small matrices, that ends up solving linear equations. These involve the original system matrices, which usually consist of a very large number of nodes, but very small density. Considering that, it is essential to find an efficient way to store them and use them for computations.

Since the whole process is iterative, the solvers used have to be rather fast and accurate in order to speed up convergence. The Eigen C++ library [26] offers a variety of such solvers, both iterative and direct, which can be tested to find the best option for this purpose.

At first, the attention goes to iterative solvers, since they are well-known for their low memory requirements and they are generally considered to be faster with small compromises

in accuracy. The Eigen C++ library offers both Bi-Conjugate Gradient (BiCG) and Conjugate Gradient (CG) implementations with an option for preconditioners, such as the Jacobi preconditioner or the IncompleteLUT, for non-SPD and SPD matrices respectively. However, after some iterations were performed, the condition of the matrices was deteriorating, resulting in bad convergence or no convergence at all and solver failure.

Considering the dissapointing results of the iterative methods and the need for high accuracy, direct methods had to be employed. There was a need to support both SPD and non-SPD matrices, therefore various LLT and LU implementations were tested, respectively. The fastest and more accurate solvers, for our purpose, were found to be the PardisoLU and PardisoLLT solvers from the Intel MKL library, which are also supported by the Eigen library itself.

Chapter 5

Experimental Evaluation

5.1 Experimental Setup

To evaluate our method, we implemented a ROM tool that is able to load the desired models, produce the reduced systems by using either Algorithm 1 or Algorithm 2, and, finally, perform transient analysis on both the original and reduced systems to compare their results. For the evaluation process, we used benchmarks that were extracted from real electrical models with lots of mutual inductances, using an industrial tool, as well as the transient IBM power grid benchmarks[27]. Their characteristics are shown in Table 5.1, where we can see all the electrical elements of each circuit as well as the total size of the MNA matrices. Also, the names RLCk_1 to RLCk_3 represent the circuits from the industrial tool and ibmpg1t to ibmpg4t represent the transient IBM power grids.

Table 5.1: Circuit benchmarks and their characteristics

Benchmark	Total size	#nodes	#resistors	#capacitors	#inductors	#mutual ind.	#ports
RLCk_1	5431	3084	2998	1282	2347	136271	2
RLCk_2	21800	12166	34635	31131	9634	23639237	6
RLCk_3	39346	22059	51128	41871	17289	91627306	11
ibmpg1t	54265	25082	40801	10774	277	0	20
ibmpg2t	164897	37168	245163	36838	330	0	20
ibmpg4t	1214288	266906	1826589	265944	962	0	20

Our ROM tool uses a configuration file as input, where all the parameters for the process are specified by the user. These parameters include the input matrices, the desired order, the

tolerance for the convergence of the iterative process of Algorithm 2, as well as, the step and the endtime for the transient analysis. A detailed example of the configuration file is shown in Figure 5.1. While testing the circuits, we had to tailor these parameters for each one, in order to receive optimal results. In fact, for the transient analysis, we needed to have enough resolution to be able to compare the behaviour of the systems. In Table 5.2, we point out all the input parameters used for each of the circuits. For the implementation of the low-rank BT MOR procedure, discussed in Chapter 4, we decided to support both the usage of the standard Krylov subspace approach and the EKS approach, and compare their results.

Our goal was to achieve at least 99% reduction, while maintaining the deviation in simulation between the original and the reduced system at only 1%. The metrics to define the deviation of the transient responses of both systems were the percentages of Mean Relative Error (MRE) and Maximum Relative Error (MAX_RE). All experiments took place on a Linux workstation, equipped with an 8-core Intel Xeon Silver 4309Y processor at 2.8GHz and 64GB of memory.

```
set_working_directory              /benchmarks/
set_output_directory            output/test3_RLC_results/

// Cn
set_capacitance_file RLC/test3/Cn_mat.bin
// Gn
set_conductance_file RLC/test3/Gn_mat.bin
// M
set_inductance_file RLC/test3/M_mat.bin
// E
set_node_to_branch_file RLC/test3/E_mat.bin
// B
set_input_to_state_file RLC/test3/B_mat.bin

//cmd spec: set_threads  <number of threads>
set_threads             16

//cmd spec: set_desired_order  <size>
set_desired_order       385

// SIMULATION system and times //
//cmd spec: set_sim_system      <original|mor|all>
set_sim_system          all

//cmd spec: set_simulation      <step endtime>
set_simulation                le-19s  2e-17s

//cmd spec: set_tolerance <tol>
set_tolerance le-14

//cmd spec: set_mor_type < bt_init | bt_low_rank | bt_low_rank_eks >
set_mor_type bt_low_rank
```

Figure 5.1: Input configuration file example.

Table 5.2: Input parameters for the evaluation of each circuit

Benchmark	Desired order	Tolerance	Transient analysis	
			Step	Endtime
RLCk_1	54	1e-14	1e-19	2e-17
RLCk_2	210	1e-14	1e-19	2e-17
RLCk_3	363	1e-14	1e-19	2e-17
ibmpg1t	320	1e-14	1e-11	1e-8
ibmpg2t	500	1e-14	1e-11	1e-8
ibmpg4t	620	1e-14	1e-11	1e-8

5.2 Accuracy Results

The accuracy results are reported in Table 5.3, where ”ROM order” is the final size of the reduced systems after the lower rank solution was applied, MRE and MAX_RE are the deviation after running transient analysis, as mentioned above, and ”Reduction (%)” is the difference in size of the original to the reduced model.

Table 5.3: Reduction accuracy results between using standard and extended Krylov subspaces

Circuit	Standard Krylov subspace low-rank BT				EKS low-rank BT			
	ROM order	MRE (%)	MAX_RE (%)	Reduction (%)	ROM order	MRE (%)	MAX_RE (%)	Reduction (%)
RLCk_1	23	0.141	0.242	99.58	52	0.281	0.6	98.95
RLCk_2	110	0.107	0.288	99.49	102	0.116	0.431	99.53
RLCk_3	170	0.106	0.344	99.57	175	0.178	0.536	99.56
ibmpg1t	157	0.0022	0.07	99.71	177	0.0014	0.218	99.67
ibmpg2t	255	0.031	0.05	99.85	184	2.71e-5	0.0003	99.89
ibmpg4t	321	0.00079	0.036	99.97	121	0.01	0.07	99.99

By observing the accuracy results of Table 5.3, we can see that our proposed methodology offers a significant reduction, higher that 99% most of the times, with very acceptable errors in transient analysis. We also notice a difference in the behaviour of the method between the RLCk models and the IBM power grids, where the latter produced exceptional results with the use of the EKS method, in comparison to the standard Krylov subspace and vice versa. Specifically, our tool managed to achieve even a 99.99% reduction with only 0.01% MRE on the ibmpg4t benchmark using the EKS. On the other hand, the usage of the standard Krylov subspace provides, at worst, MRE lower than 0.141% with a reduction of over 99.49% across all benchamrks.

In general, we come to the conclusion that the reduced size does not depend that much

on the initial size of the system. Nevertheless, it is more relevant to the number of ports and the condition of the matrices G and C, which affect the number of iterations that it takes for the Algorithm 2 to converge and produce the smallest possible error.

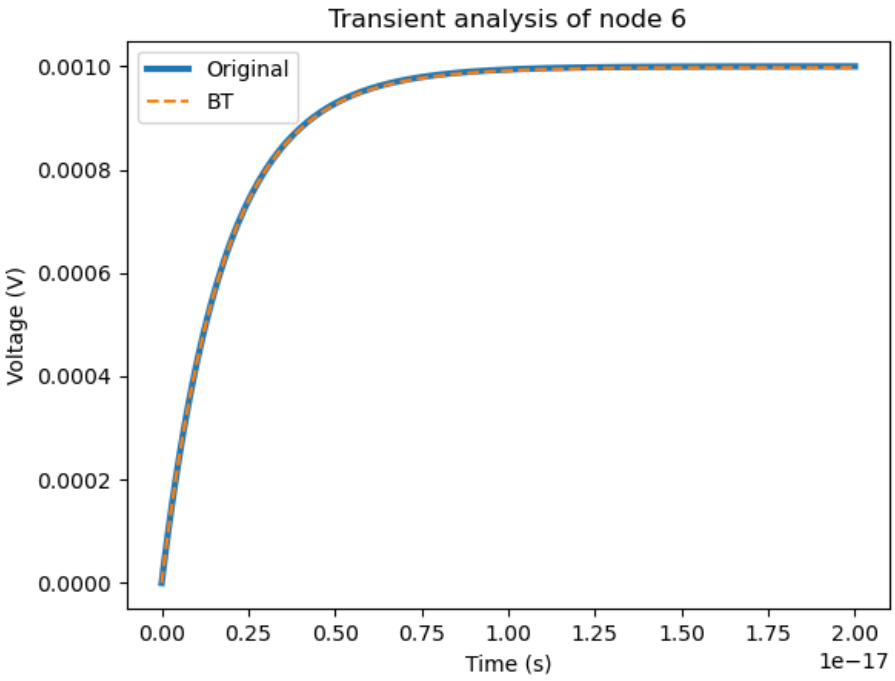

Figure 5.2: Comparison in transient analysis of RLCk_2 between original and reduced models at port 6 using the standard Krylov subspace.

5.3 Runtime and Memory Results

To examine the performance of our method, we calculated the execution time and peak memory of our tool, as presented in Table 5.4. Note that for all the results, in our tool, we used the PardisoLU as a sparse solver, which may be fast and accurate, but also consumes a lot of memory to achieve that. Again, we immediately notice the impact of the mutual inductances in the results, where the peak memory usage of the RLCk_2 circuit over-exceeds the one of the ibmpg2t benchmark, even though it has a much smaller amount of nodes. Moreover, the execution time is greatly affected by that dense matrix and can be a great challenge to achieve adequate performance for such circuits.

When we compared the use of the standard Krylov subspace to the EKS, in terms of performance, the EKS produced better results overall. We should mention, however, that

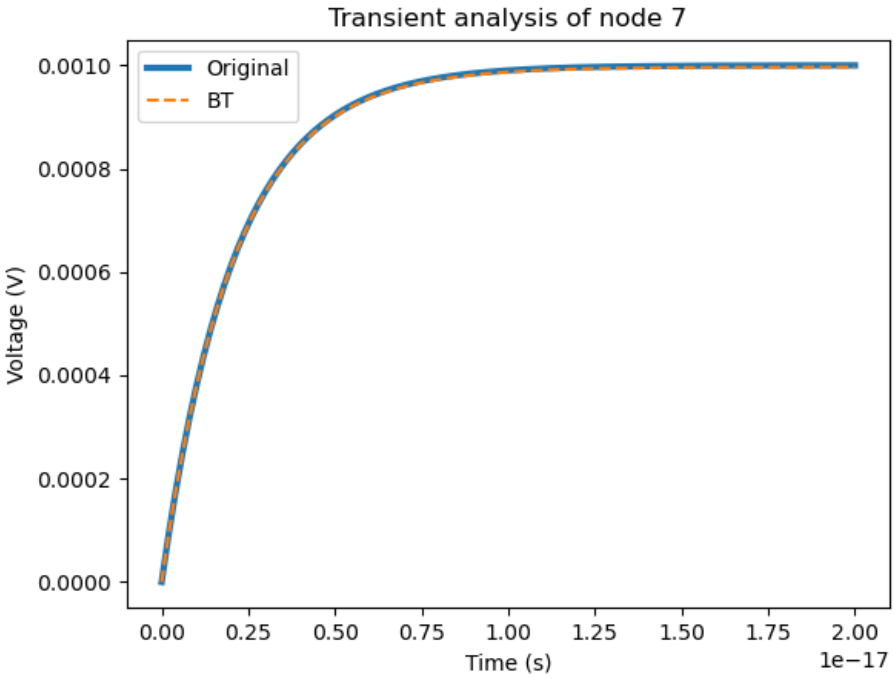

Figure 5.3: Comparison in transient analysis of RLCk_3 between original and reduced models at port 7 that demonstrates the produced 0.344% MAX_RE.

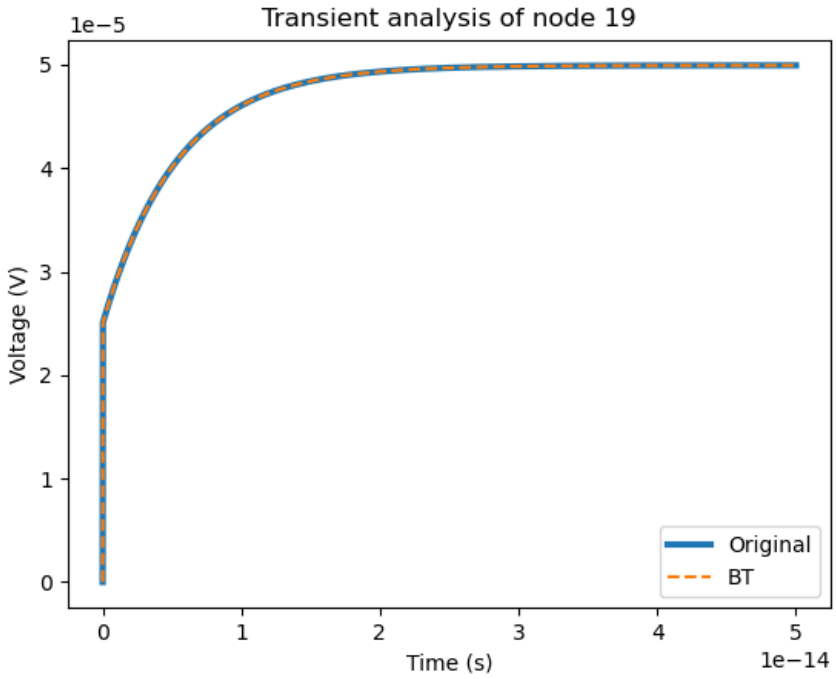

Figure 5.4: Transient response of ibmpg2t's port 19 between original and reduced models using EKS for 5e-14 s.

it takes only half iterations for the EKS to reach the same size compared to the standard Krylov. Also, the EKS method converges faster in most situations. This proves that the EKS method has leading performance, despite the fact that we have to deal with the inverse system matrices.

Table 5.4: Reduction performance results using our optimized implementation of the methods

Circuit	Standard Krylov subspace low-rank BT				EKS low-rank BT			
	Reduction time	Peak memory usage	Transient analysis time		Reduction time	Peak memory usage	Transient analysis time	
			Original	ROM			Original	ROM
RLCk_1	2.73 s	1.04 GB	11.89 s	2.23e-5 s	3.74 s	0.83 GB	11.89 s	5.42e-5 s
RLCk_2	108.61 s	12.22 GB	89.9 s	0.0001 s	74.29 s	12.26 GB	89.9 s	0.0001 s
RLCk_3	768.96 s	39.12 GB	406.4 s	0.0009 s	457.66 s	28.87 GB	406.4 s	0.0003 s
ibmpg1t	14.48 s	1.07 GB	3.73 s	0.0003 s	13.1 s	1.16 GB	3.73 s	0.00034 s
ibmpg2t	94.92 s	4.61 GB	18.33 s	0.00042 s	44.04 s	3.52 GB	18.33 s	0.00031 s
ibmpg4t	1098.26 s	41.21 GB	185.18 s	0.0006 s	165.8 s	16.95 GB	185.18 s	0.00014 s

Chapter 6

Conclusions

The implemented ROM generation tool handles real-world large-scale electromagnetic models, consisting of several thousands elements. The above mentioned tool, provides both the default BT and the improved low-rank implementation of the BT algorithm. Both methods were tested based on their performance and the accuracy that they offer. The implementation of BT and low-rank BT was based on Eigen library (C++) and was optimised by the exploitation of state-of-the-art solvers.

The overall experimental evaluation of the tool shows a clear improvement in model accuracy and performance, as well as it retains the benefits of specified error bounds. Efficient computational approaches have been provided, so as to improve in a greater degree the overall performance in runtime and memory. In general, the improved BT, using the standard Krylov subspace version, achieves model reduction at a scale of 99% with MRE of 0.141% and MAX_RE of 0.242% for an electromagnetic model of 5431 nodes and final reduced order of 23 nodes. For the same benchmark and the same method, we reach approximately 1GB peak memory usage, and reduction time of 2.73 seconds. The EKS approach of the improved BT, achieves an MRE of 0.282% and a MAX_RE of 0.6% for the same original benchmark (5432 nodes) with a reduced order of 52 nodes. The reduction time was risen to 3.74 seconds and the peak memory usage is estimated to 0.8GB.

Concluding the above, our ROM generation tool preserves the a-priori error offered by the BT method, while dealing with expensive computational tasks and storage limitations. At the same time, the performance, in runtime and memory, and the accuracy of the method is significantly improved, compared to that of the default BT algorithm.

Bibliography

[1] Wil M.P. van der Aalst. Data scientist: The engineer of the future, 2014.

[2] Allan Odabasioglu, Mustafa Celik, and Lawrence T. Pileggi. Prima: Passive reduced-order interconnect macromodeling algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17, 1998.

[3] Joel R. Phillips, Luca Daniel, and L. Miguel Silveira. Guaranteed passive balancing transformations for model order reduction. volume 22, 2003.

[4] Boyuan Yan, Sheldon X.D. Tan, and Bruce McGaughy. Second-order balanced truncation for passive-order reduction of rlck circuits. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 55, 2008.

[5] Dennis Fitzpatrick. Chapter 7 - transient analysis. In Dennis Fitzpatrick, editor, *Analog Design and Simulation Using OrCAD Capture and PSpice (Second Edition)*, pages 117–129. Newnes, second edition edition, 2018.

[6] Danish Rafiq and Mohammad Abid Bazaz. Model order reduction via moment-matching: A state of the art review, 2021.

[7] Lihong Feng, Peter Benner, and Jan G. Korvink. System-level modeling of mems by means of model order reduction (mathematical approximations) - mathematical background, 2013.

[8] Matthew B. Stephanson. High-order moment-matching mor with impedance boundaries for signal integrity analysis. *Applied Computational Electromagnetics Society Journal*, 33, 2018.

[9] Chrysostomos Chatzigeorgiou, Dimitrios Garyfallou, George Floros, Nestor Evmorfopoulos, and George Stamoulis. Exploiting extended krylov subspace for the reduc-

tion of regular and singular circuit models. In *2021 26th Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 773–778, 2021.

[10] Ulrike Baur, Peter Benner, and Lihong Feng. Model order reduction for linear and nonlinear systems: A system-theoretic perspective. *Archives of Computational Methods in Engineering*, 21, 2014.

[11] Lars Pernebo and Leonard M. Silverman. Model reduction via balanced state space representations. *IEEE Transactions on Automatic Control*, 27, 1982.

[12] A. C. Antoulas, D. C. Sorensen, and Y. Zhou. On the decay rate of hankel singular values and related issues. *Systems and Control Letters*, 46, 2002.

[13] Olympia Axelou, Dimitrios Garyfallou, and George Floros. Frequency-limited reduction of rlck circuits via second-order balanced truncation. In *SMACD / PRIME 2021; International Conference on SMACD and 16th Conference on PRIME*, pages 1–4, 2021.

[14] K. Gröchenig. Foundations of time-frequency analysis, 2003.

[15] *Numerical Methods in Electromagnetism*. 2000.

[16] B. Carpentieri, I. S. Duff, L. Giraud, and M. Magolu Monga Made. Sparse symmetric preconditioners for dense linear systems in electromagnetism. *Numerical Linear Algebra with Applications*, 11, 2004.

[17] K. Jbilou and A. J. Riquet. Projection methods for large lyapunov matrix equations. *Linear Algebra and Its Applications*, 415:344–358, 6 2006.

[18] K. Jbilou. Adi preconditioned krylov methods for large lyapunov matrix equations. *Linear Algebra and Its Applications*, 432:2473–2485, 5 2010.

[19] V. Simoncini. A new iterative method for solving large-scale lyapunov matrix equations. *SIAM Journal on Scientific Computing*, 29:1268–1288, 2007.

[20] Gene H. (Gene Howard) Golub and Charles F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1983.

[21] Ignacio Blanquer, Héctor Claramunt, Vicente Hernández, and Antonio M. Vidal. Solving the generalized lyapunov equation by the bartels-stewart method using standard

software libraries for linear algebra computations □. *IFAC Proceedings Volumes*, 31:387–392, 7 1998.

[22] Jeremy Levesley. Functions of matrices: Theory and computation. *Bulletin of the London Mathematical Society*, 41, 2009.

[23] Eugene L. Wachspress. Trail to a lyapunov equation solver. *Computers and Mathematics with Applications*, 55, 2008.

[24] Wei guo Wang and Yimin Wei. Mixed and componentwise condition numbers for matrix decompositions. *Theoretical Computer Science*, 681, 2017.

[25] L. Grasedyck. Existence of a low rank or □-matrix approximant to the solution of a sylvester equation. *Numerical Linear Algebra with Applications*, 11:371–389, 5 2004.

[26] Eigen. `http://eigen.tuxfamily.org`.

[27] Chong-Min. Kyung, ACM Digital Library., and ACM Special Interest Group on Design Automation. *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. IEEE Computer Society Press, 2008.