Mathematics in Industry 21
The European Consortium for Mathematics in Industry

Michael Günther *Editor*

Coupled Multiscale Simulation and Optimization in Nanoelectronics

Springer

MATHEMATICS IN INDUSTRY 21

More information about this series at
http://www.springer.com/series/4650

Michael Günther
Editor

Coupled Multiscale Simulation and Optimization in Nanoelectronics

Springer

Editor
Michael Günther
Lehrstuhl für Angewandte
Mathematik/Numerische Analysis
Bergische Universität Wuppertal
Wuppertal
Germany

ISSN 1612-3956 ISSN 2198-3283 (electronic)
Mathematics in Industry
ISBN 978-3-662-46671-1 ISBN 978-3-662-46672-8 (eBook)
DOI 10.1007/978-3-662-46672-8

Library of Congress Control Number: 2015941321

Mathematics Subject Classification Numbers (2010): 65M20, 65L80, 68N30, 78M34, 93A30, 97U50

Springer Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer-Verlag GmbH Berlin Heidelberg is part of Springer Science+Business Media (www.springer.com)

Angelo Marcello Anile (1948–2007)

*Dedicated to Angelo Marcello Anile,
the scientist, colleague, and dear friend.
Without his enthusiasm, mediative attitude,
and extensive knowledge, COMSON would
never have been possible.*

Preface

Circuit design based on numerical simulation relies heavily on mathematical methods. As a result, relations have long since been established between the microelectronics industry and university groups specializing in simulations for semiconductor processes and devices, electromagnetics and electronic circuits. State-of-the-art methods from the fields of applied and numerical analysis, as well as newly developed dedicated algorithms, have facilitated the large-scale use of simulations, thereby enabling the industry to reach its current high state of the art.

Designing complex integrated circuits calls for adequate simulation and optimisation tools. The current design approach involves simulations and optimizations in different physical domains (device, circuit, thermal, electromagnetic) and in electrical engineering disciplines (logic, timing, power, crosstalk, signal integrity, system functionality). The physical aspects are essential to characterizing circuit behavior from an electrical engineering and system-oriented standpoint.

Accordingly, the main scientific objectives of the COMSON (COupled Multiscale Simulation and Optimization in Nanoelectronics) project were as follows:

- To develop new descriptive models that take these mutual dependencies into account
- To combine these models with existing circuit descriptions in new simulation strategies
- To develop new optimization techniques that will accommodate new designs

COMSON was a Marie Curie Research Training Network supported by the European Commission in the programme Structuring the European Research Area, part of the EU's Sixth Framework Research Programme. The project was initiated by the three major European semiconductor companies – Infineon Technologies AG, later replaced by its spin-off Qimonda AG of Neubiberg, Germany; Koninklijke Philips N.V., later replaced by its spin-off NXP Semiconductors Netherlands N.V. of Eindhoven, the Netherlands; and STMicroelectronics of Catania, Italy – who worked in cooperation with five European academic partners in Applied Mathematics and Electrical Engineering with considerable experience in the simulation and optimization of integrated circuits – the University of Wuppertal, Germany (coordinator); Eindhoven University of Technology, the Netherlands; University of Catania, Italy; University of Calabria, Italy; and University Politehnica of Bucharest, Romania. The rationale behind the project and this book was described as follows:

> Performing the step from micro- to nanoelectronics, the semiconductor industry is confronted with very high levels of integration, introducing coupling effects that were not observed before. Currently, the complexity of this problem is beyond the capabilities of any industrial software and design environment. Furthermore, in the near future, researchers must understand all aspects of the problems faced by industry.
>
> To meet these new scientific and training challenges, the COMSON project on "COupled Multiscale Simulation and Optimization in Nanoelectronics" merges the know-how of the three major European semiconductor companies with the combined expertise of university groups specialized in developing adequate mathematical models, numerical schemes, and e-learning facilities, covering all relevant fields of interest. In COMSON, academia and industry join their efforts to realize a common Demonstrator Platform: on the one hand, to test mathematical methods and approaches, so as to assess whether they are capable of addressing the industry's problems; on the other hand, to adequately educate young researchers by providing hands-on experience with state-of-the-art problems, and beyond.

The editor thanks his colleagues for their valued contributions in the different chapters of this handbook: Roland Pulch of Greifswald, Germany (PDAE modelling); Andreas Bartel of Wuppertal, Germany, and Sebastian Schöps of Darmstadt, Germany (dynamic iteration); E.J.W. ter Maten of Eindhoven, the Netherlands (MOR); Salvatore Rinaudo of Catania, Italy (optimization); Georg Denk of Munich, Germany (demonstrator platform); and Giuseppe Alì of Cosenza, Italy (e-learning).

Wuppertal, Germany
September 2014

Michael Günther

Contents

Part I Introduction

1 The COMSON Project 3
Michael Günther and Uwe Feldmann
1.1 Trends in Microelectronics 3
1.2 Scope of the COMSON Project 4
1.3 Methodology 5
1.3.1 The Demonstrator Platform 6
1.3.2 E-Learning 9
1.4 Modelling, Simulation and Optimisation 9
1.4.1 Partial Differential Algebraic Equations 10
1.4.2 Dynamic Iteration 10
1.4.3 Model Order Reduction 11
1.4.4 Optimisation 11
References 12

Part II Partial Differential Algebraic Equations

2 PDAE Modeling and Discretization 15
Giuseppe Alì, Massimiliano Culpo, Roland Pulch, Vittorio Romano, and Sebastian Schöps
2.1 Introduction on Modeling and PDAEs 15
2.1.1 Mathematical Modeling in Nanoelectronics 16
2.1.2 Classification of PDAE Models 18
2.2 Modeling, Analysis and Discretization of Coupled Problems 21
2.2.1 Refined Modeling of Networks with Devices 21
2.2.2 Electro-Thermal Effects at the System Level 45
2.2.3 Multiphysics Modeling via Maxwell’s Equations 64
2.2.4 Thermal and Quantum Effects in Semiconductors 78
References 99

3 Simulation of Coupled PDAEs: Dynamic Iteration and Multirate Simulation 103
Giuseppe Alì, Andreas Bartel, Michael Günther, Vittorio Romano, and Sebastian Schöps
3.1 Aim and Outline 103
3.2 Theory of Dynamic Iteration Schemes for Coupled DAEs 104
3.2.1 Description of Coupled Systems 105
3.2.2 Iteration Schemes for Coupled DAE Systems 107
3.2.3 Convergence and Stability 112
3.3 Applications in Electrical Engineering 124
3.3.1 Refined Network Models 125
3.3.2 Electro-Thermal Coupling 128
3.3.3 Coupled System of Electric Networks and Maxwell's Magnetoquasistatic Equations and Their Properties 129
3.4 Coupled Numerical Simulations of the Thermal Effects in Silicon Devices 137
3.4.1 The Numerical Method 140
3.4.2 Numerical Simulation of the Crystal Lattice Heating in MOSFETs 145
3.4.3 Coupled Circuit-Device Simulation 153
References 154

Part III Model Order Reduction

4 Model Order Reduction: Methods, Concepts and Properties 159
Athanasios C. Antoulas, Roxana Ionutiu, Nelson Martins, E. Jan W. ter Maten, Kasra Mohaghegh, Roland Pulch, Joost Rommes, Maryam Saadvandi, and Michael Striebel
4.1 Circuit Simulation and Model Order Reduction 163
4.1.1 Input-Output Systems in Circuit Simulation 167
4.1.2 The Need for Model Order Reduction 168
4.1.3 MOR Methods 174
4.1.4 Projection Based MOR 174
4.1.5 Truncation Based MOR 181
4.1.6 Other Approaches 187
4.1.7 Examples 187
4.1.8 Summary 192
4.2 Eigenvalue Methods and Model Order Reduction 193
4.2.1 Transfer Functions, Dominant Poles and Modal Equivalents 194
4.2.2 Specialized Eigensolution Methods 196
4.2.3 Generalizations to Other Eigenvalue Problems 206
4.2.4 Improving Krylov Models by Using Dominant Poles 209

4.2.5 Numerical Examples 211
4.2.6 Concluding Remarks 213
4.3 Passivity Preserving Model Order Reduction 214
4.3.1 Model Reduction via Projection Matrices 216
4.3.2 Passive Systems 217
4.3.3 Spectral Zeros and Generalized Eigenvalue Problem 219
4.3.4 Passivity Preserving Model Reduction 221
4.3.5 Model Reduction by Projection 222
4.3.6 Model Reduction by Projection 226
4.3.7 Numerical Results 228
4.3.8 Conclusion 228
4.4 Passivity Preserving Model Reduction Using the Dominant Spectral Zero Method 230
4.4.1 Background on MOR 231
4.4.2 MOR by Spectral Zero Interpolation 231
4.4.3 The Dominant Spectral Zero Method 232
4.4.4 Numerical Results 235
4.4.5 Concluding Remarks 237
4.4.6 Appendix: SADPA for Computing Dominant Spectral Zeros 239
4.5 A Framework for Synthesis of Reduced Order Models 241
4.5.1 Problem Formulation 243
4.5.2 Foster Synthesis of Rational Transfer Functions 243
4.5.3 Structure Preservation and Synthesis by Unstamping 245
4.5.4 Numerical Examples 249
4.5.5 Conclusions and Outlook 255
References 257

5 Parameterized Model Order Reduction 267
Gabriela Ciuprina, Jorge Fernández Villena, Daniel Ioan, Zoran Ilievski, Sebastian Kula, E. Jan W. ter Maten, Kasra Mohaghegh, Roland Pulch, Wil H.A. Schilders, L. Miguel Silveira, Alexandra Ştefănescu, and Michael Striebel
5.1 Parametric Model Order Reduction 269
5.1.1 Representation of Parametric Systems 270
5.1.2 Reduction of Parametric Systems 271
5.1.3 Practical Consideration and Structure Preservation 278
5.1.4 Examples 280
5.1.5 Conclusions 284
5.2 Robust Procedures for Parametric Model Order Reduction of High Speed Interconnects 287
5.2.1 Field Problem Formulation: 3D – PDE Models 288
5.2.2 Numeric Discretization and State Space Models 290
5.2.3 Transmission Lines: 2D + 1D Models 293
5.2.4 Numeric Extraction of Line Parameters 297

5.2.5 Variability Analysis of Line Parameters 299
5.2.6 Parametric Models Based on Taylor Series 301
5.2.7 Parametric Circuit Synthesis 303
5.2.8 Case Study 314
5.2.9 Conclusions 317
5.3 Model Order Reduction and Sensitivity Analysis 319
5.3.1 Recap MNA and Time Integration of Circuit Equations 320
5.3.2 Sensitivity Analysis 322
5.3.3 Model Order Reduction (with POD) 328
5.3.4 The BRAM Algorithm 331
5.3.5 Sensitivity by Uncertainty Quantification 333
5.4 MOR for Singularly Perturbed Systems 341
5.4.1 Model Order Reduction and ε-Embedding 343
5.4.2 Test Example and Numerical Results 348
5.4.3 Conclusions 350
References 351

6 Advanced Topics in Model Order Reduction 361
Davit Harutyunyan, Roxana Ionutiu, E. Jan W. ter Maten, Joost Rommes, Wil H.A. Schilders, and Michael Striebel
6.1 Model Order Reduction of Nonlinear Network Problems 362
6.1.1 Linear Versus Nonlinear Model Order Reduction 363
6.1.2 Some Nonlinear MOR Techniques 365
6.1.3 TPWL and POD 367
6.1.4 Other Approaches 377
6.1.5 Numerical Experiments 377
6.2 Model Order Reduction for Multi-terminal Circuits 380
6.2.1 Reduction of *R* Networks 381
6.2.2 Reduction of *RC* Networks 385
6.2.3 Numerical Results 390
6.2.4 Concluding Remarks 398
6.3 Simulation of Mutually Coupled Oscillators Using Nonlinear Phase Macromodels and Model Order Reduction Techniques 398
6.3.1 Phase Noise Analysis of Oscillator 400
6.3.2 LC Oscillator 401
6.3.3 Mutual Inductive Coupling 404
6.3.4 Resistive and Capacitive Coupling 406
6.3.5 Small Parameter Variation Model for Oscillators 407
6.3.6 Oscillator Coupling with Balun 408
6.3.7 Oscillator Coupling to a Transmission Line 411
6.3.8 Model Order Reduction 412
6.3.9 Numerical Experiments 413
6.3.10 Conclusion 425
References 426

Part IV Optimization

7 Optimization Methods and Applications to Microelectronics CAD ... 435
Salvatore Rinaudo, Valeria Cinnera Martino, Franco Fiorante, Giovanni Stracquadanio, and Giuseppe Nicosia
7.1 Motivation ... 435
7.2 The Optimization Problem ... 437
7.3 Parameter Extraction for Compact Circuit Models ... 438
7.3.1 Automatic Physical Layout Optimization of Discrete Power MOSFETs for Reducing the Effects of Current Density ... 438
7.3.2 Modeling Approach ... 439
7.4 The Optimization Algorithm ... 441
7.4.1 The Optimization Flow ... 444
7.4.2 Simulation Results ... 446
7.5 Conclusion ... 449
References ... 449

Part V COMSON Methodology

8 COMSON Demonstrator Platform ... 455
Georg Denk, Tamara Bechtold, Massimiliano Culpo, Carlo de Falco, and Alexander Rusakov
8.1 Introduction ... 456
8.1.1 Design of the *Demonstrator Platform* ... 456
8.1.2 Modules ... 458
8.2 Tutorial on Working with the *Demonstrator Platform* with Emphasis on MOR ... 460
8.2.1 Overview and Structure of MOR Modules ... 461
8.2.2 MOR Case Studies ... 463
8.2.3 A Step by Step Model Reduction Tutorial ... 470
8.2.4 Coupled Simulation (MOR + Circuit Simulation) ... 480
8.3 The *Demonstrator Platform* as a Development Tool for Research ... 485
8.3.1 Class “C” Benchmark: n-Channel Power MOS-FET ... 486
8.3.2 Implementation and Development Procedure ... 490
8.3.3 Simulation Set-Up and Results on the Class “C” Benchmark ... 494
References ... 500

9 eLearning in Industrial Mathematics with Applications to Nanoelectronics 503
Giuseppe Alì, Eleonora Bilotta, Lorella Gabriele, Pietro Pantano, José Sepúlveda, Rocco Servidio, and Alexander Vasenev
9.1 Introduction 503
9.2 An Overview on eLearning 507
9.2.1 An Historic Perspective 507
9.2.2 eLearning in Microelectronics Industry 510
9.3 An Integrated Platform for Advanced Training in Microelectronics 513
9.3.1 User Needs Analysis 516
9.3.2 Platform Development 519
9.4 The Components of the CoMSON Platform 521
9.4.1 The Information System 521
9.4.2 The CoMSON eLearning Platform 523
9.5 Graphical User Interfaces 526
9.5.1 User Interfaces in the eLearning Platform 527
9.5.2 A Graphical Tool to Visualize Scientific Data in the Simulation Platform 528
9.5.3 Interfacing the Components of the CoMSON Platform 529
9.6 eLearning Contents Creation: Methods and Strategies 532
9.6.1 General Methods and Strategies for the Development of an eLearning Course 533
9.6.2 Some Examples of Course Implementation 534
9.6.3 Practical Strategies for eLearning Contents Creation 537
9.7 Blended Learning 545
9.7.1 First Empirical Study 545
9.7.2 Second Empirical Study 548
9.8 Platform Evaluation: Test and Revision 550
9.9 Conclusions and Future Perspectives 553
References 555

Index 561

List of Contributors

Giuseppe Alì Department of Physics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
Department of Mathematics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
INFN, Gruppo coll. Cosenza, Arcavacata di Rende, Cosenza, Italy

Athanasios C. Antoulas Department of Electrical and Computer Engineering, William Marsh Rice University, Houston, TX, USA
School of Engineering and Science, Jacobs University Bremen, Bremen, Germany

Andreas Bartel Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Wuppertal, Germany

Tamara Bechtold IMTEK – University of Freiburg, Freiburg, Germany

Eleonora Bilotta Department of Physics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
Department of Linguistics, University of Calabria, Arcavacata di Rende, Cosenza, Italy

Gabriela Ciuprina Politehnica University of Bucharest, Bucharest, Romania

Massimiliano Culpo Università di Bologna, Bologna, Italy
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Wuppertal, Germany

Georg Denk Infineon Technologies AG, München, Germany

Carlo de Falco MOX – Modeling and Scientific Computing, Dipartimento di Matematica, Politecnico di Milano, Milano, Italy
CEN – Centro Europeo di Nanomedicina, Milano, Italy

Uwe Feldmann Formerly at Qimonda AG, Munich, Germany

Franco Fiorante STMicroelectronics Catania, Italy

Lorella Gabriele Department of Physics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
Department of Linguistics, University of Calabria, Arcavacata di Rende, Cosenza, Italy

Michael Günther Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Wuppertal, Germany

Davit Harutyunyan ASML, DR Veldhoven, The Netherlands

Zoran Ilievski European Space & Technology Centre, AG Noordwijk, The Netherlands

Daniel Ioan Politehnica University of Bucharest, Bucharest, Romania

Roxana Ionutiu ATTE, ABB Switzerland Ltd, Austraße, Turgi, Switzerland

Sebastian Kula Institute of Mechanics and Applied Computer Science, Kazimierz Wielki University, Bydgoszcz, Poland

Nelson Martins CEPEL, Rio de Janeiro, RJ, Brazil

Valeria Cinnera Martino, STMicroelectronics Catania, Italy

E. Jan W. ter Maten Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Wuppertal, Germany
Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, Eindhoven, The Netherlands

Kasra Mohaghegh Multiscale in Mechanical and Biological Engineering (M2BE), Aragón Institute of Engineering Research (I3A), University of Zaragoza, Zaragoza, Spain

Giuseppe Nicosia Department of Mathematics and Computer Science, University of Catania, Catania, Italy

Pietro Pantano Department of Physics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
Department of Mathematics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
INFN, Gruppo coll. Cosenza, Arcavacata di Rende, Cosenza, Italy

Roland Pulch Institut für Mathematik und Informatik, Ernst Moritz Arndt Universität Greifswald, Greifswald, Germany

Salvatore Rinaudo STMicroelectronics Catania, Italy

Vittorio Romano Dipartimento di Matematica e informatica, Università di Catania, Catania, Italy

Joost Rommes Mentor Graphics, DSM/AMS, Le Viseo – Bâtiment B, Montbonnot, France

Alexander Rusakov Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IPPM RAS), Moscow, Russian Federation

Maryam Saadvandi Numerical Approximation and Linear Algebra Group, Department of Computer Science, KU Leuven, Heverlee, Belgium

Wil H.A. Schilders Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, Eindhoven, The Netherlands

Sebastian Schöps Graduate School of Excellence Computational Engineering, TU Darmstadt, Darmstadt, Germany

José Sepúlveda Applied Research & Technology for Infocomm Centre – Singapore Polytechnic, Industry Liaison Office – National University of Singapore, Singapore, Singapore
Department of Mathematics, University of Calabria, Arcavacata di Rende, Cosenza, Italy
INFN, Gruppo coll. Cosenza, Arcavacata di Rende, Cosenza, Italy

Rocco Servidio Department of Languages and Education Sciences, University of Calabria, Arcavacata di Rende, Cosenza, Italy
Department of Linguistics, University of Calabria, Arcavacata di Rende, Cosenza, Italy

L. Miguel Silveira INESC ID/IST – TU Lisbon, Lisbon, Portugal

Alexandra Ştefănescu Research Centre of Excellence “Micro- and nanosystems for radiofrequency and photonics”, IMT Bucharest – National Institute for Research and Development in Microtechnologies, Bucharest, Romania

Giovanni Stracquadanio Department of Mathematics and Computer Science, University of Catania, Catania, Italy

Michael Striebel ZF Lenksysteme GmbH, Schwäbisch Gmünd, Germany

Alexander Vasenev Department of Engineering and Construction Management, Twente University, Enschede, The Netherlands
Numerical Methods Laboratory, “Politehnica” University of Bucharest, Bucharest, Romania

Jorge Fernández Villena INESC ID/IST – TU Lisbon, Lisbon, Portugal

Part I
Introduction

Chapter 1
The COMSON Project

Michael Günther and Uwe Feldmann

Abstract This chapter serves as an introduction into the outcome of the COMSON project, and links the subsequent chapters to the overall idea of COMSON and its objectives. We start with a discussion of the state-of-the-art and open problems in nanoelectronics simulation at the timepoint when the COMSON Project was started. Therefrom the main scientific objectives of the COMSON project are derived. Special attention is devoted to a uniform methodology for both testing the new achievements and simultaneously educating young researchers: All mathematical codes are linked into a new Demonstrator Platform (Chap. 8), which itself is embedded into an E-Learning environment (Chap. 9). Subsequently the scientific objectives are shortly reviewed. They comprise: (i) Development of new coupled mathematical models, capturing the mutual interactions between the physical domains of interest in nanoelectronis. These are based on the PDAE approach (Chap. 2). (ii) Investigation of numerical methods to simulate these models. Our focus is on dynamic iteration schemes (Chap. 3) and for efficiency on MOR techniques (Chaps. 4–6). (iii) Usage of models and simulation tools for optimal design of nanoelectronic circuits by means of multi-objective optimisation in a compound design space (Chap. 7).

1.1 Trends in Microelectronics

The design of complex integrated circuits ICs requires adequate simulation and optimisation tools. The current design approach involves simulations and optimisations in different physical domains (device, circuit, thermal, electromagnetic) as well as in electrical engineering disciplines (logic, timing, power, crosstalk, signal

M. Günther (✉)
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal,
Gaußstraße 20, D-42119 Wuppertal, Germany
e-mail: guenther@math.uni-wuppertal.de

U. Feldmann
Formerly at Qimonda AG, Munich, Germany
e-mail: uwe.feldmann@online.de

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_1

integrity, system functionality). Our interests focus on the physical aspects, which are fundamental for characterising circuit behavior on an electrical engineering and system oriented level. To limit the complexity of the design task, these domains are currently treated in isolation ("divide and conquer" approach), and dedicated simulation and optimisation tools have been developed for the individual domains. However, this methodology approaches its limits of validity. As semiconductor technology is progressing down to the nanometer regime, it turns out that the complexity in simulating and optimising designs goes beyond the capabilities of the software and design environments used so far. Several shortcomings are clearly visible:

- With ever smaller characteristic dimensions, higher operating frequency, and increasing power density many simplifying assumptions are losing their validity. Particularly, coupling effects between the different physical domains as well as 2D/3D and higher order nonlinear effects have to be taken into account.
- Due to very high levels of integration, simulation times are becoming prohibitively long because of growing problem size and coupling effects.
- More complex design specifications have to be satisfied in a widely extended design and parameter space, while simulations for assessing a design with a given parameter configuration get more costly.

Clearly, substantial progress in this situation is not possible by just improving the single components of the design system being used, and this observation led to the setup of the COMSON project.

1.2 Scope of the COMSON Project

The COMSON project was initiated by three major European semiconductor companies in cooperation with five academic partners from Europe being experienced in simulation and optimisation of integrated circuits. The primary objective was to combine the expertise distributed over the partner nodes in their particular fields in a joint effort, to get a more global progress for the coupled systems as a whole.

Mathematical modelling and the development of numerical methods were seen as key enablers in this project. To cope with the coupled nature of problems, it was planned to pursue cosimulation strategies, where the different domains are described by Partial Differential-Algebraic Equations PDAEs or ordinary Differential-Algebraic Equations DAEs, which are – as far as possible – simply coupled by source terms or boundary conditions. For their numerical solution dynamic iteration schemes were appealing, since they naturally exploit the widely separated spectrum of time scales inherent in the various domains.

As a promising side effect of this approach it was seen that it offers to replace parts of the huge coupled system by reduced order models at least for some of the domains. So, linear and nonlinear Model Order Reduction MOR became another essential part of research in COMSON.

Finally, multi-objective optimisation in the very complex design space formed the third mathematical item of research.

The global view introduced in COMSON by coupling domains in simulation and optimisation did not only stimulate mathematical research, but also imposed two methodological problems:

- How can the new developments be assessed at hand of real life industrial designs, without implementing them into all of the commercial design tools used by the industrial partners?
- How can the transfer of knowledge be organised to assure that a researcher working at a – possibly multi-domain – coupled problem has the background information about all of the domains being involved?

For the COMSON project, these questions were answered by the decision to include the development of a software *Demonstrator Platform* into the project, as well as an E-Learning environment into which both the *Demonstrator Platform* and the real life applications foreseen as a reference problem are embedded.

In total the scope of the COMSON project comprises

- Mathematical research on modelling and discretisation of coupled PDAE systems, model order reduction, and optimisation
- And a methodological part by linking a new *Demonstrator Platform* for coupled simulation and benchmark problems of industrial relevance into an E-Learning environment.

The project name COMSON is derived from this scope: "**CO**upled **M**ultiscale **S**imulation and **O**ptimisation in **N**anoelectronics". The following sections will give a more detailed introduction into the single parts.

1.3 Methodology

In the following we explain the methodology (linkage of a *Demonstrator Platform* and E-Learning environment) used for both testing mathematical methods and educating young researchers.

Since the general scope of COMSON was too comprehensive for the restricted project time, research and development were focused on solving a few benchmark problems. The latter were specified by the industries, close to actual real life designs of medium complexity. Academic abstractions and simplifications should be avoided. Hence actual technological data and design specifications were to be used, and physical models as well as compact transistor models being state-of-the-art have been taken as a reference.

Even though there were only a few benchmark problems specified, their simulation and optimisation requires to couple all of the domains which had been considered to be relevant: Semiconductor devices, circuits, interconnects, electromagnetic EM fields, and heat flow. To this end the *Demonstrator Platform* concept was introduced, to provide an experimental framework in software for coupled simulation of the various domains. This gives excellent opportunities to test new numerical methods even in an early stage, and to make sure that they contribute to handle the coupled problems of interest. At the end, the *Demonstrator Platform* offers all coupled simulation capabilities being necessary for multi-domain optimisation of the benchmark problems. To realise this concept, and to demonstrate its functionality, became a key objective of the project.

Another methodological aspect was to provide means for rapid dissemination of knowledge over the geographically widespread partner nodes of the project. Somehow, every project member had been active in this field before, however with different focus and target applications. Now, since all of the partners were starting towards the same objectives – namely to develop and implement methods for coupled simulation and optimisation of the benchmark problems specified by industry – quick and reliable exchange of knowledge became very essential for the project. Having the complexity of multi-coupled simulation and of advanced design specifications as well as the different status of knowledge of researchers in mind, the COMSON members were convinced about the needs to include E-Learning facilities into the project. A natural step at this stage was the decision to embed the *Demonstrator Platform* into the E-Learning environment. This opened very flexible and valuable means for researchers, at any level of experience, to learn about models, methods and backgrounds of coupled nanolectronics simulation and design.

1.3.1 The Demonstrator Platform

1.3.1.1 Objectives and Benefits

The main objective of the *Demonstrator Platform* was to provide an experimental software platform for coupled simulation, which serves as a testbench for new models and methods, and finally offers an adequate simulation tool for optimisation of the benchmark design problems in a compound design space.

By the rule to integrate their new developments – be it model codes or mathematical methods – into the platform, the researchers get a natural test bench with state-of-the-art models and parameters from the different domains, rather than academic simplifications. And they get immediate feedback on the capability to address problems of industrial relevance. Furthermore, it is assured that the individual contributions seamlessly integrate into the whole system from the early beginning.

Another benefit of such a platform is to collect all knowledge about models, methods, and coupling principles. This way a homogeneous embedding into an E-Learning environment becomes possible, thus offering excellent opportunities for transfer of knowledge and mutual stimulation of new research.

1.3.1.2 State of the Art

Since the development of the coupled device/circuit simulators MEDUSA [5] and CODECS [10], there is a long tradition in coupling *two* domains in one code. At present there are powerful commercial tools like the platforms MEDICI (Synopsys Inc.) and ATLAS (Silvaco International) for coupled device/circuit simulation in use. However, they aim at device engineering and device characterisation with a very limited number of transistors. Hence they cannot be used for designing integrated circuits of a medium size complexity, nor do they allow for experiments with new mathematical algorithms from outside the software companies.

Coupling of device and circuit problems under a rigorous PDAE framework was introduced in [15, 17]; this served as a basis for the work to be done here.

Signal propagation effects have a large impact on integrated circuits performance, in general, and therefore coupled interconnect/circuit simulation is widely practised since a long time. Roughly, there are two mainstreams: One is to solve the telegraphers equations for coupled interconnect lines analytically under some simplifying assumptions, ending up in a transmission line (T-Line) model being built from controlled sources for circuit simulation [12]. The other one is to split the interconnect lines into small pieces, which are modelled by lumped R-, L-, and C-elements for circuit simulation. Due to mutual coupling, the corresponding resistance/conductance, inductance, and capacitance matrices are very large in general, and almost dense. Therefore some kind of network reduction or MOR is applied before including them into circuit simulation [2, 16].

Fully bidirectional coupling of interconnect and circuit simulation is reported in several papers, see e.g. [8, 11, 13], and the PDAE setting of this coupled problem was introduced in [9].

Coupling from EM field simulation to circuit simulation is well established in the literature and in industrial practice, however often under restrictive assumptions. Most approaches pursue the concept of partially equivalent electrical circuits (PEEC) developed by A.E. Ruehli [14], and apply linear MOR techniques for getting circuit models of a reasonable size. Alternatively, field simulators often generate scattering parameters (S-parameters) for an electro-magnetic component, which are used in circuit simulation.

Closer coupling between EM field and circuit simulation is necessary for handling the substrate noise problem in mixed-signal ICs [4], and for analysing mutual interaction of on-chip integrated passives (inductors) with semiconductor devices on radio frequency RF chips. To this end some powerful commercial tools have been developed by the companies Magwel and Sonnet Software, for example.

The coupling of the circuit domain with the thermal domain is straightforward, in principle, since due to the electrothermal analogy any circuit simulator can be "misused" for analysing thermal problems, once the latter are modelled by lumped elements [7]. This kind of coupled simulation is often done in practice. For small sized problems the more general approach of directly coupling a 2D or 3D thermal solver and a circuit simulator was pursued, see e.g. [19]. Finally, a general PDAE oriented framework for coupling thermal and circuit problems was developed in [3].

The coupling of the device and the thermal domain was mainly driven by power electronics applications, and started in the late 1970s [1]. While in the beginning the coupling terms were pretty simple, more consistent models evolved since 1990 [18]. Overall, this kind of coupling has found much attention, and is very well developed.

As an extension of the electro-thermal analogy to other physical domains, the simulator fREEDA [6] was developed for simulation of coupled multiphysics problems in an open source project. It is based on a flexible modeling concept, such that a network built from elements from different physical domains can be brought into equilibrium under an energy norm. Clearly, the scope of this approach is on the physical modeling side, while ours is more focused on mathematical analysis and numerics of coupling existing physical models.

In summary it can be stated that bilateral coupled simulation has been extensively investigated, and is implemented in a variety of tools and models which are used in academic and industrial practice. However, simultaneous coupling of all the domains which are addressed here under a common mathematical framework of DAEs/PDAEs, and with inclusion of Model Order Reduction is new, to our knowledge. Furthermore, we are not aware of any other attempt to tightly embed a software package for coupled simulation in multiple domains into an E-Learning environment, for the ease and flexibility of transfer of knowledge.

1.3.1.3 Basic Concepts

To achieve an optimal design in the very complex design space, a multi-objective optimiser will interact with a simulation platform which provides consistent data about all parts of the design specifications, inclusive their mutual dependencies. To this end the platform operates on a hierarchy of parameterised subdomains, which are connected in a common network as a carrier. In the simplest case, the subdomains on top level are electric (sub)circuits. The subdomains on the lower levels can either be other subcircuits, or semiconductor devices, or interconnects, or EM domains, or thermal domains, or Reduced Order Models ROMs for one of these domains (see Fig. 1.1).

The network approach implies coupling of domains by source terms or boundary conditions. This will not be flexible enough in certain cases, hence the subdomains may constitute internally coupled problems by themselves. However, with the network approach it requires less efforts in general to include existing model codes. Furthermore, it is well suited for mathematical analysis and development of numerical methods.

Mathematically, the coupling of domains in a network means to couple partial differential equations PDEs or differential algebraic equations DAEs by algebraic or differential algebraic equations, thus getting PDAE systems. The concept is to solve them by co-simulation in dynamic iteration schemes. To cope with the complexity, comprehensive physical subdomain models must be substituted by ROMs. Noteably, the ROMs should be parameterised, in order to be efficient along several steps of the optimisation process. For the same objective it is an important aspect of the models

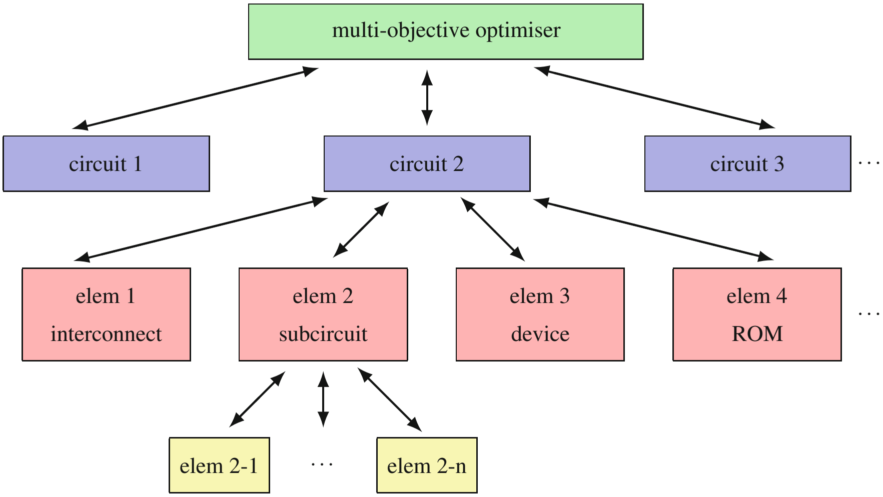


Fig. 1.1 The *Demonstrator Platform* is working on a hierarchy of parameterised subdomains

and codes to provide efficient calculation of sensitivities. Finally, for estimation of yield, an efficient handling of technological spread is a prerequisite.

1.3.2 E-Learning

One of the main aims of the CoMSON project was to define and to develop a system of E-Learning in Industrial Mathematics with applications to Microelectronics, in order to facilitate the exchange of information; to share resources, scientific and educational materials; to create common standards; to facilitate the use of advanced tools. The common idea of this project was to create a bridge being able to fill the gap that exists in the knowledge flow from University to Industry and vice-versa, and to overcome problems due to Intellectual Property claims raised by the Industries working together in the project.

1.4 Modelling, Simulation and Optimisation

The modelling is based on the PDAE approach. For numerical simulation efficient methods have to be used, applying dynamic iteration schemes and MOR techniques. Based on these models and simulation tools, multi-objective optimisation is addressed.

1.4.1 Partial Differential Algebraic Equations

Up to now, mathematical research has been mainly focused on models of one single domain, e.g. semiconductor equations. Including effects of other domains like thermal and electromagnetic coupling and high frequency aspects to improve the accuracy of the models results in so-called *Partial Differential-Algebraic Equations* (PDAEs), which couple differential-algebraic network models for lumped descriptions and partial differential equations for the spatially distributed elements and effects via source terms or boundary conditions. This approach requires new analysis with respect to consistency and validity of the overall PDAE model that links different domains and levels of physical description, existence of solutions, and robustness and efficiency of the numerical methods being applied for solving the extended sets of equations.

New, robust and efficient methods are needed to solve the resulting equations. Depending on the type of coupling and accuracy to be achieved within simulation, two approaches are feasible to cope with these coupling effects:

- Simulator coupling for systems of PDAEs based ony dynamic iteration and
- Model order reduction.

1.4.2 Dynamic Iteration

In the first approach, all *dynamic* effects (for circuits, devices, thermal effects etc.) are modeled and simulated separately using their own simulation package which is based on their own time stepping algorithm in the numerical kernel. In this approach, modular, i.e., distributed time integration methods are quite natural which exploit different time constants of the single models by using different time step sizes (multirate approach).

Assuming the packages are equipped with appropriate interfaces, the coupling of the PDAE model via right-hand sides, source terms or boundary conditions can be done by coupling the simulators at communication time points. As the PDAE systems are coupled dynamically, an outer iteration process (dynamic iteration) has to be performed until getting convergence within a macro time step from one communication time point to the next one. Equipped with adequate relaxation and overlapping techniques, dynamic iteration schemes have to be derived which can guarantee a stable error propagation from one macro time step to the next one, thus ensuring rapid convergence as well as robustness and stability of the overall scheme used for coupling the models and simulators, respectively. This *distributed time integration* approach can quite naturally exploit the multirate, i.e., multiscale behavior in the time domain, as the different time stepping algorithms can use different time step sizes in accordance with the different time constants of the single models.

1.4.3 Model Order Reduction

If all the domains are coupled together for optimisation, then the resulting systems will become very large. Moreover, they have to be solved very often, in particular if multi-objective optimisation methods are employed and/or yield improvement is one of the optimisation targets. In this setting the usage of reduced order models is appealing, since it helps to save simulation time and memory needs, and supports to focus on those features of the various domains which are the most relevant ones for achieving the design objectives. Another benefit of using reduced order models might be in some cases to enable global optimisation of a design, while hiding technological or circuit design details which are related to intellectual property issues.

One way to obtain reduced order models is to develop structural macromodels or behavioral descriptions, or to employ network reduction techiques. Alternatively, for a given set of equations – which are possibly obtained by (semi)discretisation of the original problem – numerical MOR techniques may be used to get a system of the same structure but reduced dimension. The latter approach, quite well established in the electronics design community, was to be pursued in the COMSON project. Clearly, to be useful in the framework of design optimisation, the MOR has to generate *parameterised reduced order models*, and should be insensitive against small changes of the technological parameters. Other needed features are *maintaining the DAE/PDAE structure of the models*, and tuning for usage of reduced order models in simulation of *large nonlinear systems*.

1.4.4 Optimisation

Aiming at a realistic, medium size coupled problem of industrial relevance, one faces a multiple domain space with a large number of design objectives and restrictions (about 10–100), and works in a very complex parameter space (several hundreds to thousands of parameters). As far as manufacturability requirements are concerned, optimisation deals with discrete as well as continuous variables. In addition, any evaluation of a model (functions, constraints) is very costly (each requiring a coupled simulation), and possibly noisy. So usage of sensitivity analysis techniques is advisable, but how they can be based on noisy simulation results will require special attention.

Last, the reliability and robustness of a simulator depends on the accuracy of the implemented models and, in particular, the model parameters. In fact each separate model already has several hundreds of parameters. Therefore, in order to calibrate the models, new advanced and efficient parameter extraction techniques are needed.

The hot spot benchmark example, a Power-MOS circuit introduced by STMicroelectronics as an example for electro-thermal coupling, will show how all these different levels are linked: in Sect. 2.2.2, the PDAE model describing the hot spot benchmark example is carefully discussed. Simulation results for the coupled system based on the *Demonstrator Platform* methodology can be found in Sect. 8.3. Finally, Chap. 7 discusses how to embedd an optimization flow in an industrial environment to optimize the benchmark circuit with respect to the peak current.

References

1. Adler, M.S.: Accurate calculations of the forward drop and power dissipation in thyristors. IEEE Trans. Electron Devices **25**(1), 16–22 (1978)
2. Bai, Z., Dewilde, P., Freund, R.: Reduced order modeling. In: Schilders, W., ter Maten, E.J.W. (eds.) Handbook of Numerical Analysis, vol. XIII, pp. 825–895. NorthHolland/Elsevier, Amsterdam (2005)
3. Bartel, A.: Partial Differential-Algebraic Models in Chip Design-Thermal and Semiconductor Problems. Fortschritt-Berichte VDI, vol. 391, Reihe 20. VDI-Verlag, Düsseldorf (2004)
4. Brandtner, Th.: Chip-package codesign flow for mixed-signal SiP designs. IEEE Design Test Comput. **23**(3), 196–202 (2006)
5. Engl, W.L., Laur, R., Dirks, H.K.: Medusa – a simulator for modular circuits. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **1**(2), 85–93 (2006)
6. Freeda: An Open-Source Multiphysics Simulator [Online]. Available: http://www.freeda.org/
7. Fukahori, K.: Computer simulation of monolithic circuit performance in the presence of electro-thermal interactions. Ph.D. thesis, University of California, Berkeley (1977)
8. Grotelüschen, E., Grabinksi, H., Rochel, S., Winkel, T.-M.: LOSSYWIRE – a model implementation for transient and AC analysis of lossy coupled transmission lines in the circuit simulator ELCO. AEÜ **49**, 37–43 (1995)
9. Günther, M.: Partielle Differential-Algebraische Systeme in der Numerischen Zeitbereichsanalyse Elektrischer Schaltungen. VDI-Verlag, Düsseldorf (2001)
10. Mayaram, K.: CODECS: a Mixed-level Circuit and Device Simulator. University of California, Berkeley (1988)
11. Orhanovic, N., Tripathi, V.K.: Nonlinear transient analysis of coupled RLGC lines by the method of characteristics. Int. J. Microw. Millim. Wave Comput. Aided Eng. **2**(2), 108–115 (2007)
12. Schutt-Aine, J.E.: Static analysis of V transmission lines. IEEE Trans. Microw. Theory Tech. **40**(4), 2151–2156 (1992)
13. Roychowdhury, J.S., Newton, A.R., Pederson, D.O.: Algorithms for the transient simulation of lossy interconnect. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **12**, 96–104 (1994)
14. Ruehli, A.E.: Equivalent circuit models for three-dimensional multiconductor systems. IEEE Trans. Microw. Theory Tech. **22**(3), 216–224 (1974)
15. Selva Soto, M., Tischendorf, C.: Numerical analysis of DAEs from coupled circuit and semiconductor simulation. Appl. Numer. Math. **53**(2–4), 471–488 (2005)
16. Sheehan, B.N.: Realizable reduction of RC networks. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **26**(8), 1393–1407 (2007)
17. Tischendorf, C.: Coupled Systems of Differential Algebraic and Partial Differential Equations in Circuit and Device Simulation. Humboldt Universität zu Berlin, Habilitation (2004)
18. Wachutka, G.K.: Rigorous thermodynamic treatment of heat generation and conduction in semiconductor device modeling. IEEE Trans. Comput. Aided Design Integr. Circuits Syst. **9**(11), 1141–1149 (1990)
19. Wünusche, S., Clauß, C., Schwarz, P., Winkler, F.: Electro-thermal circuit simulation using simulator coupling. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **5**(3), 277–282 (1997)

Part II
Partial Differential Algebraic Equations

Partial Differential-Algebraic Equations, for short PDAEs, couple differential-algebraic network models for lumped descriptions and partial differential equations for the spatially distributed elements and effects via right-hand sides, source terms or boundary conditions. This approach, discussed in Chap. 2, requires new analysis with respect to consistency and validity of the overall PDAE model that links different domains and levels of physical description, existence of solutions, and robustness and efficiency of the numerical methods being applied for solving the extended sets of equations.

Simulator coupling is commonly used in an industrial framework for simulating these PDAE systems numerically. In this approach discussed in Chap. 3 from a more mathematical viewpoint, all subsystems of the PDAE, which are simulated separately using their own simulation package, are dynamically coupled at communication time points. A non-trivial dynamic iteration process has to be performed until convergence is achieved when stepping from one communication time point to another. Speed of convergence, stability (error propagation) and robustness of the overall scheme are the main points for research here. New, advanced, relaxation techniques have to be developed together with *multirate* techniques, which use different time step sizes according to the different time constants of the single subsystems.

Chapter 2
PDAE Modeling and Discretization

Giuseppe Alì, Massimiliano Culpo, Roland Pulch, Vittorio Romano, and Sebastian Schöps

Abstract We consider mathematical modeling in nanoelectronics, which causes coupled systems of differential algebraic equations and partial differential equations. Both modeling and discretization are investigated for the inclusion of advanced semiconductor behavior, heat conduction and electromagnetic effects within electric networks.

2.1 Introduction on Modeling and PDAEs

In this chapter, we introduce the mathematical modeling for the simulation of circuits and devices in nanoelectronics. To include the significant effects, a refined modeling using partial differential algebraic equations (PDAEs) is necessary.

G. Alì (✉)
Department of Mathematics, University of Calabria, 87036 Arcavacata di Rende (CS), Italy
e-mail: giuseppe.ali@unical.it,

M. Culpo
Università di Bologna, Bologna, Italy
e-mail: m.culpo@cineca.it

R. Pulch
Institut für Mathematik und Informatik, Ernst Moritz Arndt Universität Greifswald,
Walther-Rathenau-Straße 47, D-17487 Greifswald, Germany
e-mail: pulchr@uni-greifswald.de

V. Romano
Dipartimento di Matematica e informatica, Università di Catania, Viale A. Doria no, 95125
Catania, Italy
e-mail: romano@dmi.unict.it

S. Schöps
Graduate School of Excellence Computational Engineering, TU Darmstadt, Dolivostraße 15,
64293 Darmstadt, Germany
e-mail: schoeps@gsc.tu-darmstadt.de

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_2

2.1.1 Mathematical Modeling in Nanoelectronics

The mathematical modeling of electronic circuits is typically based on some network approach. Thereby, we analyse the transient behavior of node voltages and branch currents. The basic elements of the circuit exhibit corresponding relations between voltages and currents, which represent differential equations or algebraic equations. The topology of the circuit is considered via Kirchhoff's current law and Kirchhoff's voltage law, which are algebraic equations. It follows a system of differential algebraic equations (DAEs).

For example, mathematical modeling using the modified nodal analysis (MNA), see [26], yields systems of the form

$$\begin{aligned} A_C \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} + A_R \mathbf{r}(A_R^T \mathbf{e}) + A_L \mathbf{i}_L + A_V \mathbf{i}_V + A_I \mathbf{i}_I &= 0, \\ \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} - A_L^T \mathbf{e} &= 0, \\ A_V^T \mathbf{e} - \mathbf{v}_V &= 0, \\ \mathbf{q} - \mathbf{q}_C(A_C^T \mathbf{e}) &= 0, \\ \boldsymbol{\phi} - \boldsymbol{\phi}_L(\mathbf{i}_L) &= 0, \end{aligned} \tag{2.1}$$

where $\mathbf{e}, \mathbf{i}_L, \mathbf{i}_V$ are the unknown node voltages and branch currents through inductors and voltage sources. The unknowns $\mathbf{q}, \boldsymbol{\phi}$ represent charges and fluxes, respectively. The functions $\mathbf{r}, \mathbf{q}_C, \boldsymbol{\phi}_L$ are predetermined. Independent current sources $\mathbf{i}_I$ and voltage sources $\mathbf{v}_V$ may appear. The incidence matrices A_C, A_L, A_R, A_V, A_I follow from the topology of the electronic circuit.

For a transient analysis of the system (2.1), consistent initial values have to be specified. The differential index of the DAE system (2.1) follows from the topology only. An appropriate mathematical modeling implies an index of one or two. Hence we can use common numerical methods for initial value problems of DAEs.

This modeling approach applies with the assumption of ideally joint lumped elements in the electronic circuit. No spatial coordinates appear, since the information on the topology is given by the incidences of the elements. For quite a long time, the mathematical modeling via time-dependent systems of DAEs has been sufficiently accurate to reproduce the transient behavior of the underlying physical circuit, i.e., the modeling error was sufficiently small. However, miniaturization causes parasitic effects in nanoelectronics, which cannot be neglected any more. Corresponding phenomena are, for example:

- *Quantum effects*: The down-scaling of transistors decreases also the size of the channel. The channel length comes close to the atomic scale. Hence quantum effects appear and have to be considered in the mathematical models.
- *Heating*: The faster clock rate in chips causes a higher power loss in the electronic network. The down-scaling implies that more heat is produced within

a unit area. Since cooling cannot ensure a homogeneous temperature any more, the heat distribution and the heat conduction has to be considered. In particular, thermal effects of transistors appear due to the semiconductor's dependence on temperature.
- *Electromagnetic effects*: The distance between transmission lines on a chip becomes tiny due to the miniaturization. The current through some transmission line can induce a significant current in a neighboring component. Thus the interference of transmission lines has to be taken into account.

These parasitic phenomena represent spatial effects. Thus corresponding mathematical models apply partial differential equations (PDEs) in time as well as space. Firstly, PDE models are required, which reproduce phenomena like quantum and thermal effects with a high accuracy. Secondly, the parasitic effects are considered in the electronic network, i.e., the PDEs are coupled to the circuit's DAEs. It follows a system of partial differential algebraic equations (PDAEs).

On the one hand, the basic network approaches for modeling electronic circuits yield time-dependent systems of DAEs, which can be written in the general form

$$\mathbf{F} : \mathbb{R}^k \times \mathbb{R}^k \times I \to \mathbb{R}^k, \quad \mathbf{F}\left(\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}, \mathbf{y}, t\right) = 0, \tag{2.2}$$

where $\mathbf{y} : I \to \mathbb{R}^k$ denotes the unknown solution in a time interval $I := [t_0, t_1]$. The MNA equations (2.1) represent an often used model of the type (2.2). A consistent initial value $\mathbf{y}(t_0) = \mathbf{y}_0$ has to be given. On the other hand, a parasitic phenomenon is included via PDEs. We arrange the general formulation

$$\mathscr{L} : D \times I \times V \to \mathbb{R}^m, \quad \mathscr{L}(\mathbf{x}, t, \mathbf{u}) = 0 \tag{2.3}$$

with a differential operator $\mathscr{L}$. Thereby, $D \subset \mathbb{R}^d$ for $d \in \{1, 2, 3\}$ represents the underlying spatial domain. The solution $\mathbf{u} : D \times I \to \mathbb{R}^m$ belongs to some function space V. Initial and boundary conditions have to be specified appropriately.

Coupling the DAEs (2.2) and the PDEs (2.3) yields systems of PDAEs in time as well as space. The coupling is feasible via

- (Artificial) coupling variables,
- Source terms,
- Boundary conditions (BCs).

More sophisticated couplings also appear. The involved PDEs may be of mixed type (elliptic, hyperbolic, parabolic). For example, the drift-diffusion equations for semiconductors, the telegrapher's equation for transmission lines or the heat equation for resistors are used in practice. The types of PDAEs, which result from the modeling in nanoelectronics, are discussed in the following subsection.

2.1.2 Classification of PDAE Models

As introduced above, we consider mathematical models of PDAEs, i.e., coupled systems of DAEs (2.2) and PDEs (2.3). The notion PDAE is also applied in the context of singular PDEs. For example, we discuss the linear PDE

$$A\frac{\partial \mathbf{u}}{\partial t} + B\frac{\partial \mathbf{u}}{\partial x} = \mathbf{s}(x, t, \mathbf{u}) \tag{2.4}$$

with matrices $A, B \in \mathbb{R}^{k\times k}$. If A and/or B are singular, then a singular PDE appears. PDAEs in the sense of singular PDEs are investigated in [44], for example. For electronic circuits with amplitude modulated signals or frequency modulated signals, the introduction of different time variables transforms the circuit's DAEs (2.1) into singular PDEs, see [51].

If the matrix B is regular and the matrix A singular and $B^{-1}A$ diagonalizable, then the system of PDEs (2.4) can be transformed into the equivalent system

$$\begin{aligned}\frac{\partial \tilde{\mathbf{u}}_1}{\partial t} + \tilde{B}_1\frac{\partial \tilde{\mathbf{u}}_1}{\partial x} &= \tilde{\mathbf{s}}_1(x, t, \tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2),\\ \frac{\mathrm{d}\tilde{\mathbf{u}}_2}{\mathrm{d}x} &= \tilde{\mathbf{s}}_2(x, t, \tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2).\end{aligned}$$

The result can be seen as a coupled systems of PDEs and ODEs, i.e., a PODE. The source term causes the coupling within the right-hand sides. Likewise, a coupled system of PDEs and DAEs appears for other cases of the matrices A, B. Thus some singular PDEs correspond to systems of PDAEs.

In the following, we consider PDAEs in the sense of coupled systems of DAEs and PDEs only. We present a rough classification of PDAE models in nanoelectronics according to [12]. Two approaches for PDAE modeling exist: refined modeling and multiphysical extensions.

2.1.2.1 Refined Modeling

Complex elements of the circuit with a spatial distribution like semiconductors and transmission lines can be modeled via substitute circuits consisting of lumped basic elements. These companion models include artificial parameters, which have to be chosen appropriately to approximate the behavior of the element. Alternatively, PDE models exist, which describe these elements directly. We consider one or several components of the electronic circuit by its PDE model and couple the PDE to the system of DAEs modeling the surrounding network.

The resulting PDAE system is more difficult to analyze and more costly to solve numerically than a DAE system based on companion models. Nevertheless, the refined modeling allows to describe certain elements of the circuits with a higher accuracy, i.e., the modeling error becomes relatively low. Hence we can focus on critical components of an electronic circuit. Moreover, the refined modeling yields results, which can be used for the construction and the validation of better companion models. Sophisticated PDE models for semiconductor behavior have been developed for this purpose, see [7–9, 48, 49, 53–58], for example. The aim is to reproduce the electric input-output behavior of the semiconductor with a high accuracy in the presence of quantum and thermal effects.

The coupling of the DAE network and the PDE systems is performed via voltages and currents. The node potentials of the connecting network yield boundary conditions of Dirichlet type for the Ohmic contacts of the PDE model. At other boundaries without electric contacts, homogeneous boundary conditions of von-Neumann type may appear. Vice versa, the output of the PDE model represents an electric current, which enters the surrounding network. It follows a source term for the DAE system. The refined modeling yields PDAE systems of the form

$$\begin{aligned} A\tfrac{\partial}{\partial t}\mathbf{u} + \mathscr{L}_D\mathbf{u} - \mathbf{s}(\mathbf{u},t) &= \mathbf{p}(\mathbf{y}) && \text{(PDE in } I \times D\text{)} \\ \mathbf{u}|_{\Gamma_1} &= \mathbf{g}(\mathbf{y}) && \text{(Dirichlet BC)} \\ \tfrac{\partial}{\partial \mathbf{n}}\mathbf{u}\big|_{\Gamma_2} &= \mathbf{h}(\mathbf{y}) && \text{(Neumann BC)} \\ \mathbf{F}\left(\tfrac{\mathrm{d}}{\mathrm{d}t}\mathbf{y},\mathbf{y},t\right) &= \mathbf{r}(\mathbf{u}) && \text{(DAE in } I\text{)} \end{aligned} \tag{2.5}$$

with a matrix A and a spatial differential operator $\mathscr{L}_D$ with domain D. The coupling can be realized via the source terms $\mathbf{p}, \mathbf{r}$ or the boundary conditions $\mathbf{g}, \mathbf{h}$, where the boundary is decomposed into $\partial D = \Gamma_1 \cup \Gamma_2$.

We categorize the refined modeling into the following cases:

- *Semiconductors*: Several transistors or diodes of the electronic circuit are modeled via drift-diffusion equations or quantum mechanical equations, which are coupled to the electric network. Existence and uniqueness of solutions for models including stationary or non-stationary drift-diffusion equations is analyzed in [4, 5]. The drift-diffusion equations represent PDEs of mixed type. Hydrodynamical models for semiconductors, which represent hyperbolic PDEs, are considered in [6].
- *Transmission lines*: Telegrapher's equation describes the physical effects in transmission lines, i.e., a PDE model of hyperbolic type. The coupling of these PDEs and the network's DAEs exhibits the form (2.5). For further details, see [36, 37].

2.1.2.2 Multiphysical Extensions

Refined modeling can be seen as a partitioning of the electronic circuit, where we describe some parts by PDEs and model the remaining larger part via the traditional DAE formulation. Moreover, the involved systems of PDEs always describe the electric or electromagnetic behavior of some components of the circuits. In contrast, multiphysical modeling introduces an additional distributed effect within the complete circuit. We consider the circuit as two or more layers, where one layer corresponds to the common network description and the other layers model another physical effect given by PDEs.

Multiphysical modeling includes the following phenomena, for example:

- *Thermal aspects*: The faster clock rate implies a significant heat production in particular parts of the electronic circuit. Thus cooling cannot achieve a homogeneous and moderate temperature. Since the electric behavior of the components depends on the temperature (for example, strongly for resistors), the heat distribution and conduction has to be considered in the numerical simulation.

  In addition to the electric network, a thermal network can be arranged, which describes the heat flow within the circuit, see [29]. The thermal network consists of zero-dimensional elements as in the electric network. Moreover, a refined modeling of the thermal network is feasible, where some elements are replaced by a PDE model based on the heat equation in one, two or three space dimensions. The heat equation, i.e., Fourier's law, represents a parabolic PDE. Further details can be found in [11]. Modeling, analysis and discretization corresponding to two dimensional heat equations is considered in [3, 20, 21, 25].

  A special case is given by the usage of the heat equation with a spatial domain including the complete electronic circuit. Consequently, we obtain two layers in parallel: the electric network described by DAEs and the thermal aspects modeled via a PDE.
- *Electromagnetics*: On the one hand, Maxwell's equations imply the network approaches, which produce the DAE formulations (2.2), via according simplifications. The aim is to achieve an efficient numerical simulation. On the other hand, the electronic circuit can be described completely by the full Maxwell's equations, i.e., a PDE system. However, this approach would cause a huge computational effort.

  Alternatively, just some parts or components of the circuit can be modeled by Maxwell's equations or its variants like the magnetoquasistatic formulation. The systems of PDEs are coupled to the network's DAEs again. Hence the same effects are described in different ways, i.e., distinct mathematical models. This approach is similar to a refined modeling. Nevertheless, the model represents a multiphysical extension, since the magnetic fluxes are considered in addition to the purely electric behavior of the circuit. An application based on magnetoquasistatic equations is presented in [59].

We note that refined modeling and multiphysical extensions can also be combined. In a multiphysical framework, we can arrange a refined modeling of some components (semiconductors, transmission lines) within the layer of the common electric network. However, such a complex structure is not considered in the following, i.e., we apply either refined modeling or multiphysical extensions.

In this chapter, we present some examples of mathematical models, which yield systems of PDAEs. The chapter is organized as follows. In Sect. 2.2.1, a refined modeling for semiconductor devices is performed, where diodes are described by systems of PDEs including two space dimensions. The resulting system of PDAEs is discussed. In Sect. 2.2.2, a multiphysical modeling is performed by considering thermal behavior at the system level. The electric network is coupled to the heat equation. In Sect. 2.2.3, multiphysical modeling of the electric circuits is considered based on Maxwell's equations. The approach applies a magnetoquasistatic formulation. In Sect. 2.2.4, a description of thermal and quantum effects for semiconductor devices is presented to obtain according mathematical models. Thereby, the focus is on the PDE level, which can be used as a module in further refined models.

2.2 Modeling, Analysis and Discretization of Coupled Problems

We present four applications of coupled problems in nanoelectronics to illustrate the essential strategies.

2.2.1 Refined Modeling of Networks with Devices

We investigate electric networks including semiconductor devices. Some devices are described by more sophisticated mathematical models based on partial differential equations now, whereas the surrounding electric network is still represented by traditional models using differential algebraic equations.

2.2.1.1 Modeling of Electric Networks

An RCL electric network is a directed graph with n_v vertices (or nodes), and n_a arcs (or branches) which contain resistors, capacitors and inductors, and independent voltage and current sources, $\mathbf{v}_V(t) \in \mathbb{R}^{n_V}$ and $\mathbf{i}_I(t) \in \mathbb{R}^{n_I}$. The branches are usually labelled according to the components they contain: R for resistors, C for capacitors, L for inductors, V for voltage sources, I for current sources.

The topology of the network can be described by an incidence matrix $A = (a_{ij}) \in \mathbb{R}^{n_v \times n_a}$, defined by:

$$a_{ij} = \begin{cases} -1 \text{ if the branch } j \text{ leaves the node } i, \\ \;\;\,1 \text{ if the branch } j \text{ enters the node } i, \\ \;\;\,0 \text{ otherwise.} \end{cases} \tag{2.6}$$

To keep track of different branches, they are collected according to their labels (R, C, L, I, V), and write

$$A = (A_R, A_C, A_L, A_I, A_V) \in \mathbb{R}^{n_v \times (n_R + n_C + n_L + n_I + n_V)} \equiv \mathbb{R}^{n_v \times n_a}.$$

The electric behavior of the network is described by a set of time-dependent variables associated to its nodes and branches. An applied potential is associated to each node ($\mathbf{u} \in \mathbb{R}^{n_v}$), a voltage drop and a current is associated to each branch ($\mathbf{v}, \mathbf{i} \in \mathbb{R}^{n_a}$). To keep track of the different labels, we write

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_R \\ \mathbf{v}_C \\ \mathbf{v}_L \\ \mathbf{v}_I \\ \mathbf{v}_V \end{pmatrix}, \quad \mathbf{i} = \begin{pmatrix} \mathbf{i}_R \\ \mathbf{i}_C \\ \mathbf{i}_L \\ \mathbf{i}_I \\ \mathbf{i}_V \end{pmatrix}.$$

The direction of each branch coincides with the positive direction of the voltage drop and the current through the branch. The voltage drops and the applied potentials are related by the voltage relation:

$$\mathbf{v} = A^\top \mathbf{u}. \tag{2.7}$$

The currents satisfy Kirchhoff's current law:

$$A\mathbf{i} = 0, \tag{2.8}$$

which ensures charge conservation. To the above relations we need to add constitutive relations for the RCL components:

$$\mathbf{i}_R = \mathbf{r}(\mathbf{v}_R), \quad \mathbf{i}_C = \frac{d\mathbf{q}}{dt}, \quad \mathbf{v}_L = \frac{d\boldsymbol{\phi}}{dt}, \tag{2.9}$$

with

$$\mathbf{q} = \mathbf{q}_C(\mathbf{v}_C), \quad \boldsymbol{\phi} = \boldsymbol{\phi}_L(\mathbf{i}_L). \tag{2.10}$$

Here, $\mathbf{q}_C$ collects the charges inside the capacitors, and $\boldsymbol{\phi}_L$ is a flux term for the inductors. Finally, for the branches with sources we assume to know the time-dependent functions $\mathbf{i}_I(t)$, $\mathbf{v}_V(t)$.

Following the formalism of Modified Nodal Analysis (MNA) [40, 50], we use the relations (2.9) in Kirchhoff's current law (2.8), together with the voltage relation (2.7) and the relations (2.10), to obtain the DAE equation (2.1), for the unknowns $\mathbf{q}$, $\boldsymbol{\phi}$, $\mathbf{u}$, $\mathbf{i}_L$, $\mathbf{i}_V$. Sometimes it is convenient to reduce the number of variables, eliminating $\mathbf{q}$ and $\boldsymbol{\phi}$. This leads to the following alternative form of the MNA equations, for the unknowns $\mathbf{u}$, $\mathbf{i}_L$, $\mathbf{i}_V$:

$$\begin{aligned} A_C \frac{\mathrm{d}\mathbf{q}_C(A_C^T\mathbf{u})}{\mathrm{d}t} + A_R\mathbf{r}(A_R^T\mathbf{u}) + A_L\mathbf{i}_L + A_V\mathbf{i}_V + A_I\mathbf{i}_I &= 0, \\ \frac{\mathrm{d}\boldsymbol{\phi}_L(\mathbf{i}_L)}{\mathrm{d}t} - A_L^T\mathbf{u} &= 0, \\ A_V^T\mathbf{u} - \mathbf{v}_V &= 0. \end{aligned} \tag{2.11}$$

The above equations apply also to electric circuits with semiconductor devices, provided that the devices are described by concentrated (companion) models, i.e., by means of equivalent RCL circuits. In this framework, a semiconductor device is represented by a subnetwork of the overall electric network. In following subsection we will show how to replace these subnetworks with distributed models for semiconductor devices.

2.2.1.2 Distributed Models for Devices

In this subsection, we consider an electric network with n_D semiconductor devices. We assume that the i-th device has $1 + K_i$ contacts. More precisely, we model the i-th device by a d-dimensional domain Ω^i, $i = 1, \ldots, n_D$, with $d = 1, 2$, or 3, and we assume that the boundary $\partial\Omega^i$ is made of a Dirichlet part Γ_D^i, union of $1 + K_i$ disjoint parts, which represent Ohmic contacts, and of a Neumann part Γ_N^i, which represents insulating boundaries (for $d > 1$),

$$\Gamma_D^i = \bigcup_{j=0}^{K_i} \Gamma_{D,j}^i, \quad \Gamma_N^i = \partial\Omega^i \setminus \Gamma_D^i, \quad i = 1, \ldots, n_D.$$

In total, the devices contain n_{vD} Ohmic contacts, with

$$n_{vD} := n_D + \sum_{j=1}^{n_D} K_j.$$

Each contact must be connected to a node of the electric network. To relate the contacts of the devices to the nodes of the network, we need to introduce a contact-to-node selection matrix, $\mathbf{S}_D = (s_{D,ij}) \in \mathbb{R}^{n_v \times n_{vD}}$, defined by:

$$s_{D,ij} = \begin{cases} 1, \text{ if the contact } j \text{ is connected to the node } i, \\ 0, \text{ otherwise.} \end{cases} \tag{2.12}$$

This definition differs with the definition of branch-to-node incidence matrix, previously given. In fact, the branch-to-node incidence matrix relates each branch to two nodes, and the values 1 and -1 give information on the orientation of the branch, while the contact-to-node selection matrix relate each contact to one node.

The behavior of the i-th device is described by an electric potential $\phi^i(\mathbf{x}, t)$, and by a vector variable $\mathbf{U}^i(\mathbf{x}, t)$, which collects the other macroscopic variables for the device, such as carrier density, flux density, energy, etc. Several models can be used, with different mathematical characters, but sharing some common features.

1. The first common feature is that the electric potential ϕ^i is generated by the built-in charge, $\rho^i_{\text{bi}}(\mathbf{x})$, due to the dopants embedded in the semiconductor, and by the charge density $\rho^i(\mathbf{U}^i)$, due to the carriers, so that it satisfies the Poisson equation:

$$-\nabla \cdot (\epsilon^i \nabla \phi^i) = \rho^i_{\text{bi}} + \rho^i(\mathbf{U}^i), \tag{2.13}$$

where $\epsilon^i(\mathbf{x})$ is the dielectric constant. This equation is supplemented with the following boundary conditions:

$$\begin{cases} \phi^i = \phi^i_{\text{bi}}(\rho^i_{\text{bi}}) + u^i_{D,j}, & \text{on } \Gamma^i_{D,j},\ j = 0, 1, \ldots, K_i, \\ \boldsymbol{\nu}^i \cdot \nabla \phi^i = 0, & \text{on } \Gamma^i_N, \end{cases} \tag{2.14}$$

where $\phi^i_{\text{bi}}(\rho^i_{\text{bi}})$ is the built-in potential, $u^i_{D,j}$, $j = 0, 1, \ldots, K_i$, are the applied potentials at the Ohmic contacts of the i-th device, and the symbol $\boldsymbol{\nu}^i$ denotes the external unit normal to $\partial \Omega^i$. For later use, we comprise the applied voltages in the vectors:

$$\mathbf{u}^i_D = \begin{pmatrix} u^i_{D,0} \\ \vdots \\ u^i_{D,K_i} \end{pmatrix} \in \mathbb{R}^{1+K_i}, \quad \mathbf{u}_D = \begin{pmatrix} \mathbf{u}^1_D \\ \vdots \\ \mathbf{u}^{n_D}_D \end{pmatrix} \in \mathbb{R}^{n_{\upsilon D}}.$$

2. The second common feature, is that the device variable $\mathbf{U}^i$ satisfies a system of partial differential equations, which is coupled to the electric potential only through the electric field $\mathbf{E}^i = -\nabla \phi^i$. Symbolically, we can write

$$\mathscr{F}^i(\mathbf{U}^i, \tfrac{\partial}{\partial t}\mathbf{U}^i, \nabla \mathbf{U}^i, \ldots; \mathbf{E}^i) = 0. \tag{2.15}$$

In the following sections we will see explicitly several of these partial differential models.

3. The last common feature is that (2.15) is consistent with the conservation of the charge density:

$$\frac{\partial \rho^i(\mathbf{U}^i)}{\partial t} + \nabla \cdot \mathbf{J}^i(\mathbf{U}^i) = 0. \tag{2.16}$$

Here, $\mathbf{J}^i(\mathbf{U}^i)$ is the electric current, which can be a component of the variable $\mathbf{U}^i$, or can be evaluated as a functional of the said variable. The electric current $\mathbf{J}^i$

depends also on the applied potentials $u^i_{D,j}$, $j = 0, 1, \dots, K_i$, due to the coupling of (2.15) with the Poisson's equation (2.13), through the electric field $\mathbf{E}^i$.

As a consequence of (2.13) and (2.16), we have

$$\nabla \cdot \left(\epsilon^i \frac{\partial}{\partial t} \mathbf{E}^i + \mathbf{J}^i(\mathbf{U}^i) \right) = 0 \tag{2.17}$$

The term $\epsilon \frac{\partial}{\partial t} \mathbf{E}^i$ is the displacement current, and represents the current induced by time-variations of the electric field. Then, the total current in the i-th device is given by

$$\mathbf{j}^i := \epsilon^i \frac{\partial}{\partial t} \mathbf{E}^i + \mathbf{J}^i(\mathbf{U}^i). \tag{2.18}$$

The current $j^i_{D,j}$ through the j-th contact of the i-th device, is defined by:

$$j^i_{D,j} = - \int_{\Gamma^i_{D,j}} \mathbf{j}^i \cdot \boldsymbol{\nu}^i \, d\sigma(\mathbf{x}). \tag{2.19}$$

We introduce the vectors

$$\mathbf{j}^i_D = \begin{pmatrix} j^i_{D,0} \\ \vdots \\ j^i_{D,K_i} \end{pmatrix} \in \mathbb{R}^{1+K_i}, \quad \mathbf{j}_D = \begin{pmatrix} \mathbf{j}^1_D \\ \vdots \\ \mathbf{j}^{n_D}_D \end{pmatrix} \in \mathbb{R}^{n_{\nu D}}.$$

Recalling the definition of the selection matrix, the MNA equations need to be modified in the following way:

$$\begin{aligned} A_C \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} + A_R \mathbf{r}(A_R^T \mathbf{u}) + A_L \mathbf{i}_L + A_V \mathbf{i}_V + A_I \mathbf{i}_I + \boldsymbol{\lambda} &= 0, \\ \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} - A_L^T \mathbf{u} &= 0, \\ A_V^T \mathbf{u} - \mathbf{v}_V &= 0. \\ \mathbf{q} - \mathbf{q}_C(A_C^T \mathbf{u}) &= 0, \\ \boldsymbol{\phi} - \boldsymbol{\phi}_L(\mathbf{i}_L) &= 0, \end{aligned} \tag{2.20}$$

where the auxiliary variable $\boldsymbol{\lambda} \in \mathbb{R}^{n_v}$ is given by the device-to-network coupling relation:

$$\boldsymbol{\lambda} = \mathbf{S}_D \mathbf{j}_D. \tag{2.21}$$

To close the system, we also need the network-to-device coupling relation:

$$\mathbf{u}_D = \mathbf{S}_D^\top \mathbf{u}. \tag{2.22}$$

Remark 2.1 The components of the vector $\mathbf{j}_D$ are not independent. In fact, by using (2.17), after integrating by parts over Ω^i, we find

$$\sum_{j=0}^{K_i} j_{D,j}^i = 0, \quad i = 1, \dots, n_D. \tag{2.23}$$

This means that we can express $\mathbf{j}_D^i$ and, consequently, $\mathbf{j}_D$ in terms of the vectors

$$\mathbf{i}_D^i = \begin{pmatrix} j_{D,1}^i \\ \vdots \\ j_{D,K_i}^i \end{pmatrix} \in \mathbb{R}^{K_i}, \quad \mathbf{i}_D = \begin{pmatrix} \mathbf{i}_D^1 \\ \vdots \\ \mathbf{i}_D^{n_D} \end{pmatrix} \in \mathbb{R}^{n_{aD}}, \tag{2.24}$$

with $n_{aD} = \sum_{i=1}^{n_D} K_i$, by means of the relations:

$$\mathbf{j}_D^i = \mathbf{A}_D^{*i} \, \mathbf{i}_D^i, \qquad \mathbf{j}_D = \mathbf{A}_D^* \, \mathbf{i}_D, \tag{2.25}$$

where

$$\mathbf{A}_D^{*i} = \begin{pmatrix} -1 & \cdots & -1 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{(1+K_i)\times K_i}, \tag{2.26}$$

$$\mathbf{A}_D^* = \operatorname{diag}(\mathbf{A}_D^{*1}, \dots, \mathbf{A}_D^{*n_D}) \in \mathbb{R}^{n_{vD}\times n_{aD}}. \tag{2.27}$$

Remark 2.2 The components of the vector $\mathbf{j}_D^i$ depend only on the voltage drops

$$\mathbf{v}_D^i = \mathbf{A}_D^{*i\top} \mathbf{u}_D^i, \quad i = 1, \dots, n_D. \tag{2.28}$$

Thus, the components of the overall vector $\mathbf{j}_D$ depend only on the voltage drops

$$\mathbf{v}_D = \mathbf{A}_D^{*\top} \mathbf{u}_D. \tag{2.29}$$

In fact, recalling (2.15), the variables $\mathbf{U}^i$ are coupled to the Poisson equation only through the electric field $\mathbf{E}^i$, and so are the components of the electric current $\mathbf{J}^i(\mathbf{U}^i)$, and the components of $\mathbf{j}^i$, which appear in (2.19). Since the electric field is not affected by a time-dependent translation of the electric potential,

$\phi_D^i \to \phi_D^i + u_{D,0}^i(t)$, we have a dependence of $\mathbf{j}^i$ on the voltage drops $v_{D,j}^i := u_{D,j}^i - u_{D,0}^i$, $j = 1, \ldots, K_i$, which in compact form can be written as in (2.28).

As a consequence of the previous remarks, the coupling conditions (2.21) and (2.22) can be replaced by the conditions

$$\boldsymbol{\lambda} = \mathbf{A}_D \mathbf{i}_D, \tag{2.30}$$

$$\mathbf{v}_D = \mathbf{A}_D^\top \mathbf{u}, \tag{2.31}$$

where we have introduced the device incidence matrix

$$\mathbf{A}_D := \mathbf{S}_D \mathbf{A}_D^*. \tag{2.32}$$

We call this matrix "incidence matrix" because, for devices with two Ohmic contacts, it reduces to the usual incidence matrix for branches with two-terminal devices.

2.2.1.3 Displacement Current and Device Capacitance Matrix

The displacement currents, present in the definition of $\mathbf{i}_D$, will cause an additional capacitance effect. To see this, we introduce the auxiliary functions φ_j^i, defined by:

$$\begin{cases} -\nabla \cdot (\epsilon^i \nabla \varphi_j^i) = 0, \text{ in } \Omega^i \\ \varphi_j^i = \delta_{jk}, & \text{on } \Gamma_{D,k}^i,\ k = 0, 1, \ldots, K_i, \\ \boldsymbol{\nu}^i \cdot \nabla \varphi_j^i = 0, & \text{on } \Gamma_N^i, \end{cases} \tag{2.33}$$

where δ_{jk} is the Kronecker delta. The auxiliary functions φ_j^i, $j = 0, 1, \ldots, K_i$, are not independent, since

$$\varphi_0^i = 1 - \sum_{j=1}^{K_i} \varphi_j^i. \tag{2.34}$$

Using these functions, we can find an alternative expression for the current $j_{D,j}^i$ through the j-th contact of the i-th device:

$$j_{D,j}^i \equiv -\int_{\partial\Omega^i} \varphi_j^i \mathbf{j}^i \cdot \boldsymbol{\nu}^i \, d\sigma = -\int_{\Omega^i} \nabla\varphi_j^i \cdot \mathbf{j}^i \, d\mathbf{x}, \tag{2.35}$$

where we have used the identity (2.17). Recalling the definition (2.18), the current $\mathbf{j}^i$ is the sum of the displacement current and the current due to the carriers. For the displacement current part, we find

$$
\begin{aligned}
-\int_{\Omega^i} \nabla\varphi_j^i \cdot \epsilon^i \frac{\partial}{\partial t}\mathbf{E}^i \, d\mathbf{x} &= \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega^i} \nabla\cdot(\epsilon^i \nabla\varphi_j^i \phi^i)\, d\mathbf{x} \\
&= \frac{\mathrm{d}}{\mathrm{d}t}\sum_{k=0}^{K_i}\int_{\Gamma^i_{D,k}} \boldsymbol{\nu}\cdot(\epsilon^i \nabla\varphi_j^i(\phi^i_{\mathrm{bi}} + u^i_{D,k})\varphi_k^i)\, d\sigma \\
&= \sum_{k=0}^{K_i}\int_{\Gamma^i_{D,k}} \boldsymbol{\nu}\cdot(\epsilon^i \nabla\varphi_j^i \varphi_k^i)\, d\sigma \frac{\mathrm{d}u^i_{D,k}}{\mathrm{d}t},
\end{aligned}
$$

which, using the divergence theorem and identity (2.34), leads to

$$
-\int_{\Omega^i} \nabla\varphi_j^i \cdot \epsilon^i \frac{\partial}{\partial t}\mathbf{E}^i \, d\mathbf{x} = \sum_{k=1}^{K_i}\int_{\Omega^i} \epsilon^i \nabla\varphi_j^i \cdot \nabla\varphi_k^i \, d\mathbf{x}\frac{\mathrm{d}v^i_{D,k}}{\mathrm{d}t}, \tag{2.36}
$$

with $v^i_{D,k} = u^i_{D,k} - u^i_{D,0}$. Combining this identity with (2.35), we find

$$
j^i_{D,j} = \sum_{k=1}^{K_i} C^i_{D,jk}\frac{\mathrm{d}v^i_{D,k}}{\mathrm{d}t} - \int_{\Omega^i} \nabla\varphi_j^i \cdot \mathbf{J}^i \, d\mathbf{x}, \tag{2.37}
$$

with

$$
C^i_{D,jk} = \int_{\Omega^i} \epsilon^i \nabla\varphi_j^i \cdot \nabla\varphi_k^i \, d\mathbf{x}. \tag{2.38}
$$

In concise form, we can write:

$$
\mathbf{i}_D = \mathbf{C}_D \frac{\mathrm{d}\mathbf{v}_D}{\mathrm{d}t} + \mathscr{I}_D(\mathbf{J}), \tag{2.39}
$$

with $\mathbf{C}_D = \mathrm{diag}(\mathbf{C}_D^1, \dots, \mathbf{C}_D^{n_D})$, $\mathbf{C}_D^i = (C^i_{D,jk}) \in \mathbb{R}^{K_i \times K_i}$, and

$$
\mathscr{I}_D(\mathbf{J}) = \begin{pmatrix} \mathscr{I}_D^1(\mathbf{J}^1) \\ \vdots \\ \mathscr{I}_D^{n_D}(\mathbf{J}^{n_D}) \end{pmatrix}, \quad \mathscr{I}_D^i(\mathbf{J}^i) = \begin{pmatrix} \mathscr{I}_1^i(\mathbf{J}^i) \\ \vdots \\ \mathscr{I}_{K_i}^i(\mathbf{J}^i) \end{pmatrix}, \quad \mathscr{I}_j^i(\mathbf{J}^i) = -\int_{\Omega^i} \nabla\varphi_j^i \cdot \mathbf{J}^i \, d\mathbf{x}
$$

for $j = 1, \dots, K_i$. Using the expression (2.39), the device-to-network coupling relation (2.30) becomes

$$
\boldsymbol{\lambda} = \mathbf{A}_D\, \mathbf{i}_D = \mathbf{A}_D \mathbf{C}_D \frac{\mathrm{d}\mathbf{v}_D}{\mathrm{d}t} + \mathbf{A}_D \mathscr{I}_D(\mathbf{J}). \tag{2.40}
$$

The matrix $\mathbf{C}_D$ is symmetric and positive definite, and can be interpreted as a capacitance matrix [2]. Thus we can write the previous relation as

$$\boldsymbol{\lambda} = \mathbf{A}_D \frac{\mathrm{d}\mathbf{q}_D}{\mathrm{d}t} + \mathbf{A}_D \mathscr{I}_D(\mathbf{J}), \tag{2.41}$$

$$\mathbf{q}_D = \mathbf{C}_D \mathbf{v}_D. \tag{2.42}$$

These relations represent an alternative formulation of the device-to-network coupling relation (2.30), to be used together with the network-to-device coupling relation (2.31).

2.2.1.4 The Drift-Diffusion Model

In what follows we exemplify the coupled equations for an electric network with semiconductor devices, by using a specific distributed model for the devices. For simplicity, we consider an RLC network which contains a single device ($n_D = 1$), with K terminals.

The basic distributed model for semiconductor devices is the drift-diffusion model. In this model, the electric behavior is described in terms of two charge carriers: electrons, with negative elementary charge $q_n = -q$, and holes, with positive elementary charge $q_p = q$. We denote by n, p, respectively, the electron and hole number density. The carrier number densities are coupled with the electric potential ϕ through Poisson's equation

$$-\nabla \cdot (\epsilon \nabla \phi) = \rho_{\mathrm{bi}} + \rho(n, p) \equiv q N_{\mathrm{bi}} - q n + q p, \tag{2.43}$$

with the doping profile N_{bi}, and satisfy the balance laws

$$\frac{\partial n}{\partial t} + \nabla \cdot \mathbf{j}_n = -R, \quad \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{j}_p = -R, \tag{2.44}$$

where $\mathbf{j}_n$, $\mathbf{j}_p$ are the electron and hole density flux, respectively, given by the following constitutive relations:

$$\mathbf{j}_n = -D_n \nabla n + \mu_n n \nabla \phi, \quad \mathbf{j}_p = -D_p \nabla p - \mu_p p \nabla \phi. \tag{2.45}$$

In the previous equations, $R = R(n, p)$ is the recombination-generation term, which is assumed to have the following structure:

$$R(n, p) = F(n, p) \cdot \Big(\frac{np}{n_{\mathrm{i}}^2} - 1\Big), \tag{2.46}$$

for some rational function $F(n, p)$, with intrinsic concentration n_{i}. In the constitutive relations, D_n, D_p are the electron and hole diffusivity, respectively, and μ_n,

μ_p are the electron and hole mobility, respectively. Diffusivities and mobilities are functions of $(n, p, \mathbf{E}, \mathbf{x})$. Generally, they satisfy the Einstein's relations

$$D_n = V_{\text{th}} \mu_n, \quad D_p = V_{\text{th}} \mu_p,$$

with thermal potential V_{th}.

The drift-diffusion equations (2.43)–(2.45) are considered for $(\mathbf{x}, t) \in \Omega \times I \subset \mathbb{R}^d \times \mathbb{R}$, $I = [t_0, t_e]$, with the following initial-boundary conditions:

- Boundary conditions for the Poisson equation:

$$\begin{cases} \phi = \phi_{\text{bi}} + u_{D,j}(t), & \text{on } \Gamma_{D,j}, \ j = 0, 1, \ldots, K, \\ \boldsymbol{\nu} \cdot \nabla \phi = 0, & \text{on } \Gamma_N, \end{cases} \tag{2.47}$$

where ϕ_{bi} is the built-in potential, given by

$$\phi_{\text{bi}} = V_{\text{th}} \ln \left(\frac{N_{\text{bi}}}{2n_{\text{i}}} + \sqrt{\left(\frac{N_{\text{bi}}}{2n_{\text{i}}} \right)^2 + 1} \right),$$

$u^i_{D,j}$, $j = 0, 1, \ldots, K$, are the applied potentials at the Ohmic contacts of the device, and $\boldsymbol{\nu}$ is the external unit normal to $\partial\Omega$. Notice that here the time $t \in I$ appears as a parameter, through the boundary data $u_{D,j}(t)$.

- Initial-boundary conditions for the continuity equations:

$$\begin{cases} n = n_{\text{bi}}, \quad p = p_{\text{bi}}, & \text{on } \Gamma_D \times I, \\ \boldsymbol{\nu} \cdot \nabla n = 0, \quad \boldsymbol{\nu} \cdot \nabla p = 0, & \text{on } \Gamma_N \times I, \\ n = n_0, \quad p = p_0, & \text{on } \Omega \times \{t_0\}, \end{cases} \tag{2.48}$$

where the Dirichlet data n_{bi}, p_{bi} are given by

$$n_{\text{bi}} = \frac{N_{\text{bi}}}{2} + \sqrt{\left(\frac{N_{\text{bi}}}{2} \right)^2 + n_{\text{i}}^2}, \ p_{\text{bi}} = -\frac{N_{\text{bi}}}{2} + \sqrt{\left(\frac{N_{\text{bi}}}{2} \right)^2 + n_{\text{i}}^2},$$

and the initial data n_0, p_0 are arbitrary functions. It is interesting to notice the identities $\phi_{\text{bi}} = V_{\text{th}} \ln(n_{\text{bi}}/n_{\text{i}})$, and $n_{\text{bi}} p_{\text{bi}} = n_{\text{i}}^2$.

The total electric current due to the carriers is:

$$\mathbf{J} = -q\mathbf{j}_n + q\mathbf{j}_p. \tag{2.49}$$

It is possible to show that $\mathbf{J}$ satisfies (2.16), with $\rho = -qn + qp$. Then we can apply the formalism described in the previous subsections.

For convenience of the reader, we write below the full coupled system.

(i) **Network equations**:

$$\begin{aligned} A_C \frac{\mathrm{d}\mathbf{q}}{\mathrm{d}t} + A_R \mathbf{r}(A_R^T \mathbf{u}) + A_L \mathbf{i}_L + A_V \mathbf{i}_V + A_I \mathbf{i}_I + \boldsymbol{\lambda} &= 0, \\ \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} - A_L^T \mathbf{u} &= 0, \\ A_V^T \mathbf{u} - \mathbf{v}_V &= 0. \\ \mathbf{q} - \mathbf{q}_C (A_C^T \mathbf{u}) &= 0, \\ \boldsymbol{\phi} - \boldsymbol{\phi}_L (\mathbf{i}_L) &= 0, \end{aligned} \tag{2.50}$$

with initial data for the differential part,

$$\mathbf{P}_C \mathbf{q}(t_0) = \mathbf{P}_C \mathbf{q}_0, \quad \boldsymbol{\phi}(t_0) = \boldsymbol{\phi}_0, \tag{2.51}$$

where $\mathbf{P}_C$ is projector which picks the component of a vector outside the null-space of the incidence matrix A_C [24]. We also need to assume index-1 conditions, that is, the algebraic equations can be solved uniquely for the remaining variables in terms of the differential variables $\mathbf{P}_C \mathbf{q}$, $\boldsymbol{\phi}$.

(ii) **Poisson equation**:

$$-\nabla \cdot (\epsilon \nabla \phi) = q N_{\mathrm{bi}} - q n + q p, \tag{2.52}$$

with boundary data:

$$\begin{cases} \phi = \phi_{\mathrm{bi}} + u_{D,j}(t), & \text{on } \Gamma_{D,j}, \ j = 0, 1, \ldots, K, \\ \nu \cdot \nabla \phi = 0, & \text{on } \Gamma_N. \end{cases} \tag{2.53}$$

(iii) **Device equations**:

$$\begin{aligned} &\frac{\partial n}{\partial t} + \nabla \cdot \mathbf{j}_n = -R, \\ &\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{j}_p = -R, \\ &\mathbf{j}_n = -D_n \nabla n + \mu_n n \nabla \phi, \\ &\mathbf{j}_p = -D_p \nabla p - \mu_p p \nabla \phi, \end{aligned} \tag{2.54}$$

with initial-boundary data:

$$\begin{cases} n = n_{\mathrm{bi}}, \quad p = p_{\mathrm{bi}}, & \text{on } \mathit{\Gamma}_D \times I, \\ \boldsymbol{\nu} \cdot \nabla n = 0, \quad \boldsymbol{\nu} \cdot \nabla p = 0, & \text{on } \mathit{\Gamma}_N \times I, \\ n = n_0, \quad p = p_0, & \text{on } \mathit{\Omega} \times \{t_0\}. \end{cases} \tag{2.55}$$

(iv) **Network-to-device coupling:**

$$\mathbf{v}_D = \mathbf{A}_D^{*\top} \mathbf{u}_D, \qquad \mathbf{u}_D = \mathbf{S}_D^{\top} \mathbf{u}, \tag{2.56}$$

where

$$\mathbf{A}_D^* = \begin{pmatrix} -1 & \cdots & -1 \\ 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix}, \quad \mathbf{u}_D = \begin{pmatrix} u_{D,0} \\ u_{D,1} \\ \vdots \\ u_{D,K} \end{pmatrix}.$$

(v) **Device-to-network coupling:**

$$\boldsymbol{\lambda} = \mathbf{A}_D \mathbf{i}_D, \tag{2.57}$$

with $\mathbf{A}_D = \mathbf{S}_D \mathbf{A}_D^*$, and

$$\mathbf{i}_D = \begin{pmatrix} j_{D,1} \\ \vdots \\ j_{D,K} \end{pmatrix}, \qquad j_{D,i} = -\int_{\mathit{\Gamma}_{D,i}} \mathbf{j} \cdot \boldsymbol{\nu} \, d\sigma, \quad i = 1, \ldots, K, \tag{2.58}$$

where

$$\mathbf{j} := \epsilon \frac{\partial}{\partial t} \mathbf{E} - q \mathbf{j}_n + q \mathbf{j}_p.$$

As we have seen, the device-to-network coupling relation can be replaced by the equivalent relation:

(v)' **Device-to-network coupling (alternative formulation):**

$$\begin{aligned} \boldsymbol{\lambda} &= \mathbf{A}_D \frac{\mathrm{d}\mathbf{q}_D}{\mathrm{d}t} + \mathbf{A}_D \mathscr{I}_D(\mathbf{J}), \\ \mathbf{q}_D &= \mathbf{C}_D \mathbf{v}_D, \end{aligned} \tag{2.59}$$

with $\mathbf{J} = -q\mathbf{j}_n + q\mathbf{j}_p$, and $\mathbf{C}_D = (C_{D,ij}) \in \mathbb{R}^{K\times K}$,

$$C_{D,ij} = \int_\Omega \epsilon \nabla\varphi_i \cdot \nabla\varphi_j \, d\mathbf{x}, \quad i,j = 1,\dots,K, \tag{2.60}$$

where φ_j are defined by (2.33), and

$$\mathscr{I}_D(\mathbf{J}) = \begin{pmatrix} \mathscr{I}_1(\mathbf{J}) \\ \vdots \\ \mathscr{I}_K(\mathbf{J}) \end{pmatrix}, \quad \mathscr{I}_j(\mathbf{J}) = -\int_\Omega \nabla\varphi_j \cdot \mathbf{J}\, d\mathbf{x}.$$

2.2.1.5 Space Discretization of the Distributed Model: The Gummel Map

In this section we discuss the space discretization of the drift-diffusion model, for later use in the following chapter. We need to address two different topics: (1) space discretization of the PDE model, and (2) derivation of discrete device-to-network coupling relations.

Whatever method we use, the space discretization amounts to replacing the space-dependent unknowns, depending on a continuous variable $\mathbf{x} \in \Omega \subset \mathbb{R}^d$, with corresponding index-dependent unknowns, that is, vector unknowns, depending on an index $i \in \mathscr{I} \subset \mathbb{N}$. At the same time, the space-differential operators appearing in the equations are mapped to finite-dimensional operators on $\mathbb{R}^{|\mathscr{I}|}$, with values on the same space. This mapping procedure is achieved, for finite difference methods or Box Integration methods by discretizing the operator itself, while for finite element methods by "discretizing" the functional space on which the original operator acts, that is, by constructing appropriate finite-dimensional functional spaces with dimension $|\mathscr{I}|$.

Since the starting model is generally nonlinear, the discretization is performed after linearizing the system by iteration. The linearization procedure is better discussed at a continuos level. For simplicity, in this discussion we do not write explicitly the initial-boundary conditions. Let us consider the drift-diffusion equations, written in the form:

$$\begin{aligned} \nabla\cdot\mathbf{D} &= qN_{\mathrm{bi}} - qn + qp, \\ \frac{\partial n}{\partial t} + \nabla\cdot\mathbf{j}_n &= -R, \\ \frac{\partial p}{\partial t} + \nabla\cdot\mathbf{j}_p &= -R, \\ \mathbf{D} &= -\epsilon\nabla\phi, \\ \mathbf{j}_n &= -D_n\nabla n + \mu_n n\nabla\phi, \\ \mathbf{j}_p &= -D_p\nabla p - \mu_p p\nabla\phi, \end{aligned} \tag{2.61}$$

where $\mathbf{D}$ is the electric displacement field. In this formulation, we have singled out the fluxes, and after replacing their expressions in the remaining equations, we get a parabolic-elliptic system of partial differential equations. Nonlinearities are present only in the recombination-generation term R, and in the constitutive equations for the carrier density fluxes $\mathbf{j}_n, \mathbf{j}_p$.

The nonlinearities in the constitutive equations are the more delicate to treat because, roughly speaking, the solution of the drift-diffusion equations tends rapidly to the equilibrium solution, in which there is an exponential relationship between the carrier densities and the electric potential. Thus, in a small region, such as a discretization cell, there might be small variations of the electric field and the carrier density fluxes but big variations of the carrier densities. For this reason, it is not convenient to linearize the system in the form written below, and the natural variables n, p are usually transformed into a different set of variables. The Slotboom variables ρ_n, ρ_p are the most common choice. They are defined by the relations:

$$n = n_{\mathrm{i}}\rho_n \exp\left(\frac{\phi}{V_{\mathrm{th}}}\right), \quad p = n_{\mathrm{i}}\rho_p \exp\left(-\frac{\phi}{V_{\mathrm{th}}}\right), \tag{2.62}$$

where n_{i} is the intrinsic concentration and V_{th} is the thermal potential. In equilibrium, the difference

$$np - n_{\mathrm{i}}^2 = n_{\mathrm{i}}^2(\rho_n\rho_p - 1)$$

is identically zero, so we can conclude that equilibrium is characterized by the product of the Slotboom variables to be equal to 1.

In these new variables, system (2.61) becomes

$$\begin{aligned} \nabla\cdot\mathbf{D} &= qN_{\mathrm{bi}} - qn_{\mathrm{i}}\rho_n e^{\phi/V_{\mathrm{th}}} + qn_{\mathrm{i}}\rho_p e^{-\phi/V_{\mathrm{th}}}, \\ \frac{\partial}{\partial t}\left(n_{\mathrm{i}}\rho_n e^{\phi/V_{\mathrm{th}}}\right) + \nabla\cdot\mathbf{j}_n &= -R, \\ \frac{\partial}{\partial t}\left(n_{\mathrm{i}}\rho_p e^{-\phi/V_{\mathrm{th}}}\right) + \nabla\cdot\mathbf{j}_p &= -R, \\ \mathbf{D} &= -\epsilon\nabla\phi, \\ \mathbf{j}_n &= -D_n n_{\mathrm{i}} e^{\phi/V_{\mathrm{th}}}\nabla\rho_n, \\ \mathbf{j}_p &= -D_p n_{\mathrm{i}} e^{-\phi/V_{\mathrm{th}}}\nabla\rho_p. \end{aligned} \tag{2.63}$$

This system is usually solved in three steps, by using an iteration procedure called Gummel map, $(\phi^{k-1}, \rho_n^{k-1}, \rho_p^{k-1}) \mapsto (\phi^k, \rho_n^k, \rho_p^k)$, starting from an initial guess $(\phi^0, \rho_n^0, \rho_p^0)$.

First step We solve the Poisson equation for ϕ^k:

$$\begin{aligned} \nabla \cdot \mathbf{D}^k &= qN_{\mathrm{bi}} - qn_{\mathrm{i}}\rho_n^{k-1}e^{\phi^k/V_{\mathrm{th}}} + qn_{\mathrm{i}}\rho_p^{k-1}e^{-\phi^k/V_{\mathrm{th}}}, \\ \mathbf{D}^k &= -\epsilon\nabla\phi^k. \end{aligned} \tag{2.64}$$

This is a nonlinear problem, so it can be solved by using a modified Raphson-Newton method, which involves another iteration procedure. Starting from an initial guess $\phi^{[0]}$ which satisfies the boundary conditions, given an approximate solution $\phi^{[i-1]}$, we compute the solution $\phi^{[i]}$, given by

$$\begin{aligned} \phi^{[i]} &= \phi^{[i-1]} + \delta\phi^{[i]}, \\ -\nabla \cdot (\epsilon\nabla\delta\phi^{[i]}) &= -\frac{qn_{\mathrm{i}}}{V_{\mathrm{th}}}\left(\rho_n^{k-1}e^{\phi^{[i-1]}/V_{\mathrm{th}}} + \rho_p^{k-1}e^{-\phi^{[i-1]}/V_{\mathrm{th}}}\right)\delta\phi^{[i]} \\ &\quad + \nabla \cdot (\epsilon\nabla\phi^{[i-1]}) + qN_{\mathrm{bi}} - qn_{\mathrm{i}}\rho_n^{k-1}e^{\phi^{[i-1]}/V_{\mathrm{th}}} + qn_{\mathrm{i}}\rho_p^{k-1}e^{-\phi^{[i-1]}/V_{\mathrm{th}}}. \end{aligned}$$

This equation for $\delta\phi^{[i]}$ is linear and can be discretized and solved by using any appropriate numerical method.

Second step We solve the continuity equation for ρ_n^k:

$$\begin{aligned} \frac{\partial}{\partial t}\left(n_{\mathrm{i}}\rho_n^k e^{\phi^k/V_{\mathrm{th}}}\right) + \nabla \cdot \mathbf{j}_n^k &= -R_n^k, \\ \mathbf{j}_n^k &= -D_n^k n_{\mathrm{i}} e^{\phi^k/V_{\mathrm{th}}}\nabla\rho_n^k. \end{aligned} \tag{2.65}$$

Here, the recombination-generation term R_n^k is the usual term R evaluated at ρ_n^{k-1}, ρ_p^{k-1} in such a way to be a linear relaxation term for ρ_n^k. Recalling the general expression (2.46) for $R(n, p)$, it is sufficient to take

$$R_n^k = F(n_{\mathrm{i}}\rho_n^{k-1}e^{\phi^{k-1}/V_{\mathrm{th}}}, n_{\mathrm{i}}\rho_p^{k-1}e^{-\phi^{k-1}/V_{\mathrm{th}}})n_{\mathrm{i}}^2(\rho_n^k\rho_p^{k-1} - 1).$$

As for the diffusivity D_n^k, it is usually dependent on the electric field $\mathbf{E} = -\nabla\phi$, so it should be evaluated at $\phi = \phi^k$. The resulting equation is linear parabolic for the unknown ρ_n^k, and can be discretized and solved by using any appropriate numerical method.

For the discretization of the constitutive relation for $\mathbf{j}_n^k$, exponential interpolation is the most common choice. The basic example is the Scharfetter-Gummel discretization, which provides a formula for the carrier density flux

$$j_{n,ij}^k := \mathbf{j}_n^k \cdot \mathbf{n}_{ij} \equiv D_n^k n_{\mathrm{i}} e^{\phi^k/V_{\mathrm{th}}}\frac{\mathrm{d}\rho_n^k}{\mathrm{d}s},$$

along the line connecting two adjacent grid points $\mathbf{x}_i, \mathbf{x}_j$. In this definition, the vector $\mathbf{n}_{ij} := \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}$ is the unit vector along the segment $[\mathbf{x}_i, \mathbf{x}_j]$, and the parameter s is the line element on the same segment, so that $s_j - s_i = |\mathbf{x}_j - \mathbf{x}_i|$. Assuming that the density flux $j^k_{n,ij}$ and the electric field

$$E^k_{ij} := \mathbf{E}^k \cdot \mathbf{n}_{ij} \equiv -\frac{\mathrm{d}\phi^k}{\mathrm{d}s},$$

are approximately constant along the connecting line, we have

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}s}\left(D^k_n n_{\mathrm{i}} e^{\phi^k/V_{\mathrm{th}}} \frac{\mathrm{d}\rho^k_n}{\mathrm{d}s}\right) = 0, \quad s \in [s_i, s_j],\\
&\rho^k_n(s_i) = \rho^k_{n,i} := \rho^k_n(\mathbf{x}_i), \quad \rho^k_n(s_j) = \rho^k_{n,j} := \rho^k_n(\mathbf{x}_j),
\end{aligned}$$

with

$$\begin{aligned}
&\frac{\mathrm{d}^2\phi^k}{\mathrm{d}s^2} = 0, \quad s \in [s_i, s_j],\\
&\phi^k(s_i) = \phi^k_i := \phi^k(\mathbf{x}_i), \quad \phi^k(s_j) = \phi^k_j := \phi^k(\mathbf{x}_j).
\end{aligned}$$

The result for the electric potential is

$$\frac{\phi^k(s) - \phi^k(s_i)}{s - s_i} = \frac{\phi^k(s_j) - \phi^k(s_i)}{s_j - s_i} \equiv -E^k_{ij},$$

and thus, assuming that the diffusivity depends only on the electric field, we find

$$j^k_{n,ij} = D^k_{n,ij} n_{\mathrm{i}} e^{\phi^k_i/V_{\mathrm{th}}} B\left(\frac{\phi^k_i - \phi^k_j}{V_{\mathrm{th}}}\right) \frac{\rho^k_{n,j} - \rho^k_{n,i}}{|\mathbf{x}_j - \mathbf{x}_i|}, \tag{2.66}$$

where $D^k_{n,ij} = D_n(E^k_{ij})$, and B is the Bernoulli function,

$$B(z) = \begin{cases} \frac{z}{e^z - 1}, & \text{if } z \neq 0,\\ 1, & \text{if } z = 0. \end{cases}$$

Third step We solve the continuity equation for ρ^k_p:

$$\begin{aligned}
\frac{\partial}{\partial t}\left(n_{\mathrm{i}} \rho^k_p e^{-\phi^k/V_{\mathrm{th}}}\right) + \nabla \cdot \mathbf{j}^k_p &= -R^k_p,\\
\mathbf{j}^k_p &= -D^k_p n_{\mathrm{i}} e^{-\phi^k/V_{\mathrm{th}}} \nabla \rho^k_p,
\end{aligned} \tag{2.67}$$

with

$$R^k_p = F(n_{\mathrm{i}} \rho^{k-1}_n e^{\phi^{k-1}/V_{\mathrm{th}}}, n_{\mathrm{i}} \rho^{k-1}_p e^{-\phi^{k-1}/V_{\mathrm{th}}}) n^2_{\mathrm{i}} (\rho^k_n \rho^k_p - 1).$$

As before, the diffusivity D_p^k is evaluated at $\phi = \phi^k$. This equation is linear parabolic for the unknown ρ_p^k, and can be discretized and solved by using any appropriate numerical method. A Scharfetter-Gummel discretization for the hole density flux $\mathbf{j}_p^k$ can be derived using a similar argument as before. The result is

$$j_{p,ij}^k = D_{p,ij}^k n_{\mathrm{i}} e^{-\phi_i^k/V_{\mathrm{th}}} B\left(\frac{\phi_j^k - \phi_i^k}{V_{\mathrm{th}}}\right) \frac{\rho_{n,i}^k - \rho_{n,j}^k}{|\mathbf{x}_j - \mathbf{x}_i|}, \tag{2.68}$$

with obvious notation.

The Gummel map generally converges after few iterations. Instead of separating the original nonlinear problem in three subproblems, it is also possible to apply a Newton-like method to the full system. In either case, we end up with a sequence of linear problems that can be thought as a method for solving a nonlinear differential algebraic system. As we have seen in the description of the Gummel map, it is not simple to obtain an explicit representation of this differential algebraic system, nor is it relevant to know it. In fact, what really matters is the convergence and stability of the method.

For later use in the next chapter, it is nevertheless useful to have at least an explicit example. For this reason we derive a space-discretized system by using the Box Integration method [27, 60]. The discretized coupling conditions will be discussed diffusely for this example, since the general treatment follows along the same line.

2.2.1.6 Space Discretization of the Distributed Model: The Box Integration Method

The Box Integration method consists of two sets of equations – a set of exact equations for the fluxes on the boundaries of the Voronoi cells of a numerical grid, and a set of approximate equations for the fluxes in terms of the value of the unknown function on the grid points. In addition, we need discrete equations for supplementing the boundary conditions. To exemplify the Box Integration method, first we give a rough sketch of its application for the Poisson equation, and then we just show the result of the method for the continuity equations.

Some notation, first. We consider a tessellation $\mathscr{T}_h$ of the domain Ω, which might be a Delaunnay triangulation, a rectangular grid, or a hybrid grid, with vertices (grid points) $\mathscr{X}_h = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and edges $\mathscr{E}_h = \{e_1, \dots, e_M\}$. We also consider the set of the internal grid points, $\mathscr{X}_h' = \{\mathbf{x}_1, \dots, \mathbf{x}_{N'}\}$, and the set $\mathscr{E}_h' = \{e_1, \dots, e_{M'}\}$ of the internal edges, for which at least one of the two end vertices is internal. We denote by $e_{ij} \in \mathscr{E}_h$ the edge which connects the grid points $\mathbf{x}_i, \mathbf{x}_j \in \mathscr{E}_h$. For each grid point $\mathbf{x}_i$, we introduce the set of indices $I(i)$ of the neighboring grid points, that is, $j \in I(i)$ if and only if $e_{ij} \in \mathscr{X}_h$.

We consider the Dirichlet tessellation $\mathscr{D}_h$, dual to $\mathscr{T}_h$, made of the Dirichlet (or Voronoi) cells of the grid points $\mathscr{X}_h$, and we denote by $\mathscr{D}_h'$ the Dirichlet tessellation corresponding to the internal grid points $\mathscr{X}_h'$. We denote by $V_i \in \mathscr{D}_h$ the Voronoi cell of the grid point $\mathbf{x}_i \in \mathscr{X}_h$. The Voronoi cell V_i has at most as many faces as the cardinality of $I(i)$, and we use the notation $v_{ij} = V_i \cap V_j$, $j \in I(i)$. The face v_{ij}, whenever the area $|v_{ij}| \neq 0$, is orthogonal to e_{ij} for any $j \in I(i)$, and equidistant from $\mathbf{x}_i$ and $\mathbf{x}_j$, so the external unit normal on v_{ij}, external with respect to V_i, is $\mathbf{n}_{ij} = \frac{\mathbf{x}_j - \mathbf{x}_i}{|\mathbf{x}_j - \mathbf{x}_i|}$, which we have already encountered when discussing the Scharfetter-Gummel discretization.

Now we are ready to apply the Box Integration method to the Poisson equation

$$\nabla \cdot \mathbf{D} = \rho := qN_{\mathrm{bi}} - qn_{\mathrm{i}}\rho_n e^{\phi/V_{\mathrm{th}}} + qn_{\mathrm{i}}\rho_p e^{-\phi/V_{\mathrm{th}}},$$

$$\mathbf{D} = -\epsilon \nabla \phi,$$

with $\rho = \rho(\mathbf{x}, \rho_n, \rho_p, \phi)$. Integrating the first equation on the internal Voronoi cell $V_i \in \mathscr{D}_h'$, and using the divergence theorem, we get:

$$\sum_{j \in I(i)} \int_{v_{ij}} \mathbf{D} \cdot \mathbf{n}_{ij} \, \mathrm{d}\sigma = \int_{V_i} \rho \, \mathrm{d}\mathbf{x}, \qquad i = 1, \ldots, N'. \tag{2.69}$$

These exact equations are approximated as

$$\sum_{j \in I(i)} |v_{ij}| D_{ij} = |V_i| \rho_i, \qquad i = 1, \ldots, N', \tag{2.70}$$

where $D_{ij} := \mathbf{D} \cdot \mathbf{n}_{ij}$ is evaluated on the mid point of the edge e_{ij}, that is, on $\mathbf{x}_{ij} := \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$, and the index i in the source term denotes evaluation on $\mathbf{x}_i$, in all its arguments.

Next, we need to approximate the flux D_{ij}, and this is done by assuming that the electric field is constant along the edge e_{ij}. Then, we can derive the expression

$$D_{ij} = -\epsilon_{ij} \frac{\phi_j - \phi_i}{|e_{ij}|}, \qquad j \in I(i), \quad i = 1, \ldots, N', \tag{2.71}$$

where the dielectric constant is evaluated on $\mathbf{x}_{ij}$, and is generally approximated by $\epsilon_{ij} \approx \frac{1}{2}(\epsilon_i + \epsilon_j)$.

Using (2.71) in (2.70), we find

$$\sum_{j \in I(i)} |v_{ij}| \epsilon_{ij} \frac{\phi_i - \phi_j}{|e_{ij}|} = |V_i| \rho_i, \qquad i = 1, \ldots, N', \tag{2.72}$$

which is a nonlinear system of N' equations for the N unknowns $\phi_1, \ldots, \phi_N$. In compact form, we can write

$$\mathbf{A}_\phi \boldsymbol{\phi} + \mathbf{A}_\phi^\partial \boldsymbol{\phi}^\partial = \mathbf{b}_\phi(\boldsymbol{\phi}, \boldsymbol{\rho}_n, \boldsymbol{\rho}_p), \tag{2.73}$$

with $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_{N'})^\top$, $\boldsymbol{\phi}^\partial = (\phi_{N'+1}, \ldots, \phi_N)^\top$. This equation is the discrete analog of the Poisson equation.

The remaining $N - N'$ equations, needed to determine the unknowns, come from the boundary conditions. We have $N - N' = N_D + N_N$, where N_D is the number of nodes on Γ_D, and N_N the number of nodes on Γ_N. It is simple to impose N_D Dirichlet conditions,

$$\phi_i = \phi_{\mathrm{bi},i} + u_{D,k}, \quad \text{if } \mathbf{x}_i \in \Gamma_{D,k}. \tag{2.74}$$

It is a bit more complicated to impose N_N Neumann conditions, at least in the framework of the Box Integration method. A possible way of doing it, is by using a BDF formula for expressing the normal derivative on a Neumann grid point in terms of inner grid points along the normal direction, possibly with the help of some interpolation. Whatever method we use, we end up with N_N equations of the form

$$\phi_i + \sum_{j=1}^{N'} a_{ij}\phi_j = 0, \quad \text{if } \mathbf{x}_i \in \Gamma_N, \tag{2.75}$$

with many zero coefficients. Combining Eqs. (2.74) and (2.75), we can write them in the compact form

$$\mathbf{A}^\partial \boldsymbol{\phi} + \boldsymbol{\phi}^\partial = \mathbf{b}_\phi^\partial(\mathbf{u}_D). \tag{2.76}$$

We notice that the matrix $\mathbf{A}^\partial$ does not depend on the differential equation but only on the tessellation $\mathscr{T}_h$ and on the formula used for expressing the normal derivative with respect to the internal nodes. Equation (2.76) is the discrete analogue of the boundary conditions for the Poisson equation, and together with (2.73) form a set of equations which can be solved for $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^\partial$.

We can apply the same procedure to the electron continuity equation,

$$\begin{aligned} \frac{\partial n}{\partial t} + \nabla \cdot \mathbf{j}_n &= -R, \\ \mathbf{j}_n &= -D_n n_\mathrm{i} e^{\phi/V_\mathrm{th}} \nabla \rho_n, \end{aligned} \tag{2.77}$$

and to the hole continuity equation,

$$\begin{aligned} \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{j}_p &= -R, \\ \mathbf{j}_p &= -D_p n_\mathrm{i} e^{-\phi/V_\mathrm{th}} \nabla \rho_p, \end{aligned} \tag{2.78}$$

with n, p given in terms of ρ_n, ρ_p and ϕ by (2.62). Using the Scharfetter-Gummel discretization (2.66) and (2.68) for the fluxes, we obtain the discretized equations

$$
\begin{aligned}
&|V_i|\frac{\mathrm{d}n_i}{\mathrm{d}t} + \sum_{j\in I(i)} |v_{ij}|\, j_{n,ij} = -|V_i|R_i,\\
&j_{n,ij} = D_{n,ij} n_{\mathrm{i}} e^{\phi_i/V_{\mathrm{th}}} B\left(\frac{\phi_i-\phi_j}{V_{\mathrm{th}}}\right)\frac{\rho_{n,j}-\rho_{n,i}}{|e_{ij}|}, \quad j \in I(i),
\end{aligned}
\tag{2.79}
$$

and

$$
\begin{aligned}
&|V_i|\frac{\mathrm{d}p_i}{\mathrm{d}t} + \sum_{j\in I(i)} |v_{ij}|\, j_{p,ij} = -|V_i|R_i,\\
&j_{p,ij} = D_{p,ij} n_{\mathrm{i}} e^{-\phi_i/V_{\mathrm{th}}} B\left(\frac{\phi_j-\phi_i}{V_{\mathrm{th}}}\right)\frac{\rho_{n,i}-\rho_{n,j}}{|e_{ij}|}, \quad j \in I(i),
\end{aligned}
\tag{2.80}
$$

with $i = 1, \dots, N'$. To these equations we need to add the discrete Dirichlet and Neumann boundary conditions for both equations,

$$
\rho_{n,i} = e^{-u_{D,k}/V_{\mathrm{th}}}, \quad \text{if } \mathbf{x}_i \in \Gamma_{D,k}, \tag{2.81}
$$

$$
\rho_{n,i} + \sum_{j=1}^{N'} a_{ij}\rho_{n,j} = 0, \qquad \text{if } \mathbf{x}_i \in \Gamma_N, \tag{2.82}
$$

and

$$
\rho_{p,i} = e^{u_{D,k}/V_{\mathrm{th}}}, \quad \text{if } \mathbf{x}_i \in \Gamma_{D,k}, \tag{2.83}
$$

$$
\rho_{p,i} + \sum_{j=1}^{N'} a_{ij}\rho_{p,j} = 0, \qquad \text{if } \mathbf{x}_i \in \Gamma_N. \tag{2.84}
$$

In compact form, the spatially discrete continuity equations can be written as:

$$
\mathbf{A}_0\frac{\mathrm{d}\boldsymbol{n}(\boldsymbol{\phi},\boldsymbol{\rho}_n)}{\mathrm{d}t} + \mathbf{A}_n(\boldsymbol{\phi})\boldsymbol{\rho}_n + \mathbf{A}_n^{\partial}(\boldsymbol{\phi})\boldsymbol{\rho}_n^{\partial} = \mathbf{b}_n(\boldsymbol{\phi},\boldsymbol{\rho}_n,\boldsymbol{\rho}_p), \tag{2.85}
$$

$$
\mathbf{A}^{\partial}\boldsymbol{\rho}_n + \boldsymbol{\rho}_n^{\partial} = \mathbf{b}_n^{\partial}(\mathbf{u}_D), \tag{2.86}
$$

$$
\mathbf{A}_0\frac{\mathrm{d}\boldsymbol{p}(\boldsymbol{\phi},\boldsymbol{\rho}_p)}{\mathrm{d}t} + \mathbf{A}_p(\boldsymbol{\phi})\boldsymbol{\rho}_p + \mathbf{A}_p^{\partial}(\boldsymbol{\phi})\boldsymbol{\rho}_p^{\partial} = \mathbf{b}_p(\boldsymbol{\phi},\boldsymbol{\rho}_n,\boldsymbol{\rho}_p), \tag{2.87}
$$

$$
\mathbf{A}^{\partial}\boldsymbol{\rho}_p + \boldsymbol{\rho}_p^{\partial} = \mathbf{b}_p^{\partial}(\mathbf{u}_D), \tag{2.88}
$$

with notation analogous to the one used for the discretized Poisson equation (2.73) and (2.76). Besides the presence of the time derivative, the main difference is that now the matrices corresponding to the elliptic operators, that is, $\mathbf{A}_n$ and $\mathbf{A}_p$, depend nonlinearly on the electric potential.

We notice that, within the Box Integration method framework, other spatial discretizations are possible. In particular, we can start from the drift-diffusion system written for the natural variables ϕ, n, p. In this case, the discrete Poisson equation becomes linear and it is possible to write the Scharfetter-Gummel discretization for $j_{n,ij}$ as a linear combination of n_i, n_j, with coefficients depending nonlinearly on the electric potential,

$$j_{n,ij} = \frac{D_{n,ij}}{|e_{ij}|}\left(B\left(\frac{\phi_j-\phi_i}{V_{\text{th}}}\right)n_j - B\left(\frac{\phi_i-\phi_j}{V_{\text{th}}}\right)n_i\right), \tag{2.89}$$

$$j_{p,ij} = \frac{D_{p,ij}}{|e_{ij}|}\left(B\left(\frac{\phi_i-\phi_j}{V_{\text{th}}}\right)p_j - B\left(\frac{\phi_j-\phi_i}{V_{\text{th}}}\right)p_i\right). \tag{2.90}$$

Then, we obtain a linear ordinary differential equation for $\boldsymbol{n}$, with coefficients depending nonlinearly on the electric potential, and similarly for $\boldsymbol{p}$. This form looks much simpler than the one we have derived above, but it becomes unstable if we try to decouple the three main problems by iteration, as in the Gummel map. Nevertheless it can be used if the system is solved by Newton iteration, without using the Gummel map. For this reason, we will apply it in the next chapter, and we summarize it as follows:

$$\mathbf{A}_\phi \boldsymbol{\phi} + \mathbf{A}_\phi^\partial \boldsymbol{\phi}^\partial = \mathbf{b}_\phi(\boldsymbol{n}, \boldsymbol{p}), \tag{2.91}$$

$$\mathbf{A}^\partial \boldsymbol{\phi} + \boldsymbol{\phi}^\partial = \mathbf{b}_\phi^\partial(\mathbf{u}_D), \tag{2.92}$$

$$\mathbf{A}_0 \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}t} + \mathbf{A}_n(\boldsymbol{\phi})\boldsymbol{n} + \mathbf{A}_n^\partial(\boldsymbol{\phi})\boldsymbol{n}^\partial = \mathbf{b}_n(\boldsymbol{n}, \boldsymbol{p}), \tag{2.93}$$

$$\mathbf{A}^\partial \boldsymbol{n} + \boldsymbol{n}^\partial = \mathbf{b}_n^\partial, \tag{2.94}$$

$$\mathbf{A}_0 \frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} + \mathbf{A}_p(\boldsymbol{\phi})\boldsymbol{p} + \mathbf{A}_p^\partial(\boldsymbol{\phi})\boldsymbol{p}^\partial = \mathbf{b}_p(\boldsymbol{n}, \boldsymbol{p}), \tag{2.95}$$

$$\mathbf{A}^\partial \boldsymbol{p} + \boldsymbol{p}^\partial = \mathbf{b}_p^\partial. \tag{2.96}$$

Note that Eqs. (2.94) and (2.96) do not depend on $\mathbf{u}_D$, because the Dirichlet boundary conditions for the variables n, p are now given by (2.55).

2.2.1.7 Space Discretization of the Distributed Model: The Coupling Conditions

The last item to be discussed is the coupling conditions with the network. The network-to-device coupling condition is immediate, because the term $\mathbf{b}_\phi^\partial$ (in the formulation with the Slotboom variables, also $\mathbf{b}_n^\partial$ and $\mathbf{b}_p^\partial$) depends on the applied potentials $\mathbf{u}_D$, which are related to the network node potentials by the coupling relation (2.56). The device-to-network coupling is more delicate, because we need to introduce the discretized current transmitted to the network through the k-th Ohmic contact, $\Gamma_{D,k}$, $k = 1, \ldots, K$.

First, we implement the coupling condition as in (2.57). At this aim, we consider the Voronoi cells V_i corresponding to grid nodes $\mathbf{x}_i$ in $\Gamma_{D,k}$, and we integrate the charge conservation equation on the union of these Voronoi cells, $V_{D,k} = \bigcup_{\mathbf{x}_i \in \Gamma_{D,k}} V_i$:

$$\int_{V_{D,k}} \left[\frac{\partial}{\partial t}(-qn + qp) + \nabla \cdot (-q\mathbf{j}_n + q\mathbf{j}_p) \right] \mathrm{d}\mathbf{x} = 0. \tag{2.97}$$

Using Poisson's equation, we find

$$\frac{\partial}{\partial t}(-qn + qp) = \nabla \cdot \frac{\partial \mathbf{D}}{\partial t}, \qquad \mathbf{D} = \epsilon \mathbf{E} = -\epsilon \nabla \phi.$$

Then, by the divergence theorem, we can write

$$\begin{aligned} \int_{V_{D,k}} \nabla \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\mathbf{x} &= \int_{\partial V_{D,k}} \mathbf{n} \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\sigma \\ &= \int_{\partial V_{D,k} \cap \partial \Omega} \mathbf{n} \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\sigma \\ &\quad + \int_{\partial V_{D,k} \setminus \partial \Omega} \mathbf{n} \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\sigma = 0. \end{aligned}$$

The first integral is approximately the outer current flux through the Ohmic contact $\Gamma_{D,k}$, that is, with our convention,

$$\int_{\partial V_{D,k} \cap \partial \Omega} \mathbf{n} \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\sigma \approx -j_{D,k}.$$

The maximum error in this approximation occurs when the grid points on $\Gamma_{D,k}$ closer to the neighboring Neumann boundary are located on the junction between Dirichlet and Neumann boundary, in which case $\partial V_{D,k} \cap \partial \Omega$ consists of $\Gamma_{D,k}$ bordered with a strip whose thickness is the order of half the diameter of the Voronoi cells. On the other hand, we can write

$$\begin{aligned} &\int_{\partial V_{D,k} \setminus \partial \Omega} \mathbf{n} \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\sigma \\ &= \sum_{\mathbf{x}_i \in \Gamma_{D,k}} \sum_{\substack{j \in I(i) \\ \mathbf{x}_j \notin \Gamma_{D,k}}} \int_{v_{ij}} \mathbf{n}_{ij} \cdot \left[\frac{\partial \mathbf{D}}{\partial t} - q\mathbf{j}_n + q\mathbf{j}_p \right] \mathrm{d}\sigma \\ &\approx \sum_{\mathbf{x}_i \in \Gamma_{D,k}} \sum_{\substack{j \in I(i) \\ \mathbf{x}_j \notin \Gamma_{D,k}}} |v_{ij}| \left[\frac{\mathrm{d}D_{ij}}{\mathrm{d}t} - qj_{n,ij} + qj_{p,ij} \right], \end{aligned}$$

where D_{ij}, $j_{n,ij}$, $j_{p,ij}$ are defined as in (2.71) and (2.79), (2.80), or (2.89), (2.90). Combining the previous relations we find the approximation

$$j_{D,k} = \sum_{\mathbf{x}_i \in \Gamma_{D,k}} \sum_{\substack{j \in I(i) \\ \mathbf{x}_j \notin \Gamma_{D,k}}} |v_{ij}| \left[\frac{\mathrm{d}D_{ij}}{\mathrm{d}t} - q j_{n,ij} + q j_{p,ij} \right], \tag{2.98}$$

which can be used as device-to-network discrete coupling condition. In short, recalling the definition of the coupling term $\boldsymbol{\lambda}$, we can write

$$\boldsymbol{\lambda} = \mathbf{A}_D \mathbf{i}_D, \quad \mathbf{i}_D = \mathbf{A}^c \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} + \mathbf{A}_n^c(\boldsymbol{\phi})\mathbf{n} + \mathbf{A}_p^c(\boldsymbol{\phi})\mathbf{p}. \tag{2.99}$$

We note that in this coupling condition, the time derivative of D_{ij} occurs, that is, the time derivative of $\boldsymbol{\phi}$, which is an “algebraic variable” for the discretized device equations with no coupling.

Next, we formulate the discrete version of the alternative formulation of the device-to-network coupling conditions (2.59). We need to evaluate the capacitance matrix $\mathbf{C}_D$, defined by (2.60), and to formulate the discrete version of the operator $\mathscr{I}_k(\mathbf{J})$ appearing in (2.59). As for the capacitance matrix, we can write

$$\begin{aligned}
C_{D,kl} &= \sum_{i=1}^{N} \int_{V_i} \epsilon \nabla \varphi_k \cdot \nabla \varphi_l \, d\mathbf{x} \\
&= \sum_{i=1}^{N} \left[\int_{\partial V_i} \epsilon \varphi_k \nabla \varphi_l \cdot \mathbf{n} \, \mathrm{d}\sigma - \int_{V_i} \varphi_k \nabla \cdot (\epsilon \nabla \varphi_l) \, d\mathbf{x} \right] \\
&= \sum_{i=1}^{N} \int_{\partial V_i} \epsilon (\varphi_k - \varphi_{k,i}) \nabla \varphi_l \cdot \mathbf{n} \, \mathrm{d}\sigma,
\end{aligned}$$

where φ_j are defined by (2.33), and $\varphi_{k,i} = \varphi_k(\mathbf{x}_i)$. The last equality follows because $\nabla \cdot (\epsilon \nabla \varphi_l)$ is identically zero due to the definition of φ_l. If $\mathbf{x}_i \in \mathscr{X}_h'$, this integral can be approximated by

$$\begin{aligned}
\int_{\partial V_i} \epsilon (\varphi_k - \varphi_{k,i}) \nabla \varphi_l \cdot \mathbf{n} \, \mathrm{d}\sigma &= \sum_{j \in I(i)} \int_{v_{ij}} \epsilon (\varphi_k - \varphi_{k,i}) \nabla \varphi_l \cdot \mathbf{n}_{ij} \, \mathrm{d}\sigma \\
&\approx \sum_{j \in I(i)} |v_{ij}| \epsilon_{ij} (\varphi_{k,ij} - \varphi_{k,i}) \frac{\varphi_{l,j} - \varphi_{l,i}}{|e_{ij}|},
\end{aligned}$$

with $\epsilon_{ij} = \epsilon(\mathbf{x}_{ij}) \approx \frac{1}{2}(\epsilon_i + \epsilon_j)$, $\varphi_{k,ij} = \varphi_k(\mathbf{x}_{ij}) \approx \frac{1}{2}(\varphi_{k,i} + \varphi_{k,j})$. Then, we find the following approximation:

$$\int_{V_i} \epsilon \nabla\varphi_k \cdot \nabla\varphi_l \, d\mathbf{x} \approx \sum_{j\in I(i)} \frac{|v_{ij}|}{2|e_{ij}|}\epsilon_{ij}(\varphi_{k,j} - \varphi_{k,i})(\varphi_{l,j} - \varphi_{l,i}). \tag{2.100}$$

If $\mathbf{x}_i \in \mathscr{X}_h \cap \partial\Omega$, we have

$$\begin{aligned}\int_{\partial V_i} \epsilon(\varphi_k - \varphi_{k,i})\nabla\varphi_l \cdot \mathbf{n}\,\mathrm{d}\sigma = &\sum_{j\in I(i)} \int_{v_{ij}} \epsilon(\varphi_k - \varphi_{k,i})\nabla\varphi_l \cdot \mathbf{n}_{ij}\,\mathrm{d}\sigma \\ &+ \int_{\partial V_i \cap \partial\Omega} \epsilon(\varphi_k - \varphi_{k,i})\nabla\varphi_l \cdot \mathbf{n}\,\mathrm{d}\sigma.\end{aligned}$$

The second integral vanishes because either $\nabla\varphi_l \cdot \mathbf{n} = 0$, if V_i touches the Neumann boundary, or $\varphi_k - \varphi_{k,i} = 0$, if V_i touches a Dirichlet boundary, so we are led to the same approximation (2.100).

In conclusion, the capacitance matrix is approximated by

$$\tilde{C}_{D,kl} = \sum_{i=1}^{N} \sum_{j\in I(i)} \frac{|v_{ij}|}{2|e_{ij}|}\epsilon_{ij}(\varphi_{k,j} - \varphi_{k,i})(\varphi_{l,j} - \varphi_{l,i}). \tag{2.101}$$

In a similar way, we can approximate $\mathscr{I}_k(\mathbf{J})$. We can write

$$\begin{aligned}\mathscr{I}_k(\mathbf{J}) &= -\sum_{i=1}^{N} \int_{V_i} \nabla\varphi_k \cdot \mathbf{J}\, d\mathbf{x} \\ &= -\sum_{i=1}^{N} \left[\int_{\partial V_i} \varphi_k \mathbf{J} \cdot \mathbf{n}\,\mathrm{d}\sigma - \int_{V_i} \varphi_k \nabla \cdot \mathbf{J}\, d\mathbf{x}\right] \\ &\approx -\sum_{i=1}^{N} \int_{\partial V_i} (\varphi_k - \varphi_{k,i})\mathbf{J} \cdot \mathbf{n}\,\mathrm{d}\sigma,\end{aligned}$$

so an approximation is given by

$$\tilde{\mathscr{I}}_k = -\sum_{i=1}^{N} \sum_{j\in I(i)} \frac{|v_{ij}|}{2}(\varphi_{k,j} - \varphi_{k,i})(-qj_{n,ij} + qj_{p,ij}). \tag{2.102}$$

In short, the coupling conditions can be written as:

$$
\begin{aligned}
\boldsymbol{\lambda} &= \mathbf{A}_D \frac{\mathrm{d}\mathbf{q}_D}{\mathrm{d}t} + \mathbf{A}_D \tilde{\mathscr{I}}_D, \quad \tilde{\mathscr{I}}_D = \mathbf{A}_n^c(\boldsymbol{\phi})\mathbf{n} + \mathbf{A}_p^c(\boldsymbol{\phi})\mathbf{p}, \\
\mathbf{q}_D &= \tilde{\mathbf{C}}_D \mathbf{v}_D \equiv \tilde{\mathbf{C}}_D \mathbf{A}_D^\top \mathbf{u}.
\end{aligned}
\tag{2.103}
$$

2.2.2 Electro-Thermal Effects at the System Level

The typical trend associating new technology generations with a reduced power consumption has been reversed in the last decade making an accurate electro-thermal analysis of ICs a necessity for a reliable and cost-effective design. To support this need computer aided design (CAD) tools must provide dependable means to simulate coupled electro-thermal effects.

The development of a robust algorithm for this purpose requires a high degree of integration inside usual industrial design flows to be effectively usable, and the possibility to account for 2D/3D heat diffusion to properly describe thermal effects at the system level. In particular it should allow an efficient handling of the space-time multiscale effects associated with the problem at hand. Figure 2.1 shows a brief sketch of a new strategy (originally proposed in [20]) to automatically perform system level electro-thermal simulations inside an industrial design flow.

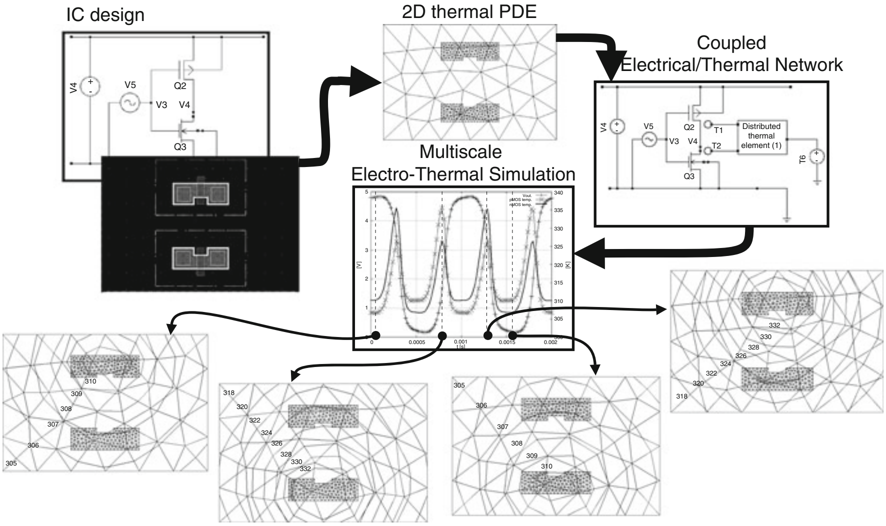


Fig. 2.1 Automated design flow for the electro-thermal simulation of ICs. A thermal element model is automatically constructed from available circuit schematic and design layout, permitting the set-up and simulation of an electro-thermal network that accounts for heat diffusion at the system level

In this approach the electrical behavior of possibly each circuit element is modeled by standard compact models with an added temperature node. Mutual heating is then accounted for by a *novel circuital element* embedding a 2D or 3D diffusion-reaction partial differential equation (PDE) in its constitutive relations to describe heat-diffusion on a distributed domain. By imposing suitable integral conditions this element is casted in a form analogous to that of usual electrical circuit elements, so that its use in a standard circuit simulator requires only the implementation of a new element evaluator, but no modification to the main structure of the solver. This permits the automatic set-up and simulation of an electro-thermal network that accounts for heat diffusion at the system level.

2.2.2.1 Definition of the PDE-Based Thermal Element Model

A suitable thermal element balancing power fluxes at junction temperature nodes is required to extend a purely electrical description of a circuit to an electro-thermal one. In the following it is shown how a multiscale model that fits such a purpose can be derived starting from information that are readily available during IC design phase, i.e. 2D or 3D layout geometry and possibly 3D package geometry.

As sketched in Fig. 2.2 this information is used to describe the overall physical region where to simulate thermal effects as an open, bounded domain:

$$\Omega \subset \mathbb{R}^d \quad (d = 2, 3),$$

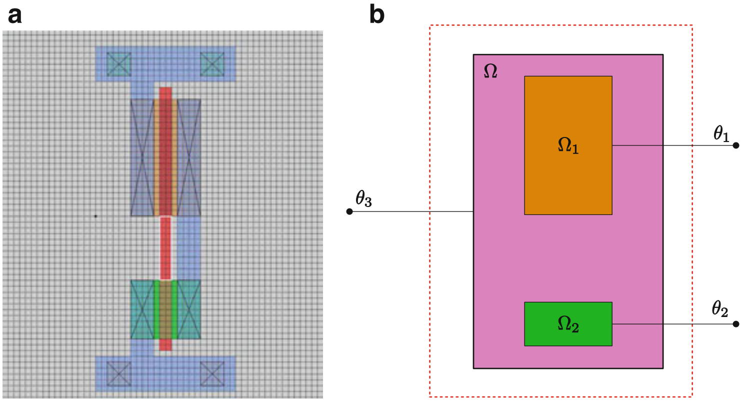


Fig. 2.2 Layout or package information from IC design are automatically converted into a geometrical description of the domains in which suitable PDEs describing heat diffusion at the system level are casted. Notice that while θ_1 and θ_2 refer to mean temperature values over Ω_1 and Ω_2 respectively, θ_3 represents ambient temperature. (**a**) Inverter layout. (**b**) Extracted geometry

and to associate each thermally active device with a subset related to its layout positioning:

$$\Omega_k \subset \Omega \text{ for } k = 1, \ldots, K$$

where its power flux is supposed to be dissipated. Each subset is required one to fulfill the following properties:

$$\begin{aligned} \mathtt{int}(\Omega_k) &\neq \emptyset && \forall k = 1, \ldots, K, \\ \bar{\Omega}_k &\subset \Omega && \forall k = 1, \ldots, K, \\ \bar{\Omega}_k \cap \bar{\Omega}_j &= \emptyset \ \forall j, k = 1, \ldots, K, && k \neq j. \end{aligned}$$

Furthermore it is supposed for either Ω and Ω_k $(k = 1, \ldots, K)$ to have Lipschitz boundary. The unknowns considered in the thermal element model are the junction temperature vector:

$$\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{K+1}]^T \ ,$$

where the first K components are associated with each subset region while the last one represents ambient temperature, the power density vector:

$$\mathbf{p} = [p_1, \ldots, p_K]^T \ ,$$

where each component represents the Joule power per unit area dissipated in each region and the distributed temperature field $T(\mathbf{x}, t)$ on Ω.

Assuming $(\cdot, \cdot)$ to denote the usual $\mathbb{L}^2(\Omega)$ scalar product and $\mathbf{1}_{\Omega_k}$ to denote the indicator function of the set Ω_k, then the distributed temperature field $T(\mathbf{x}, t)$ is linked to junction temperature nodes through:

$$\frac{1}{|\Omega_k|}(T, \mathbf{1}_{\Omega_k}) = \theta_k \text{ for } k = 1, \ldots, K,$$

i.e. θ_k represents the mean value over Ω_k of $T(\mathbf{x}, t)$. In the same way the power flux entering each node is related to the Joule power per unit area via:

$$(p_k, \mathbf{1}_{\Omega_k}) = p_k |\Omega_k| = P_k \text{ for } k = 1, \ldots, K.$$

The total power P_k dissipated over Ω_k is thus equal, for every fixed time instant, to the product of a mean power density p_k times the area of each active region $|\Omega_k|$. Finally the power flux to ambient temperature node is defined to be:

$$P_{K+1} = -\sum_{k=1}^{K} p_k |\Omega_k|.$$

to ensure energy conservation inside the thermal element. Though the decisions to uniformly distribute the dissipated power P_k inside Ω_k and define θ_k as the mean temperature over Ω_k are somehow arbitrary, they constitute a sound physical approximation at a macro-scale level, if it is considered that usually:

$$\mathrm{diam}(\Omega_k) \ll \mathrm{diam}(\Omega) \text{ for } k = 1, \dots, K.$$

Anyhow, other shapes for the power distribution inside Ω_k, as well as any other means to define junction temperatures starting from the distributed field $T(\mathbf{x}, t)$ may have been adopted in principle.

If packaging information is available, then heat-diffusion on a 3D domain is supposed to be modeled by a quasi-linear PDE:

$$c_V(T, \mathbf{x}) \frac{\partial T(\mathbf{x}, t)}{\partial t} + \mathscr{L}_3\, T(\mathbf{x}, t) = \sum_{k=1}^{K} p_k(t)\, \mathbf{1}_{\Omega_k}(\mathbf{x}) \qquad \text{in } \Omega, \tag{2.104}$$

where:

$$\mathscr{L}_3\, T(\mathbf{x}, t) := -\sum_{i,j=1}^{3} D_i \Big[\kappa_{ij}(T, \mathbf{x}) D_j\, T(\mathbf{x}, t) \Big], \quad D_i := \frac{\partial}{\partial x_i}. \tag{2.105}$$

In (2.104) the term $c_V(T, \mathbf{x})$ represents the distributed thermal capacitance of the material, while in (2.105) the terms $\kappa_{ij}(T, \mathbf{x})$ $(i, j = 1, \dots, 3)$ account for possibly anisotropic heat-diffusion. A common assumption, stemming from physical considerations, is that:

$$\kappa_{ij}(T, \mathbf{x}) = \kappa_{ji}(T, \mathbf{x}),$$

so that the associated tensor results to be symmetric. This PDE has to be complemented by suitable boundary conditions that are, here and in the following, assumed to be of Robin type:

$$\frac{\partial\, T(\mathbf{x}, t)}{\partial\, \mathbf{n}_{\mathscr{L}}} = R(T, \theta_{K+1}) \quad \text{on } \partial\Omega. \tag{2.106}$$

In (2.106) the term $\dfrac{\partial\, T(\mathbf{x}, t)}{\partial\, \mathbf{n}_{\mathscr{L}}}$ denotes the conormal derivative of $T(\mathbf{x}, t)$ on $\partial\Omega$ and is defined as:

$$\frac{\partial\, T(\mathbf{x}, t)}{\partial\, \mathbf{n}_{\mathscr{L}}} := \sum_{i,j=1}^{3} n_i \kappa_{ij} D_j\, T(\mathbf{x}, t)\ ,$$

where n_i is the i-th component of the normal outward oriented unit vector on $\partial\Omega$.

In the case that only layout information is available, or that package temperature field is not of interest, then heat diffusion can be modeled by a quasi-linear PDE similar to the one used in the 3D case:

$$\hat{c}_V(T,\mathbf{x})\frac{\partial T(\mathbf{x},t)}{\partial t} + \mathscr{L}_2\, T(\mathbf{x},t) = \sum_{k=1}^{K} p_k(t)\, \mathbf{1}_{\Omega_k}(\mathbf{x}) \qquad \text{in } \Omega,$$

the only difference being that now the operator $\mathscr{L}_2$, defined as:

$$\mathscr{L}_2\, T(\mathbf{x},t) := -\sum_{i,j=1}^{2} D_i\left[\hat{\kappa}_{ij}(T,\mathbf{x}) D_j T(\mathbf{x},t)\right] + \hat{c}(T,\mathbf{x})T(\mathbf{x},t),$$

embodies a reaction term $\hat{c}(T,\mathbf{x})$ to model heat loss in the missing third direction. Suitable boundary conditions are needed also in this case to close the model.

2.2.2.2 Analysis of the Thermal Element Model

The well-posedness of the thermal element model when externally controlled by independent sources fixing the Joule power per unit area stems directly from its definition in Sect. 2.2.2.1. The reader interested in a broader treatment of this subject is referred to [20, Chapter 3]. Existence and uniqueness of a solution can also be proven in the case where the external independent sources fix the average temperature over a region. In particular, a result of this type is given in this section.

In the case at hand heat-diffusion processes are restricted to the case of the linear operator:

$$\mathscr{L}\, T(\mathbf{x}) := -\sum_{i,j=1}^{d} D_i\left[\kappa_{ij}(\mathbf{x}) D_j T(\mathbf{x})\right] + c(\mathbf{x})T(\mathbf{x}), \tag{2.107}$$

where $\kappa_{ij}(\mathbf{x}), c(\mathbf{x}) \in \mathbb{L}^{\infty}(\Omega)$ and:

$$\begin{aligned} c(\mathbf{x}) &\geq 0 \qquad \text{a.e. in } \Omega\ , \\ \kappa_{ij}(\mathbf{x}) &= \kappa_{ji}(\mathbf{x})\ i,j = 1,\dots,d\ . \end{aligned}$$

Furthermore it is assumed for $\mathscr{L}$ to be uniformly elliptic in Ω, i.e. it exists $\tau > 0$ such that:

$$\sum_{i,j=1}^{d} \kappa_{ij}(\mathbf{x})\xi_j\xi_i \geq \tau\, |\boldsymbol{\xi}|^2\,, \tag{2.108}$$

for each $\boldsymbol{\xi} \in \mathbb{R}^d$ and almost every $\mathbf{x} \in \Omega$. The PDE employed to describe thermal effects is enforced in a weak formulation that reads:

$$\frac{d}{dt}(T, v) + a(T, v) + \hat{\alpha}(T - \theta_{K+1}, v)_{\partial\Omega} = \sum_{k=1}^{K} p_k (\mathbf{1}_{\Omega_k}, v) \,, \tag{2.109}$$

where:

$$a(T, v) := \int_{\Omega} \Big(\sum_{i,j=1}^{d} \kappa_{ij}(\mathbf{x}) \; D_j T \; D_i v \Big) \, d\mathbf{x} + \int_{\Omega} c(\mathbf{x}) \; T \; v \; d\mathbf{x}. \tag{2.110}$$

is the bilinear form associated with $\mathscr{L}$, while $(\cdot,\cdot)_{\partial\Omega}$ denote the $\mathbb{L}^2(\partial\Omega)$ scalar product. Under these hypothesis it is possible to prove the following:

Theorem 2.1 *Given:*

1. $T_0 \in \mathbb{L}^2(\Omega)$,
2. $\theta_k \in \mathbb{C}^0[0, \mathrm{t}_1]$ *and* $\theta_k(0)$ *consistent with* T_0 $(k = 1, \ldots, K)$,
3. $\theta_{K+1} \in \mathbb{C}^0[0, \mathrm{t}_1]$,

there exist unique:

1. $T \in \mathbb{C}^0\big([0, \mathrm{t}_1]; \mathbb{L}^2(\Omega)\big) \cap \mathbb{L}^2\big((0, \mathrm{t}_1); \mathbb{H}^1(\Omega)\big)$,
2. $p_k \in \mathbb{C}^0[0, \mathrm{t}_1]$ $(k = 1, \ldots, K)$,

such that:

$$\begin{aligned} \frac{d}{dt}(T, v) + a(T, v) + (\hat{\alpha} T, v)_{\partial\Omega} &= \sum_{k=1}^{K} p_k (\mathbf{1}_{\Omega_k}, v) + (\hat{\alpha}\theta_{K+1}, v)_{\partial\Omega} \\ &\qquad \textit{for all } v \in \mathbb{H}^1(\Omega), \\ T(\mathbf{x}, 0) &= T_0(\mathbf{x}), \\ (T, \mathbf{1}_{\Omega_k}) &= \theta_k(t) |\Omega_k| \quad \textit{for } k = 1, \ldots, K. \end{aligned}$$

Readers interested in the proof of this theorem are referred to [20, Chapter 3.2], where further considerations on the practical role played by Theorem 2.1 and its elliptic counterpart are also given.

2.2.2.3 Evaluation of the Thermal Element Model

The structure most commonly adopted in the design of a software package for transient circuit simulation is usually based upon a set of *element evaluators* that provide a non-linear solver with the local Jacobian matrices and residuals needed to assemble the linearized system corresponding to each Newton iteration. These local contributions, commonly referred to as *stamps*, completely define the behavior of each circuit element and are usually represented in a table-like format [20, Chapter 5]. In the following the stamp associated with the thermal element model defined in Sect. 2.2.2.1 will be given.

Introduce to this aim the vectors:

$$\begin{aligned}\boldsymbol{\theta}_k &= [\theta_1(t_k), \dots, \theta_{K+1}(t_k)]^T, \\ \mathbf{p}_k &= [p_1(t_k), \dots, p_K(t_k)]^T, \\ \mathbf{T}_k &= \left[\mathbf{T}_C(t_k), \mathbf{T}_1(t_k), \dots, \mathbf{T}_K(t_k)\right]^T,\end{aligned}$$

associated with the thermal element unknowns at the time instant t_k. The particular structure of the vector $\mathbf{T}_k$ stems from the space discretization of the distributed temperature field $T(\mathbf{x}, t)$ with the patches of finite elements methods [34]. If the linear operator (2.107) is assumed to properly describe heat-diffusion effects, and a p-step linear multi-step method of the form:

$$\dot{y}(t_k) + f(y(t_k), t_k) \approx \sum_{j=0}^{p} \alpha_j y(t_{k-j}) + h \sum_{j=0}^{p} \beta_j f(y(t_{k-j}), t_{k-j}) \ ,$$

is supposed to be used for time-discretization purposes, then the stamp associated with the thermal element reads:

	$\boldsymbol{\theta}_k$	$\mathbf{r}_k$	
	$\mathbb{J}_{k,\boldsymbol{\theta}}$	$\mathbb{J}_{k,\mathbf{r}}$	$\mathbf{F}_k$,
	$\mathbb{Q}_{k,\boldsymbol{\theta}}$	$\mathbb{Q}_{k,\mathbf{r}}$	$\mathbf{G}_k$,

where:

$$\mathbf{r}_k = \begin{bmatrix} \mathbf{p}_k \\ \mathbf{T}_k \end{bmatrix} .$$

Assuming $\mathbf{T}$ to have n_T components, and defining:

$$\boldsymbol{\Omega} \in \mathbb{R}^{K+1 \times K} \quad \text{such that} \quad \boldsymbol{\Omega} := \begin{bmatrix} |\Omega_1| & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & |\Omega_K| \\ -|\Omega_1| & \cdots & -|\Omega_K| \end{bmatrix} ,$$

then it is possible to provide an explicit formulation for the entries referring to the first line of the stamp:

$$\mathbb{J}_{k,\boldsymbol{\theta}} \in \mathbb{R}^{K+1 \times K+1} \quad \text{with } \mathbb{J}_{k,\boldsymbol{\theta}} := \begin{bmatrix} 0 \end{bmatrix} ,$$

$$\mathbb{J}_{k,\mathbf{r}} \in \mathbb{R}^{K+1 \times K+n_T} \text{ with } \mathbb{J}_{k,\mathbf{r}} := \begin{bmatrix} \boldsymbol{\Omega} & 0 \end{bmatrix} ,$$

$$\mathbf{F}_k \in \mathbb{R}^{K+1} \qquad \text{with } \mathbf{F}_k := \boldsymbol{\Omega}\ \mathbf{p}_k .$$

The definition of the remaining entries results to be a bit more involved. Assume $\{\phi_j, j = 1, \dots, n_T\}$ to represent the full basis set associated with the space discretized vector $\mathbf{T}$ and define:

$$M_{\boldsymbol{\theta}} \in \mathbb{R}^{K \times K+1} \text{ with } \quad M_{\boldsymbol{\theta}} := \begin{bmatrix} 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & 0 \\ 0 & \cdots & 1 & 0 \end{bmatrix} ,$$

$$M_{\mathbf{T}} \in \mathbb{R}^{K \times n_T} \quad \text{with } [M_{\mathbf{T}}]_{ij} := \frac{1}{|\Omega_i|}(\phi_j, \mathbf{1}_{\Omega_i}) .$$

The space discretized counterpart of the relation linking junction temperatures and distributed temperature field reads then:

$$M_{\boldsymbol{\theta}}\boldsymbol{\theta} - M_{\mathbf{T}}\mathbf{T} = 0 .$$

Denote with:

$$B \in \mathbb{R}^{n_T \times K+1} \text{ with } \quad B := \begin{bmatrix} 0 & \cdots & 0 & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & b_{n_T} \end{bmatrix} ,$$

$$P \in \mathbb{R}^{n_T \times K} \quad \text{with } [P]_{ij} := (\mathbf{1}_{\Omega_j}, \phi_i) ,$$

the matrices accounting for the PDE boundary conditions and heat generation terms, respectively. Notice that only the last column of B has non-zero entries, as boundary conditions depend only on the environment temperature. Assume finally A and C to be the stiffness and mass matrix stemming from patches of finite element method (for more insight on the construction of these matrices the interested reader is referred to [20, Chapter 4]). The space discretized formulation of the heat-diffusion equation reads then:

$$C\dot{\mathbf{T}} + A\mathbf{T} + P\mathbf{p} + B\boldsymbol{\theta} = 0 .$$

Applying the linear multi-step time discretization introduced before it is possible to write the Jacobian contributions as:

$$\mathbb{Q}_{k,\boldsymbol{\theta}} \in \mathbb{R}^{K+n_T \times K+1} \quad \text{with } \mathbb{Q}_{k,\boldsymbol{\theta}} := \begin{bmatrix} M_{\boldsymbol{\theta}} \\ h\beta_0 B \end{bmatrix},$$

$$\mathbb{Q}_{k,\mathbf{r}} \in \mathbb{R}^{K+n_T \times K+n_T} \text{ with } \mathbb{Q}_{k,\mathbf{r}} := \begin{bmatrix} 0 & M_{\mathbf{T}} \\ h\beta_0 P & (\alpha_0 C + h\beta_0 A) \end{bmatrix},$$

while defining:

$$\mathbf{g}_k = \sum_{j=1}^{p} \alpha_j C \mathbf{T}_{k-j} + h \sum_{j=1}^{p} \beta_j \left(A\mathbf{T}_{k-j} + P\mathbf{p}_{k-j} + B\boldsymbol{\theta}_{k-j} \right),$$

gives the following expression for the residual $\mathbf{G}_k \in \mathbb{R}^{K+n_T}$:

$$\mathbf{G}_k = - \begin{bmatrix} 0 \\ h\beta_0 B\boldsymbol{\theta}_k + h\beta_0 P\mathbf{p}_k + (\alpha_0 C + h\beta_0 A)\mathbf{T}_k + \mathbf{g}_k \end{bmatrix}.$$

2.2.2.4 Analysis of the Coupled System

To conclude this section the existence and uniqueness of a solution to the whole electro-thermal system is discussed. This result is of major importance to show that under non-restrictive assumptions the extended electro-thermal netlist introduced in Fig. 2.1 enjoys the same smoothness of the original electrical netlist, that is here formalized as:

$$\begin{aligned} A_C \frac{d\mathbf{q}}{dt} + A_R \mathbf{r}(A_R^T \mathbf{e}, \boldsymbol{\theta}) + A_L \mathbf{i}_L + A_V \mathbf{i}_V + A_I \mathbf{i}(A_C^T \mathbf{e}, \boldsymbol{\theta}) &= 0, \\ \frac{d\boldsymbol{\phi}}{dt} - A_L^T \mathbf{e} &= 0, \\ A_V^T \mathbf{e} - \mathbf{v}(t) &= 0, \\ \mathbf{q} - \mathbf{q}_C(A_C^T \mathbf{e}) &= 0, \\ \boldsymbol{\phi} - \boldsymbol{\phi}_L(\mathbf{i}_L) &= 0. \end{aligned} \tag{2.111}$$

Notice that an additional dependence on junction temperatures is assumed for resistors and controlled current sources. The electrical part has then to be complemented by the balance of Joule power at the thermal network nodes:

$$|\Omega_k| p_k - W_k(\boldsymbol{\theta}, \mathbf{e}) = 0 \text{ for } k = 1, \ldots, K, \tag{2.112}$$

by the thermal element interface conditions:

$$|\Omega_k|\theta_k - (T, \mathbf{1}_{\Omega_k}) = 0 \text{ for } k = 1, \dots, K, \tag{2.113}$$

and by the PDE describing heat diffusion:

$$\frac{d}{dt}(T, v) + a(T, v) + \hat{\alpha}(T, v)_{\partial\Omega} - \sum_{k=1}^{K} p_k(\mathbf{1}_{\Omega_k}, v) - \hat{\alpha}(g_k, v)_{\partial\Omega} = 0 \quad \forall v \in \mathbb{H}^1(\Omega). \tag{2.114}$$

The electrical part (2.111) is supposed in the following to be index-1 for any given $\boldsymbol{\theta} \in \mathbb{C}^0[0, \mathrm{t}_1]$. Defining Q_C to be the orthogonal projector onto the kernel of A_C^T and P_C to be its complement, then sufficient conditions to fulfill the index-1 requirement are [24]:

1. $\ker(A_C, A_R, A_V)^T = \{0\}$, $\ker Q_C^T A_V = \{0\}$,
2. $\mathbf{i}(A_C^T \mathbf{e}, \boldsymbol{\theta})$ uniformly continuous in $\boldsymbol{\theta}$ and Lipschitz continuous in $A_C^T \mathbf{e}$,
3. $\mathbf{V}(\cdot)$ continuous,
4. $\boldsymbol{\phi}_L(\cdot)$ and $\mathbf{q}_C(\cdot)$ differentiable functions of their arguments,
5. $\dfrac{\partial \mathbf{q}_C(A_C^T \mathbf{e})}{\partial(A_C^T \mathbf{e})}$, $\dfrac{\partial \boldsymbol{\phi}_L(\mathbf{i}_L)}{\partial(\mathbf{i}_L)}$ positive definite,
6. $\mathbf{r}(A_R^T \mathbf{e}, \boldsymbol{\theta})$ uniformly continuous in $\boldsymbol{\theta}$ and differentiable in $A_R^T \mathbf{e}$,
7. $\dfrac{\partial \mathbf{r}(A_R^T \mathbf{e}, \boldsymbol{\theta})}{\partial(A_R^T \mathbf{e})}$ positive definite and uniformly continuous in $\boldsymbol{\theta}$.

Under these assumptions the existence and uniqueness of a global solution to an initial value problem with consistent initial conditions on $[0, \mathrm{t}_1]$ follows from standard results [35, Theorem 15]. Furthermore, for each component of the solution in the time interval $[0, \mathrm{t}_1]$ a bound of the form:

$$|x(t)| \leq |d_A(\boldsymbol{\theta}(t))| + \int_0^t |d_D(\boldsymbol{\theta}(\tau))| d\tau \ , \tag{2.115}$$

holds, where $d_A(\cdot)$ and $d_D(\cdot)$ are continuous functions. Notice that the form of (2.115) is due to the index-1 condition, thanks to which the time-derivatives of $\boldsymbol{\theta}(t)$ do not appear in the bound. In this case also the following a-priori bound, uniformly in $\boldsymbol{\theta}$, can be shown to hold:

$$|x(t)| \leq \max_{\mathscr{G}} |d_A(\boldsymbol{\theta})| + |t| \max_{\mathscr{G}} |d_D(\boldsymbol{\theta})| \ , \tag{2.116}$$

where $\mathscr{G}$ is a closed set, such that:

$$\mathscr{F} := \left\{ \mathbf{s} \in \mathbb{R}^K : |\mathbf{s}| \leq \max_{t \in [0, \mathrm{t}_1]} |\boldsymbol{\theta}(t)| \right\} \subseteq \mathscr{G} \subseteq \mathbb{R}^K \ . \tag{2.117}$$

The assumptions made on the thermal part of the system are:

1. $g(\mathbf{x}, t) \in \mathbb{C}^0\left([0, \mathrm{t}_1], \mathbb{L}^2(\partial\Omega)\right)$,
2. $W_k(\cdot, \cdot)$ continuous function of its arguments ($k = 1, \ldots, K$),

To provide system (2.111)–(2.114) with consistent initial conditions it is possible to prescribe arbitrarily $T(\mathbf{x}, 0) := T_0(\mathbf{x}) \in \mathbb{L}^2(\Omega)$, $P_C\mathbf{e}(0)$ and $\mathbf{i}_L(0)$. Then $\boldsymbol{\theta}(0)$ is obtained from (2.113), $Q_C\mathbf{e}(0)$, $\mathbf{i}_V(0)$, $\boldsymbol{\phi}(0)$, $\mathbf{q}(0)$ are computed from the algebraic constraints of (2.111) once $\boldsymbol{\theta}(0)$ is known, and $\mathbf{p}(0)$ is finally determined from (2.112).

The existence and uniqueness of a solution to (2.111)–(2.114) in a given time interval $t \in [0, \mathrm{t}_1]$ is investigated in the next:

Theorem 2.2 *Consider system (2.111)–(2.114) with the further hypothesis that:*

1. There exist $C_\mathbf{W} > 0$ such that $|W_k(\boldsymbol{\theta}, \mathbf{e})| \leq C_\mathbf{W}$ for $k = 1, \ldots, K$.

Suppose furthermore that the assumptions outlined in the previous paragraphs on the electrical and thermal part of the network are fulfilled. Then, given consistent initial conditions, there exist a unique solution to an initial value problem on a given time interval $[0, \mathrm{t}_1]$ and:

1. *$P_C\mathbf{e}$, $\mathbf{i}_L$, $\mathbf{q}$ and $\boldsymbol{\phi}$ are differentiable,*
2. *$Q_C\mathbf{e}$, $\mathbf{i}_V$, $\boldsymbol{\theta}$ and $\mathbf{p}$ are continuous,*
3. *The regularity of the PDE solution is at least:*

$$T \in \mathbb{L}^2\left((0, \mathrm{t}_1), \mathbb{H}^1(\Omega)\right) \cap \mathbb{C}^0\left([0, \mathrm{t}_1], \mathbb{L}^2(\Omega)\right) ,$$

while:

$$\frac{\partial T}{\partial t} \in \mathbb{L}^2\left((0, \mathrm{t}_1), \mathbb{H}^{-1}(\Omega)\right) ,$$

4. *The energy estimate:*

$$\|T(\mathbf{x}, t)\|^2_{\mathbb{L}^2(\Omega)} + \eta \int_0^t \|T(\mathbf{x}, \tau)\|^2_{\mathbb{H}^1(\Omega)} \, d\tau \leq \|T_0(\mathbf{x})\|^2_{\mathbb{L}^2(\Omega)} + \frac{1}{\eta} \int_0^t S^2 d\tau ,$$

holds for each $t \in [0, \mathrm{t}_1]$ where:

$$S = S(C_\mathbf{W}, \hat{\alpha}, \Omega_k, g) := C_\mathbf{W} \sum_{k=1}^{K} \sqrt{|\Omega_k|} + \hat{\alpha} \, \|g(t)\|_{\mathbb{L}^2(\partial\Omega)} .$$

Proof In the following the so-called Faedo-Galerkin method is exploited to construct a sequence of DAE systems that approximate the PDAE system (2.111)–(2.114). The line followed stems directly from the one usually employed to prove the well posedness of parabolic PDEs casted in a weak formulation (see [52, Chapter 11, Theorem 11.1.1]).

That being said, since $\mathbb{H}^1(\Omega)$ is a separable Hilbert space it admits a complete orthonormal basis $\{\phi_j\}_{j\geq 1}$. Define then:

$$V^N := \text{span}\{\phi_1, \ldots, \phi_N\} \ .$$

Substitute the PDE appearing in (2.111)–(2.114) with the approximate problem:

$$\frac{d}{dt}(T^N, v) + a(T^N, v) + \hat{\alpha}(T^N, v)_{\partial\Omega} - \sum_{k=1}^{K} p_k^N (\mathbf{1}_{\Omega_k}, v) - \hat{\alpha}(g, v)_{\partial\Omega} = 0 \qquad (2.118)$$

for all $v \in V^N$, where $N \geq K$ in order to fulfill the constraints imposed by (2.113). Writing:

$$T^N(\mathbf{x}, t) := \sum_{s=1}^{N} c_s^N(t)\phi_s(\mathbf{x}) \ , \qquad (2.119)$$

then (2.118) results to be equivalent to:

$$M\frac{d\mathbf{c}^N}{dt} + A\mathbf{c}^N - B\mathbf{p}^N - \mathbf{F}^N(t) = 0 \ . \qquad (2.120)$$

where the stiffness and mass matrices are defined as:

$$M \in \mathbb{R}^{N\times N} \text{ with } \left[M_{ij}\right] := [(\phi_i, \phi_j)] \ ,$$

$$A \in \mathbb{R}^{N\times N} \text{ with } \ [A_{ij}] := [a(\phi_j, \phi_i) + \hat{\alpha}(\phi_j, \phi_i)_{\partial\Omega}] \ ,$$

$$B \in \mathbb{R}^{N\times K} \text{ with } \ [B_{ij}] := [(\mathbf{1}_{\Omega_k}, \phi_i)] \ ,$$

while the known vector $\mathbf{F}^N$ reads:

$$\mathbf{F}^N \in \left[\mathbb{C}^0[0, \mathrm{t}_1]\right]^N \text{ with } [F_i^N] := [\hat{\alpha}(g, \phi_i)_{\partial\Omega}] \ .$$

Finally the unknown vectors in (2.120) are:

$$\mathbf{p}^N(t) := [p_1^N(t), \ldots, p_K^N(t)]^T \ ,$$

$$\mathbf{c}^N(t) := [c_1^N(t), \ldots, c_N^N(t)]^T \ .$$

Similarly it is possible to substitute (2.119) in (2.113) and obtain the equivalent system:

$$\boldsymbol{\Omega}\,\boldsymbol{\theta}^N - B^T\mathbf{c}^N = 0 \ , \qquad (2.121)$$

where:

$$\boldsymbol{\Omega} \in \mathbb{R}^{K\times K} \text{ with } \boldsymbol{\Omega} := \operatorname{diag}(|\Omega_1|, \ldots, |\Omega_K|) ,$$

and:

$$\boldsymbol{\theta}^N(t) := [\theta_1^N(t), \ldots, \theta_K^N(t)]^T .$$

Reformulating (2.112) in matrix notation:

$$\boldsymbol{\Omega}\mathbf{p}^N - \mathbf{W}(\boldsymbol{\theta}^N, \mathbf{e}^N) = 0 , \tag{2.122}$$

with:

$$\boldsymbol{W}(\boldsymbol{\theta}^N, \mathbf{e}^N) := [W_1(\boldsymbol{\theta}^N, \mathbf{e}^N), \ldots, W_K(\boldsymbol{\theta}^N, \mathbf{e}^N)]^T ,$$

it is possible to write the DAE system approximating (2.111)–(2.114) as:

$$\begin{aligned}
A_C\frac{d\mathbf{q}^N}{dt} + A_R\mathbf{r}(A_R^T\mathbf{e}^N, \boldsymbol{\theta}^N) + A_L\mathbf{i}_L^N + A_V\mathbf{i}_V^N + A_I\mathbf{i}(A_C^T\mathbf{e}^N, \boldsymbol{\theta}^N) &= 0 ,\\
\frac{d\boldsymbol{\phi}^N}{dt} - A_L^T\mathbf{e}^N &= 0 ,\\
A_V^T\mathbf{e}^N - \mathbf{V}(t) &= 0 ,\\
\mathbf{q}^N - \mathbf{q}_C(A_C^T\mathbf{e}^N) &= 0 ,\\
\boldsymbol{\phi}^N - \boldsymbol{\phi}_L(\mathbf{i}_L^N) &= 0 ,\\
\boldsymbol{\Omega}\mathbf{p}^N - \mathbf{W}(\boldsymbol{\theta}^N, \mathbf{e}^N) &= 0 ,\\
\boldsymbol{\Omega}\boldsymbol{\theta}^N - B^T\mathbf{c}^N &= 0 ,\\
M\frac{d\mathbf{c}^N}{dt} + A\mathbf{c}^N - B\mathbf{p}^N - \mathbf{F}^N(t) &= 0 .
\end{aligned} \tag{2.123}$$

Notice that M can be inverted as it is positive definite. Thus (2.120) defines an explicit differential equation for the variable $\mathbf{c}^N$:

$$\frac{d\mathbf{c}^N}{dt} = -M^{-1}\left[A\mathbf{c}^N - B\mathbf{p}^N - \mathbf{F}^N(t)\right] . \tag{2.124}$$

From (2.121) it holds:

$$\boldsymbol{\theta}^N = \boldsymbol{\Omega}^{-1}B^T\mathbf{c}^N , \tag{2.125}$$

due to the regularity of $\boldsymbol{\Omega}$. Differentiating (2.125) and taking into account (2.124) the following explicit differential equation is obtained for $\boldsymbol{\theta}^N$:

$$\frac{d\boldsymbol{\theta}^N}{dt} = \boldsymbol{\Omega}^{-1} B^T \frac{d\mathbf{c}^N}{dt} = -\boldsymbol{\Omega}^{-1} B^T M^{-1} \left[A\mathbf{c}^N - B\mathbf{p}^N - \mathbf{F}^N(t)\right] .$$

Substituting (2.125) into (2.111) reads:

$$\begin{aligned} A_C \frac{d\mathbf{q}}{dt} + A_R \hat{\mathbf{r}}(A_R^T \mathbf{e}, \mathbf{c}^N) + A_L \mathbf{i}_L + A_V \mathbf{i}_V + A_I \hat{\mathbf{i}}(A_C^T \mathbf{e}, \mathbf{c}^N) &= 0, \\ \frac{d\boldsymbol{\phi}}{dt} - A_L^T \mathbf{e} &= 0, \\ A_V^T \mathbf{e} - \mathbf{V}(t) &= 0, \\ \mathbf{q} - \mathbf{q}_C(A_C^T \mathbf{e}) &= 0, \\ \boldsymbol{\phi} - \boldsymbol{\phi}_L(\mathbf{i}_L) &= 0, \end{aligned} \tag{2.126}$$

where:

$$\begin{aligned} \hat{\mathbf{r}}(A_R^T \mathbf{e}, \mathbf{c}^N) &:= \mathbf{r}(A_R^T \mathbf{e}, \boldsymbol{\Omega}^{-1} B^T \mathbf{c}^N) , \\ \hat{\mathbf{i}}(A_C^T \mathbf{e}, \mathbf{c}^N) &:= \mathbf{i}(A_C^T \mathbf{e}, \boldsymbol{\Omega}^{-1} B^T \mathbf{c}^N) . \end{aligned}$$

The assumptions on the electrical part of the system ensure that only one differentiation of (2.126) is needed to derive, through appropriate algebraic manipulations, a set of explicit differential equations for the variables $\mathbf{e}$, $\mathbf{q}$, $\boldsymbol{\phi}$, $\mathbf{i}_L$ and $\mathbf{i}_V$. Finally from (2.122) it stems:

$$\mathbf{p}^N = \boldsymbol{\Omega}^{-1} \mathbf{W}(\boldsymbol{\theta}^N, \mathbf{e}^N) .$$

Even here only one differentiation is necessary to derive an explicit differential equation for $\mathbf{p}^N$. The index of system (2.123) results then to be one.

Defining the orthogonal projection:

$$P_N : \mathbb{L}^2(\Omega) \to V^N ,$$

it is possible to derive a set of consistent initial conditions for (2.123). In fact, it just suffices to define the initial conditions for the approximate problem (2.118) as:

$$T_0^N := P_N(T_0) , \tag{2.127}$$

and proceed as done in the original PDAE system. Notice that the initial condition for system (2.120) equivalent to (2.127) is given by the solution of the linear system:

$$(M\mathbf{c}^N(0))_j = (T_0, \phi_j) \text{ for } j = 1, \ldots, N .$$

As consistent initial conditions have been obtained, then (2.123) admits a unique global solution [35].

To proceed with the Faedo-Galerkin method it is necessary at this point to recover, for all the variables in (2.123), upper bounds in $\mathbb{L}^2(0, \mathrm{t}_1)$ that are independent of N. These bounds will be employed afterwards to pass to the weak-limit $N \to \infty$ and determine then a solution to the initial PDAE system. Due to the hypothesis made on the boundedness of $|W_k(\cdot, \cdot)|$ it is convenient to start from the thermal part of the network, noticing that:

$$p_k^N \in \mathbb{C}^0[0, \mathrm{t}_1] \subset \mathbb{L}^2(0, \mathrm{t}_1) \;\; k = 1, \ldots, K,$$

$$c_k^N \in \mathbb{C}^1[0, \mathrm{t}_1] \subset \mathbb{H}^1(0, \mathrm{t}_1) \;\; k = 1, \ldots, K,$$

hold, from which it follows naturally:

$$T^N \in \mathbb{H}^1\left((0, \mathrm{t}_1), \mathbb{H}^1(\Omega)\right) .$$

Choosing T^N as a test function in (2.118) gives:

$$\left(\frac{d}{dt}T^N, T^N\right) + a(T^N, T^N) + \hat{\alpha}(T^N, T^N)_{\partial\Omega} \tag{2.128}$$

$$= \sum_{k=1}^{K} p_k^N(\mathbf{1}_{\Omega_k}, T^N) + \hat{\alpha}(g, T^N)_{\partial\Omega} . \tag{2.129}$$

Exploiting the coercivity of the bilinear form it is possible to obtain:

$$\frac{1}{2}\frac{d}{dt}\left\|T^N\right\|^2_{\mathbb{L}^2(\Omega)} + \eta\left\|T^N\right\|^2_{\mathbb{H}^1(\Omega)} \tag{2.130}$$

$$\leq \left(\frac{d}{dt}T^N, T^N\right) + a(T^N, T^N) + \hat{\alpha}(T^N, T^N)_{\partial\Omega} , \tag{2.131}$$

while from the continuity of the right-hand side in (2.128) and the hypothesis on the boundedness of $|W_k(\cdot, \cdot)|$ ($k = 1, \ldots, K$) follows:

$$\begin{aligned} &\sum_{k=1}^{K} p_k^N(\mathbf{1}_{\Omega_k}, T^N) + \hat{\alpha}(g, T^N)_{\partial\Omega} \\ &\leq \left(C_{\mathrm{W}}\sum_{k=1}^{K}\sqrt{|\Omega_k|} + \hat{\alpha}\left\|g(t)\right\|_{\mathbb{L}^2(\partial\Omega)}\right)\left\|T^N(t)\right\|_{\mathbb{L}^2(\Omega)} . \end{aligned} \tag{2.132}$$

Recapitulating the definition of $S(C_{\mathbf{W}}, \hat{\alpha}, \Omega_k, g) := C_{\mathbf{W}} \sum_{k=1}^{K} \sqrt{|\Omega_k|} + \hat{\alpha} \, \|g(t)\|_{\mathbb{L}^2(\partial\Omega)}$, and combining (2.130) with (2.132) it is possible to obtain:

$$\frac{1}{2}\frac{d}{dt} \left\| T^N(t) \right\|^2_{\mathbb{L}^2(\Omega)} + \eta \left\| T^N(t) \right\|^2_{\mathbb{H}^1(\Omega)} \le S(C_{\mathbf{W}}, \hat{\alpha}, \Omega_k, g) \left\| T^N(t) \right\|_{\mathbb{L}^2(\Omega)} .$$

Integrating over $(0, t)$ with $t \in (0, \mathrm{t}_1)$, employing Young's inequality and taking into account that:

$$\|T_0^N\|_{\mathbb{L}^2(\Omega)} \le \|T_0\|_{\mathbb{L}^2(\Omega)} ,$$

as T_0^N is a projection of T_0 onto a finite dimensional space, it follows then:

$$\left\| T^N(t) \right\|^2_{\mathbb{L}^2(\Omega)} + \eta \int_0^t \left\| T^N(\tau) \right\|^2_{\mathbb{H}^1(\Omega)} d\tau \le \|T_0\|^2_{\mathbb{L}^2(\Omega)} + \frac{1}{\eta} \int_0^t S^2 d\tau . \tag{2.133}$$

The sequence T^N is thus bounded in $\mathbb{L}^2\left((0, \mathrm{t}_1), \mathbb{H}^1(\Omega)\right) \cap \mathbb{L}^\infty\left((0, \mathrm{t}_1), \mathbb{L}^2(\Omega)\right)$ and from (2.112) it is trivial to infer that also $\mathbf{p}^N$ is bounded in the $\mathbb{L}^2(0, \mathrm{t}_1)$ sense. From (2.113) it is possible to obtain, after some algebra:

$$|\theta_k^N(t)|^2 \le \frac{1}{|\Omega_k|^2} \|T^N(t)\|^2_{\mathbb{L}^2(\Omega)} \quad k = 1, \ldots, K,$$

and derive an upper bound in $\mathbb{L}^2(0, \mathrm{t}_1)$ for $\boldsymbol{\theta}^N$ by means of (2.133):

$$|\theta_k^N(t)|^2 \le \frac{1}{|\Omega_k|^2} \Big[\|T_0\|^2_{\mathbb{L}^2(\Omega)} + \frac{1}{\eta} \int_0^{\mathrm{t}_1} S^2 d\tau \Big] \quad k = 1, \ldots, K.$$

Also this bound does not depend on N. It is now possible to define:

$$C_{\boldsymbol{\theta}} := \max_{k=1,\ldots,K} \Big(\frac{1}{|\Omega_k|^2} \Big[\|T_0\|^2_{\mathbb{L}^2(\Omega)} + \frac{1}{\eta} \int_0^{\mathrm{t}_1} S^2 d\tau \Big] \Big) ,$$

and then:

$$\mathscr{G} := \Big\{ \mathbf{s} \in \mathbb{R}^K : |s| \le \sqrt{C_{\boldsymbol{\theta}}} \Big\} .$$

As $\mathscr{G}$ does not depend on N and fulfills condition (2.117) then the bound on the variables of the electrical part is derived from (2.116). Notice that this is possible in our framework due to the index-1 hypothesis made on (2.111). Finally, due to the continuity of the non-linear functions in (2.123) also the terms:

$$\begin{aligned}
\mathbf{r}^N &:= \mathbf{r}(A_R^T \mathbf{e}^N, \boldsymbol{\theta}^N) \ , \quad \mathbf{i}^N := \mathbf{i}(A_C^T \mathbf{e}^N, \boldsymbol{\theta}^N) \ , \\
\mathbf{q}_C^N &:= \mathbf{q}_C(A_C^T \mathbf{e}^N) \quad , \boldsymbol{\phi}_L^N := \boldsymbol{\phi}_L(\mathbf{i}_L^N) \qquad , \\
\mathbf{W}^N &:= \mathbf{W}(\boldsymbol{\theta}^N, \mathbf{e}^N) \quad ,
\end{aligned}$$

are bounded in the $\mathbb{L}^2(0, \mathrm{t}_1)$ norm by a constant that is independent of N. At this point upper bounds for every entity in (2.123) have been determined. Hence it is possible to select a subsequence (still denoted with the N super-script) in which (see e.g. [42]):

- T^N converges in the weak* topology of $\mathbb{L}^\infty\big((0, \mathrm{t}_1), \mathbb{L}^2(\Omega)\big)$,
- T^N converges weakly in $\mathbb{L}^2\big((0, \mathrm{t}_1), \mathbb{H}^1(\Omega)\big)$,
- $\mathbf{e}^N, \mathbf{i}_L^N, \mathbf{q}^N, \boldsymbol{\phi}^N, \mathbf{i}_V^N, \mathbf{p}^N$ and $\boldsymbol{\theta}^N$ converge weakly in the $\mathbb{L}^2(0, \mathrm{t}_1)$ sense,
- $\mathbf{r}^N, \mathbf{i}^N, \mathbf{q}_C^N, \boldsymbol{\phi}_L^N$ and $\mathbf{W}^N$ converge weakly in the $\mathbb{L}^2(0, \mathrm{t}_1)$ sense.

Anyhow, to exploit weak convergence properties in order to construct a solution to the original PDAE system it is still necessary to prove that:

$$\begin{aligned}
\mathbf{r}^N &\rightharpoonup \mathbf{r}(A_R^T \mathbf{e}, \boldsymbol{\theta}) \ , \quad \mathbf{i}^N \rightharpoonup \mathbf{i}(A_C^T \mathbf{e}, \boldsymbol{\theta}) \ , \\
\mathbf{q}_C^N &\rightharpoonup \mathbf{q}_C(A_C^T \mathbf{e}) \ , \boldsymbol{\phi}_L^N \rightharpoonup \boldsymbol{\phi}_L(\mathbf{i}_L) \qquad , \\
\mathbf{W}^N &\rightharpoonup \mathbf{W}(\boldsymbol{\theta}, \mathbf{e}) \quad ,
\end{aligned}$$

when:

$$\mathbf{e}^N \rightharpoonup \mathbf{e} \ , \mathbf{i}_L^N \rightharpoonup \mathbf{i}_L \ , \boldsymbol{\theta}^N \rightharpoonup \boldsymbol{\theta} \ .$$

This will be shown in the following taking advantage of regularity results that hold for the PDE part of this system. Indeed it will turn out that the convergence of the DAE part of (2.111)–(2.114) is to be intended at least pointwise.

Let us start then multiplying the first term at the left hand side in (2.118) by:

$$\Psi \in \mathbb{C}^1([0, \mathrm{t}_1]) \ , \ \Psi(\mathrm{t}_1) = 0 \ ,$$

and integrating by parts ($j = 1, \dots, N$):

$$\int_0^{\mathrm{t}_1} \left(\frac{dT^N}{dt}(\tau), \phi_j \right) \Psi(\tau) d\tau = - \int_0^{\mathrm{t}_1} \big(T^N(\tau), \phi_j \big) \frac{d\Psi}{dt}(\tau) d\tau - (T_0^N, \phi_j)\Psi(0) \ .$$

Passing to the limit in (2.118), choosing an arbitrary $N_0 \geq K$ and recalling that T_0^N converges in $\mathbb{L}^2(\Omega)$ to T_0 while p_k^N converges in $\mathbb{L}^2(0, \mathrm{t}_1)$ to p_k, it is finally obtained:

$$\begin{aligned}
&-\int_0^{\mathrm{t}_1}\big(T(\tau),\phi_j\big)\frac{d\Psi}{dt}(\tau)d\tau-(T_0,\phi_j)\Psi(0)+\int_0^{\mathrm{t}_1}a(T,\phi_j)\Psi(\tau)d\tau\\
&+\int_0^{\mathrm{t}_1}\hat{\alpha}(T(\tau),\phi_j)_{\partial\Omega}\Psi(\tau)d\tau=\int_0^{\mathrm{t}_1}\sum_{k=1}^{K}p_k(\mathbf{1}_{\Omega_k},\phi_j)\Psi(\tau)d\tau\\
&\quad+\int_0^{\mathrm{t}_1}\hat{\alpha}(g,\phi_j)_{\partial\Omega}\Psi(\tau)d\tau\qquad j=1,\dots,N_0.
\end{aligned}\tag{2.134}$$

Since the linear combinations of ϕ_j are dense in $\mathbb{H}^1(\Omega)$, then (2.134) can be written equivalently testing on each $v\in\mathbb{H}^1(\Omega)$. Thus:

$$T(\mathbf{x},t)\in\mathbb{L}^2\left((0,\mathrm{t}_1),\mathbb{H}^1(\Omega)\right)\cap\mathbb{L}^\infty\left((0,\mathrm{t}_1),\mathbb{L}^2(\Omega)\right)\ ,\tag{2.135}$$

fulfills (2.114) with p_k $(k=1,\dots,K)$ as source terms. From (2.135) it follows also:

$$T(\mathbf{x},t)\in\mathbb{L}^2\left((0,\mathrm{t}_1),\mathbb{H}^1(\Omega)\right)\cap\mathbb{H}^1\left((0,\mathrm{t}_1),\mathbb{H}^{-1}(\Omega)\right)\ ,\tag{2.136}$$

and using the arguments in [52, Chapter 11, p.369] and [43, p.23]:

$$T\in\mathbb{C}^0\left([0,\mathrm{t}_1],\mathbb{L}^2(\Omega)\right)\quad,\quad\frac{\partial T}{\partial t}\in\mathbb{L}^2\left((0,\mathrm{t}_1),\mathbb{H}^{-1}(\Omega)\right)\ .$$

Define:

$$\Delta T^N(t):=T^N(t)-T(t)\quad,\quad\Delta p_k^N(t):=p_k^N(t)-p_k(t)\ .$$

Subtracting (2.114) from (2.118) and choosing $\Delta T^N(t)$ as a test function reads:

$$\frac{1}{2}\frac{d}{dt}\|\Delta T^N\|^2_{\mathbb{L}^2(\Omega)}+a(\Delta T^N,\Delta T^N)+\hat{\alpha}\|\Delta T^N\|^2_{\mathbb{L}^2(\partial\Omega)}=\sum_{k=1}^{K}\Delta p_k^N(\mathbf{1}_{\Omega_k},\Delta T^N)\ .$$

Integrating over $(0,t)$ and exploiting the coercivity of the bilinear form it is then possible to obtain the following inequality:

$$\left\|\Delta T^N(t)\right\|^2_{\mathbb{L}^2(\Omega)}\le\Delta K^N(t)\xrightarrow{N\to\infty}0\ ,\tag{2.137}$$

with:

$$\Delta K^N(t):=\left\|\Delta T^N(0)\right\|^2_{\mathbb{L}^2(\Omega)}+2\sum_{k=1}^{K}\Big[\int_0^t\Delta p_k^N(\tau)(\mathbf{1}_{\Omega_k},\Delta T^N(\tau))d\tau\Big]\ .$$

As both sides of (2.137) are continuous, this inequality holds also in the form:

$$\max_{t\in[0,\mathrm{t}_1]} \left\|\Delta T^N(t)\right\|^2_{\mathbb{L}^2(\Omega)} \leq \max_{t\in[0,\mathrm{t}_1]} \Delta K^N(t) \xrightarrow{N\to\infty} 0\,.$$

Introducing $\Delta\theta_k^N := \theta_k^N(t) - \theta_k(t)$ and noticing that:

$$\max_{t\in[0,\mathrm{t}_1]} |\Delta\theta_k^N(t)|^2 \leq \frac{1}{|\Omega_k|^2} \max_{t\in[0,\mathrm{t}_1]} \left\|\Delta T^N(t)\right\|^2_{\mathbb{L}^2(\Omega)} \qquad k = 1,\ldots,K,$$

it follows that the convergence of $\boldsymbol{\theta}^N$ to $\boldsymbol{\theta}$ is not only weak, but uniform. Then, due to the stability properties of (2.111) the electrical variables also converge to their limit uniformly and not only weakly. In particular it can be inferred that:

- $P_C\mathbf{e}$, $\mathbf{i}_L$, $\mathbf{q}$ and $\boldsymbol{\phi}$ are differentiable,
- $Q_C\mathbf{e}$ and $\mathbf{i}_V$ are continuous.

As at this point $\mathbf{e}$ and $\boldsymbol{\theta}$ are known to be continuous, then it follows that $\mathbf{W}^N$ converges to $\mathbf{W}$ pointwise and thus $\mathbf{p}$ is also continuous.

Finally it remains to show that $T(\mathbf{x}, 0) = T_0(\mathbf{x})$ in order to prove that the constructed solution actually solves the initial value problem prescribed in the beginning. Multiplying (2.114) by:

$$\Psi \in \mathbb{C}^1([0,\mathrm{t}_1])\,,\ \Psi(\mathrm{t}_1) = 0\,,$$

and integrating by parts it follows:

$$-\int_0^{\mathrm{t}_1} (T(\tau), v)\,\frac{d\Psi}{dt}(\tau)d\tau \,-\, (T(0), v)\Psi(0) + \int_0^{\mathrm{t}_1} a(T, v)\Psi(\tau)d\tau$$
$$+\int_0^{\mathrm{t}_1} \hat{\alpha}(T(\tau), v)_{\partial\Omega}\Psi(\tau)d\tau = \int_0^{\mathrm{t}_1} \sum_{k=1}^{K} p_k(\mathbf{1}_{\Omega_k}, v)\Psi(\tau)d\tau$$
$$+\int_0^{\mathrm{t}_1} \hat{\alpha}(g, v)_{\partial\Omega}\Psi(\tau)d\tau \qquad \forall v \in \mathbb{H}^1(\Omega),$$

thus, taking $\Psi(0) = 1$:

$$(T(0) - T_0, v) = 0 \quad \forall v \in \mathbb{H}^1(\Omega)\,.$$

This implies $T(\mathbf{x}, 0) = T_0(\mathbf{x})$, and proves the existence and uniqueness of a solution to a prescribed initial value problem for system (2.111)–(2.114).

2.2.3 Multiphysics Modeling via Maxwell's Equations

The mathematical model of circuit analysis is given by element relations connected by Kirchhoff's laws, yielding a system of DAEs. Each relation originates from Maxwell's equations, but typically it is simplified to avoid the simulation of PDEs, where it is not necessary. But if an application demands distributed field effects, e.g. eddy currents, those effects need to be reintroduced by a PDE, in which some conducting parts are identified by circuit branches. We consider here two examples, that are especially important in the analysis of magnetoquasistatic fields, the *solid* and *stranded conductor* models. Finally the coupling of the networks DAEs with the (magnetoquasistatic) field PDEs yields a system of PDAEs.

Let us start with the network model of circuits, as introduced in system (2.1). [26], that yields a system of DAEs. We extend the current balance equation by two additional vectors $\mathbf{i}_{\mathrm{sol}} \in \mathbb{R}^{N_{\mathrm{sol}}}$ and $\mathbf{i}_{\mathrm{str}} \in \mathbb{R}^{N_{\mathrm{str}}}$, that describe the unknown currents through N_{sol} solid and N_{str} stranded conductors

$$\begin{aligned}
\mathbf{A}_{\mathrm{C}}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q} + \mathbf{A}_{\mathrm{R}}r(\mathbf{A}_{\mathrm{R}}^{\top}\mathbf{u},t) + \mathbf{A}_{\mathrm{L}}\mathbf{i}_{\mathrm{L}} + \mathbf{A}_{\mathrm{V}}\mathbf{i}_{\mathrm{V}} + \mathbf{A}_{\mathrm{I}}\mathbf{i}(t) + \mathbf{A}_{\mathrm{str}}\mathbf{i}_{\mathrm{str}} + \mathbf{A}_{\mathrm{sol}}\mathbf{i}_{\mathrm{sol}} &= \mathbf{0},\\
\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\phi} - \mathbf{A}_{\mathrm{L}}^{\top}\mathbf{u} = \mathbf{0}, \qquad \mathbf{A}_{\mathrm{V}}^{\top}\mathbf{u} - \mathbf{v}(t) &= \mathbf{0},\\
\mathbf{q} - \mathbf{q}_{\mathrm{C}}(\mathbf{A}_{\mathrm{C}}^{\top}\mathbf{u},t) = \mathbf{0}, \quad \boldsymbol{\phi} - \boldsymbol{\phi}_{\mathrm{L}}(\mathbf{i}_{\mathrm{L}},t) &= \mathbf{0},
\end{aligned}$$

with consistent initial values for the node potentials $\mathbf{u}$, charges $\mathbf{q}$, fluxes $\boldsymbol{\phi}$, and currents $\mathbf{i}_{\mathrm{L}}$, $\mathbf{i}_{\mathrm{V}}$. We will address the whole system in the following more abstractly by the semi-explicit initial-value problem

$$\begin{aligned}
\dot{\mathbf{y}}_1 &= \mathbf{f}_1(\mathbf{y}_1,\mathbf{z}_1,\mathbf{z}_{2b}), \quad \text{with} \quad \mathbf{y}_1(0) = \mathbf{y}_{1,0}\\
\mathbf{0} &= \mathbf{g}_1(\mathbf{y}_1,\mathbf{z}_1,\mathbf{z}_{2b}),
\end{aligned} \tag{2.138}$$

with the unknowns

$$\mathbf{y}_1 := (\mathbf{q},\boldsymbol{\phi})^{\top}, \quad \mathbf{z}_1 := (\mathbf{u},\mathbf{i}_{\mathrm{L}},\mathbf{i}_{\mathrm{V}})^{\top} \quad \text{and} \quad \mathbf{z}_{2b} := (\mathbf{i}_{\mathrm{str}},\mathbf{i}_{\mathrm{sol}})^{\top}$$

We assume that System (2.138) is an index-1 DAE, i.e., $\partial\mathbf{g}_1/\partial\mathbf{z}_1$ nonsingular, which is the case if several topological conditions are fulfilled, [24]. The field PDE will describe a relation between the unknown currents $\mathbf{z}_{2b}$ and the voltage drops

$$\mathbf{v}_{\mathrm{str}} := \mathbf{A}_{\mathrm{str}}^{\top}\mathbf{u} \qquad \text{and} \qquad \mathbf{v}_{\mathrm{sol}} := \mathbf{A}_{\mathrm{sol}}^{\top}\mathbf{u}\ ,$$

that will serve as an external excitement of Maxwell's Equations.

2.2.3.1 Maxwell's Equations

Maxwell's equations can be applied to describe a wide range of electromagnetic devices; in our focus are device parts that are typically embedded in electrical circuits exhibiting significant magnetic effects and dissipation losses, but with a disregardable displacement current. This kind of application is covered by the *magnetoquasistatic* (MQS) subset of Maxwell's Equations, [39], that is given by the partial differential equations

$$\begin{aligned} \nabla \times \boldsymbol{E} &= -\frac{\mathrm{d}\boldsymbol{B}}{\mathrm{d}t}\,, & \nabla \times \boldsymbol{H} &= \boldsymbol{J}\,, \\ \nabla \cdot \boldsymbol{D} &= \boldsymbol{\rho}\,, & \nabla \cdot \boldsymbol{B} &= 0\,, \end{aligned} \tag{2.139a}$$

with algebraic material relations

$$\boldsymbol{J} = \sigma \boldsymbol{E}\,, \qquad \boldsymbol{D} = \varepsilon_0\,\varepsilon_r \boldsymbol{E} = \varepsilon \boldsymbol{E}\,, \qquad \boldsymbol{B} = \mu_0\,\mu_r \boldsymbol{H} = \mu \boldsymbol{H}\,, \tag{2.139b}$$

on a domain Ω and typically with the *flux wall* boundary condition

$$\boldsymbol{B} \cdot \mathbf{n}_\perp = 0 \quad \text{on } \partial\Omega\,, \tag{2.139c}$$

where $\boldsymbol{E} = \boldsymbol{E}(\mathbf{r}, t)$ is the electric field strength, depending on its location in space $\mathbf{r} = (x, y, z)^\top$ and time t, similarly $\boldsymbol{B} = \boldsymbol{B}(\mathbf{r}, t)$ is the magnetic flux density, whose normal component is vanishing at the boundary, since the vector $\mathbf{n}_\perp$ defines here the outer normal at the boundary. $\boldsymbol{H} = \boldsymbol{H}(\mathbf{r}, t)$ denotes the magnetic field strength, $\boldsymbol{D} = \boldsymbol{D}(\mathbf{r}, t)$ the electric flux density, $\boldsymbol{\rho} = \boldsymbol{\rho}(\mathbf{r}, t)$ the electric charge density and $\boldsymbol{J} = \boldsymbol{D}(\mathbf{r}, t)$ the electric current density. The material parameters $\varepsilon = \varepsilon(\mathbf{r})$, $\sigma = \sigma(\mathbf{r})$, $\mu = \mu(\mathbf{r}, \boldsymbol{H})$ are rank-2 tensors describing the permittivity, conductivity and permeability; the first two tensors are assumed constant but the permeability may depend nonlinearly on the field strength. If we neglect furthermore hysteresis, the Jacobian $\partial \boldsymbol{B}/\partial \boldsymbol{H}$ is symmetric positive definite, [38], and we can derive from the second relation of (2.139b) the *HB-characteristic*

$$\boldsymbol{H} = \nu \boldsymbol{B}$$

with the (nonlinear) reluctivity $\nu = \nu(\mathbf{r}, \boldsymbol{B})$ acting as the inverse of the permeability. Now when expressing the magnetic flux and the electric field in terms of the magnetic vector potential $\boldsymbol{A} = \boldsymbol{A}(\mathbf{r}, t)$ and the electric scalar potential $\boldsymbol{\varphi} = \boldsymbol{\varphi}(\mathbf{r}, t)$

$$\boldsymbol{B} = \nabla \times \boldsymbol{A}\,, \qquad \boldsymbol{E} = -\nabla\boldsymbol{\varphi} - \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t}\,, \tag{2.140}$$

Ampère's Law may be equivalently given as the *curl-curl* equation

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) = \boldsymbol{J} \ . \tag{2.141}$$

The curl-curl equation does not determine the potentials uniquely, because the definitions (2.140) are still fulfilled after a gauge transformation. Typically one defines a representant from the class of equivalent potentials as the desired solution by enforcing an additional gauge condition, for example *Coulomb's gauge*

$$\nabla \cdot \boldsymbol{A} = 0 \ , \tag{2.142}$$

which ensures on simply connected domains a unique solution of the problem in the vector potential formulation, [14].

In the 2D case, where a planar model is embedded in an 3D environment both, the magnetic vector potential $\boldsymbol{A}$ and the source current density $\boldsymbol{J}$ exhibit only components in z-direction, which are perpendicular to the planar model in the $x - y$ plane, i.e.,

$$\boldsymbol{A} = \begin{pmatrix} 0 \ 0 \ \boldsymbol{A}_z \end{pmatrix}^\top \qquad \text{and} \qquad \boldsymbol{J} = \begin{pmatrix} 0 \ 0 \ \boldsymbol{J}_z \end{pmatrix}^\top .$$

Thus the potential $\boldsymbol{A}$ fulfills automatically the Coulomb gauge

$$\nabla \cdot \boldsymbol{A} = \frac{\partial \boldsymbol{A}_x}{\partial x} + \frac{\partial \boldsymbol{A}_y}{\partial y} + \frac{\partial \boldsymbol{A}_z}{\partial z} = 0 \ ,$$

since $\boldsymbol{A}_x = \boldsymbol{A}_y = 0$ is trivial and $\boldsymbol{A}_z$ is independent of z and therefore the potential is uniquely defined without enforcing a gauge explicitly; this is in contrast to the 3D case.

2.2.3.2 Conductor Models

In the following the models for the *solid* and *stranded conductor* are derived, whose characteristics are determined by Maxwell's Equations (2.139), but on the other hand allow us to identify parts of the field domain Ω as branches in a circuit using voltages $\mathbf{v}_{\text{sol}}$, $\mathbf{v}_{\text{str}}$ and currents $\mathbf{i}_{\text{sol}}$, $\mathbf{i}_{\text{str}}$, [13]. We denote the corresponding parts of the domain by

$$\Omega_{\text{sol},l} \subset \Omega \qquad \text{and} \qquad \Omega_{\text{str},k} \subset \Omega \qquad \text{for } 1 \leq l \leq N_{\text{sol}},\ 1 \leq k \leq N_{\text{str}}$$

and assume furthermore that they are mutually non-overlapping.

The solid conductor model describes the behavior of a massive bar of conducting material, as shown in Fig. 2.3b. For high frequencies there is a tendency for the current density in the core of those conductor to be smaller than near the surface, [39]. This phenomenon is called *skin effect*. It causes the resistance of the conductor

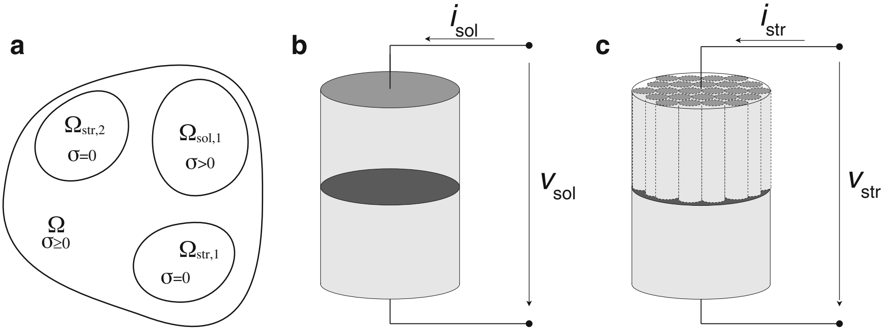


Fig. 2.3 Conductors Models. (**a**) Sketch of a 2D domain with two stranded and one solid conductor, (**b**) solid conductor made of massive conducting material causing eddy currents and (**c**) stranded conductor made of thin strands

to increase with the frequency of the current. A similar phenomenon appears in a solid conductor when localized in the neighborhood of other current carrying conductors. Also then, eddy currents and eddy-current losses appear in the solid conductor. These effects are to be simulated in the following: the solid conductor will serve as the device in a electrical circuit, where skin- and proximity effects are considered.

The voltage drop along each solid conductor is applied as the potential difference between two electrodes, i.e.,

$$-\nabla\boldsymbol{\varphi} = \sum_{l=1}^{N_{\text{sol}}} \boldsymbol{\chi}_{\text{sol},l}\,(\mathbf{v}_{\text{sol}})_l$$

where $\boldsymbol{\chi}_{\text{sol},l}$ is the *potential distribution* function of the l-th solid conductor with $\operatorname{supp}\boldsymbol{\chi}_{\text{sol},l} = \Omega_{\text{sol},l}$. Inserting the voltage drop into Ohm's Law, first equation in (2.139b), and applying it as the only excitement of the curl-curl equation yields (2.141)

$$\sigma\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t} + \nabla\times(\nu\nabla\times\boldsymbol{A}) = \sum_{l=1}^{N_{\text{sol}}} \sigma\boldsymbol{\chi}_{\text{sol},l}\,(\mathbf{v}_{\text{sol}})_l\ . \tag{2.143a}$$

The current through the solid conductor is found by integrating the current density over the electrodes. This is equivalent to integrating the quantity $\boldsymbol{\chi}_{\text{sol}}\cdot\boldsymbol{J}$ over the whole computational domain, i.e.,

$$(\mathbf{i}_{\text{sol}})_l = \int_\Omega \boldsymbol{\chi}_{\text{sol},l}\cdot\boldsymbol{J}\,\mathrm{d}\Omega = (\mathbf{G}_{\text{sol}})_{l,l}\,(\mathbf{v}_{\text{sol}})_l - \int_\Omega \sigma\boldsymbol{\chi}_{\text{sol},l}\cdot\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t}\,\mathrm{d}\Omega\ , \tag{2.143b}$$

for $l = 1, \ldots, N_{\mathrm{sol}}$, where each entry of the positive definite diagonal matrix

$$(\mathbf{G}_{\mathrm{sol}})_{l,l} = \int_{\Omega} \sigma \boldsymbol{\chi}_{\mathrm{sol},l} \cdot \boldsymbol{\chi}_{\mathrm{sol},l} \, \mathrm{d}\Omega \ . \tag{2.143c}$$

corresponds to the lumped DC conductivity of a solid conductor.

In contrast to the solid conductor, the stranded conductor is not built of a single solid material, but consists of thin individual strands wound to form a coil, as depicted in Fig. 2.3c. Each strand does not exhibit significant eddy currents because of its cross section, which is assumed to be substantially smaller than the skin depth related to the frequencies occurring in the model, hence the conductivity, which introduces eddy current effects in the curl-curl equation, is assumed to vanish within stranded conductors

$$\sigma \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t}\bigg|_{\Omega_{\mathrm{str},k}} = \mathbf{0} \quad \text{with } k = 1, \ldots, N_{\mathrm{str}}. \tag{2.144}$$

We assume furthermore windings with constant cross-section and thus a homogeneous current distribution holds in the conductor domain, i.e.,

$$\boldsymbol{J} = \sum_{k=1}^{N_{\mathrm{str}}} \boldsymbol{\chi}_{\mathrm{str},k} \, (\mathbf{i}_{\mathrm{str}})_k$$

where $\boldsymbol{\chi}_{\mathrm{str},k}$ is the *winding function* for the k-th stranded conductor with $\operatorname{supp} \boldsymbol{\chi}_{\mathrm{str},l} = \Omega_{\mathrm{str},l}$, such that the curl-curl equation becomes

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) = \sum_{k=1}^{N_{\mathrm{str}}} \boldsymbol{\chi}_{\mathrm{str},k} \, (\mathbf{i}_{\mathrm{str}})_k \ . \tag{2.145a}$$

The flux linked with the winding is given by

$$\psi_k = \int_{\Omega} \boldsymbol{\chi}_{\mathrm{str},k} \cdot \boldsymbol{A} \, \mathrm{d}\Omega$$

and the total voltage drop along the stranded conductor consists of this induced part and a resistive part, i.e.,

$$(\mathbf{v}_{\mathrm{str}})_k = (\mathbf{R}_{\mathrm{str}})_{k,k} \, (\mathbf{i}_{\mathrm{str}})_k + \frac{\mathrm{d}\psi_k}{\mathrm{d}t} \ , \tag{2.145b}$$

where the diagonal DC resistance matrix $\mathbf{R}_{\mathrm{str}}$ can be computed from the model by

$$(\mathbf{R}_{\mathrm{str}})_{k,k} = \int_{\Omega} \frac{1}{f_{\mathrm{str}}} \sigma^{-1} \boldsymbol{\chi}_{\mathrm{str},k} \cdot \boldsymbol{\chi}_{\mathrm{str},k} \, \mathrm{d}\Omega \ , \tag{2.145c}$$

and $f_{\mathrm{str}} \in (0, 1]$ is the fill factor accounting for the cross-sectional fraction of conductive versus insulating materials; in this equation the σ^{-1} is only evaluated in the domains $\Omega_{\mathrm{str},k}$ ($k = 1, \ldots, N_{\mathrm{str}}$), where $\sigma > 0$but anywhere else in Ω the inverse is not necessarily well defined due to non-conducting materials.

Now summing up all excitements for solid and stranded conductors, i.e., Eqs. (2.143)–(2.145), and putting everything together, we obtain the following PDE system

$$\sigma \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t} + \nabla \times (\nu \nabla \times \boldsymbol{A}) = \sum_k \boldsymbol{\chi}_{\mathrm{str},k} \, (\mathbf{i}_{\mathrm{str}})_k + \sum_l \sigma \boldsymbol{\chi}_{\mathrm{sol},l} \, (\mathbf{v}_{\mathrm{sol}})_l \tag{2.146a}$$

$$\int_\Omega \boldsymbol{\chi}_{\mathrm{str},k} \cdot \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t} \, \mathrm{d}\Omega = (\mathbf{v}_{\mathrm{str}})_k - (\mathbf{R}_{\mathrm{str}})_{k,k} \cdot (\mathbf{i}_{\mathrm{str}})_k \, , \tag{2.146b}$$

$$\int_\Omega \sigma \boldsymbol{\chi}_{\mathrm{sol},l} \cdot \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t} \, \mathrm{d}\Omega = (\mathbf{G}_{\mathrm{sol}})_{l,l} \cdot (\mathbf{v}_{\mathrm{sol}})_l - (\mathbf{i}_{\mathrm{sol}})_l \, , \tag{2.146c}$$

with Coulomb gauging, flux wall boundary and initial conditions

$$\nabla \cdot \boldsymbol{A} = 0, \qquad \boldsymbol{A} \times \mathbf{n}_\perp = 0 \text{ on } \partial\Omega, \qquad \boldsymbol{A}(\mathbf{r}, t_0) = \boldsymbol{A}_0 \text{ at } t = t_0. \tag{2.146d}$$

Finally the coupling of the field PDE (2.146) and the circuit DAE (2.138) yields the full field/circuit PDAE problem.

2.2.3.3 Discretization

Following the method of lines, a spatial discretization of the PDE has to be applied first and a time discretization of the overall system in the second step. For spatial discretization we apply the Finite Integration Technique (FIT), [63], which translates the continuous Maxwell equations one by one into a space-discrete set, called the Maxwell grid equations (MGE). The topology is approximated by a finite number of cells $\mathbf{V}(n)$ for $1 \leq n \leq N$. In 3D those cells are hexahedra when applying the simplest mesh, such that the scheme is equivalent to the finite-difference time-domain method proposed by Yee, [66]. Other methods (FEM) are analoguously, see [15].

The hexahedra discretization yields a cell complex $\mathbf{G}$, composed of intervals defined by equidistant distributed coordinates x_i, y_j and z_k

$$\begin{aligned} \mathbf{G} :=& \{\mathbf{V}(n) := \mathbf{V}(i,j,k) \mid \mathbf{V}(i,j,k) = [x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]; \\ & i = 1, \ldots, I-1; j = 1, \ldots, J-1; k = 1, \ldots, K-1\}, \end{aligned}$$

where the three indices i, j and k are combined into one space index, which allows us to number the elements consecutively:

$$n = n(i,j,k) = i + (j-1) \cdot I + (k-1) \cdot I \cdot J \ , \tag{2.147}$$

such that $n \leq N := I \cdot J \cdot K$.

The intersection of two volumes is by construction either empty for non-neighboring volumes or one of the following p-cells, where $p \in \{0, 1, 2, 3\}$ denotes the dimension of the geometrical object and $w \in \{x, y, z\}$ a direction in space:

- 0-cell: a simple point $\mathbf{P}(n)$,
- 1-cell: an edge $\mathbf{L}_w(n)$,
- 2-cell: a facet $\mathbf{A}_w(n)$,
- 3-cell: a volume $\mathbf{V}(n)$.

Every object is associated with its smallest numbered connected point $\mathbf{P}(n)$. An edge $\mathbf{L}_w(n)$ connects two in w-direction neighbored points $\mathbf{P}(n)$ and $\mathbf{P}(n')$ $(n < n')$ and is always directed from $\mathbf{P}(n)$ towards $\mathbf{P}(n')$. A facet $\mathbf{A}_w(n)$ is defined by $\mathbf{P}(n)$ and the direction w, in which its normal vector points.

The basic idea of FIT is the usage of two grids, the primary grid $\mathbf{G}$ is supported by the dual grid $\tilde{\mathbf{G}}$, which is identically but shifted in x-, y- and z-direction by half of a cell length, see Fig. 2.4a. The definition of the dual p-cells, i.e., edges $\tilde{\mathbf{L}}_w(n)$, facets $\tilde{\mathbf{A}}_w(n)$ and volumes $\tilde{\mathbf{V}}(n)$ is analogous to the primary grid ($w \in \{x, y, z\}$). In the following each primary p-cell of $\mathbf{G}$ will be related to one $(3 - p)$-cell of $\tilde{\mathbf{G}}$.

As state variables of the FIT, we introduce electric and magnetic voltages and fluxes. They are defined as integrals of the electric and magnetic field strengths and flux densities over geometrical objects of the computational grid, with respect to the directions $w \in \{x, y, z\}$. The state variables are assigned diacritics $(\widehat{\cdot})$ according to their dimension p of the underlying object. The grid voltages over the edges read as

$$\widehat{\mathbf{e}}_w(n) = \int\limits_{\mathbf{L}_w(n)} \boldsymbol{E}\ \mathrm{ds}\,, \qquad \widehat{\mathbf{a}}_w(n) = \int\limits_{\mathbf{L}_w(n)} \boldsymbol{A}\ \mathrm{ds}\,, \qquad \text{and} \qquad \widehat{\mathbf{h}}_w(n) = \int\limits_{\tilde{\mathbf{L}}_w(n)} \boldsymbol{H}\ \mathrm{ds}\,.$$

The fluxes are located on the grid facets and read

$$\widehat{\widehat{\mathbf{b}}}_w(n) = \int\limits_{\mathbf{A}_w(n)} \boldsymbol{B}\ \mathrm{d}\mathbf{A}\,, \qquad \widehat{\widehat{\mathbf{d}}}_w(n) = \int\limits_{\tilde{\mathbf{A}}_w(n)} \boldsymbol{D}\ \mathrm{d}\mathbf{A}\,, \qquad \text{and} \qquad \widehat{\widehat{\mathbf{j}}}_w(n) = \int\limits_{\tilde{\mathbf{A}}_w(n)} \boldsymbol{J}\ \mathrm{d}\mathbf{A}\,.$$

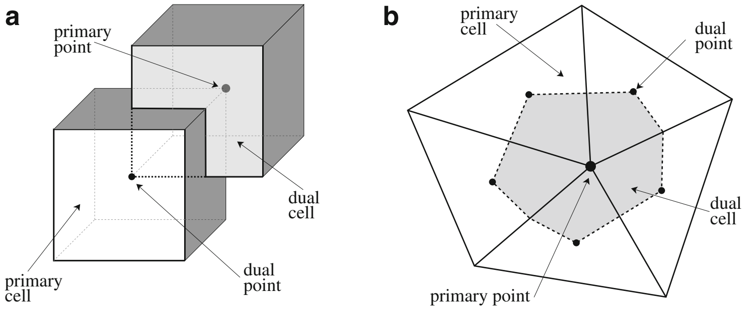


Fig. 2.4 Examples for primary and dual grid cells in 3D and 2D discretizations. (**a**) Staggered hexahedra. (**b**) Barycentric triangulation

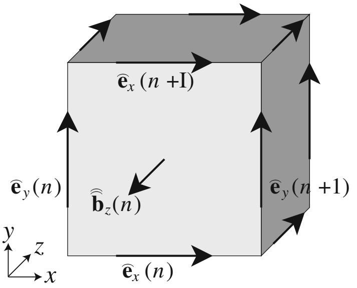

$\widehat{\mathbf{e}}_x\,(n\;+\mathrm{I})$
$\widehat{\mathbf{e}}_y(n)$
$\widehat{\widehat{\mathbf{b}}}_z(n)$
$\widehat{\mathbf{e}}_y(n\;+1)$
y
z
x
$\widehat{\mathbf{e}}_x\,(n)$

Fig. 2.5 Discretization of Faradays Law

To simplify the notation we will build augmented vectors for each of the newly defined quantities with a length of $3N$, including every spatial direction. For example the discrete electric field strengths are collected in

$$\widehat{\mathbf{e}} = (\widehat{\mathbf{e}}_x(1), \ldots, \widehat{\mathbf{e}}_x(N), \widehat{\mathbf{e}}_y(1), \ldots, \widehat{\mathbf{e}}_y(N), \widehat{\mathbf{e}}_z(1), \ldots, \widehat{\mathbf{e}}_z(N))^\top \,. \tag{2.148}$$

The remaining vectors $\widehat{\mathbf{a}}$, $\widehat{\mathbf{h}}$, $\widehat{\widehat{\mathbf{b}}}$, $\widehat{\widehat{\mathbf{d}}}$ and $\widehat{\widehat{\mathbf{j}}}$ are defined analogously.

Using these notations we are able to discretize Maxwell's Equations (2.139) in terms of FIT. For example, Faraday's law, Fig. 2.5, for a single grid facet $A_z(n)$ can be written discretely as

$$\widehat{\mathbf{e}}_x(n) + \widehat{\mathbf{e}}_y(n+1) - \widehat{\mathbf{e}}_x(n+I) - \widehat{\mathbf{e}}_y(n) = -\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\widehat{\mathbf{b}}}_z(n) \,, \tag{2.149}$$

which exploits the new order of numbering and is easily generalized to all facets. The relations for all grid facets are collected in the matrix equation

$$\underbrace{\begin{pmatrix} & & & \vdots & & \\ \cdots\,1 & \cdots\,-1 & \cdots & -1 & 1 & \cdots \\ & & & \vdots & & \end{pmatrix}}_{\mathbf{C}} \underbrace{\begin{pmatrix} \vdots \\ \widehat{\mathbf{e}}_x(n) \\ \vdots \\ \widehat{\mathbf{e}}_x(n+I) \\ \vdots \\ \widehat{\mathbf{e}}_y(n) \\ \widehat{\mathbf{e}}_y(n+1) \\ \vdots \end{pmatrix}}_{\widehat{\mathbf{e}}} = -\frac{\mathrm{d}}{\mathrm{d}t} \underbrace{\begin{pmatrix} \vdots \\ \widehat{\widehat{\mathbf{b}}}_z(n) \\ \vdots \end{pmatrix}}_{\widehat{\widehat{\mathbf{b}}}} \,. \tag{2.150}$$

Applying this procedure to all continuous MQS equations yields the MQS *Maxwell's Grid Equations*, where the differential operators are represented by the discrete curl operators $\mathbf{C}$, $\tilde{\mathbf{C}} = \mathbf{C}^{\top}$ and divergence operators $\mathbf{S}$, $\tilde{\mathbf{S}}$, which live on the primary and dual grid, respectively

$$\begin{aligned} \mathbf{C}\widehat{\mathbf{e}} &= -\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\widehat{\mathbf{b}}}\,, & \tilde{\mathbf{C}}\widehat{\mathbf{h}} &= \widehat{\widehat{\mathbf{j}}}\,, \\ \tilde{\mathbf{S}}\widehat{\widehat{\mathbf{d}}} &= \mathbf{q}\,, & \mathbf{S}\widehat{\widehat{\mathbf{b}}} &= \mathbf{0}\,, \end{aligned} \tag{2.151}$$

with the vector $\mathbf{q}$ containing the electrical charges allocated at the dual grid cells, resembles closely the continuous system (2.139) and maintains several of its properties.

The laws of the continuous magnetic vector potential (2.140) transfer to

$$\mathbf{C}\widehat{\mathbf{a}} = \widehat{\widehat{\mathbf{b}}} \qquad \text{and} \qquad \widehat{\mathbf{e}} = -\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} - \mathbf{S}^{\top}\phi\,, \tag{2.152}$$

with the discrete electric scalar potential ϕ. The discrete potentials are not uniquely defined, similar to the continuous ones, because the curl matrix $\mathbf{C}$ has a non-trivial nullspace.

Working towards a complete discretization of Maxwell's Equations, the material relations (2.139b) have to be given in terms of the discrete quantities. This relates the fluxes on the primary grid $\mathbf{G}$ to the voltages on the dual analogon $\tilde{\mathbf{G}}$ and vice versa. Hence, the material relations establish a coupling between both grids, but their construction requires approximations through averaging processes and here lies the fundamental difference between the various discretization approaches, e.g. FEM and FIT, [15]. FIT has the advantage, that for isotropic and anisotropic materials, whose principal directions coincide with the mesh directions, the material matrices are always diagonal.

For example the magnetic flux density $\boldsymbol{B}$ is related to the magnetic field strength $\boldsymbol{H}$ through the permeability μ. In coherence with our earlier requirements we will assume that there are local permeabilities $\mu(n)$ for each grid volume $\mathbf{V}(n)$. When we start with the definition of the discrete magnetic field strength in conjunction with constitutive relation and averaging its value over the facet $\mathbf{A}_w(n)$ to $|\boldsymbol{B}|$, we get the integral quantity

$$\begin{aligned} \widehat{\mathbf{h}}_w(n) &= \int\limits_{\tilde{\mathbf{L}}_w(n)} \boldsymbol{H}\cdot \mathrm{d}\boldsymbol{s} = \int\limits_{\tilde{\mathbf{L}}_w(n)} \mu^{-1}\boldsymbol{B}\cdot \mathrm{d}\boldsymbol{s} \\ &= \bar{\mu}^{-1}(n)\,|\tilde{\mathbf{L}}_w(n)|\cdot|\boldsymbol{B}| + \mathcal{O}(h^l) && (2.153) \\ &\doteq \bar{\mu}^{-1}(n)\,|\tilde{\mathbf{L}}_w(n)|\cdot|\boldsymbol{B}|\,, && (2.154) \end{aligned}$$

with averaged permeabilities $\bar{\mu}(n)$ that gives an error, whose order l depends on the used discretization grid (in this particular case of a Cartesian grid $l = 2$) and the maximum length of the cell edges $h := \max \mathbf{L}_w(n)$, with $w \in \{x, y, z\}$ and $1 \leq n \leq N$. In a similar manner, we derive for the magnetic flux density

$$\begin{aligned}\widehat{\widehat{\mathbf{b}}}_w(n) &= \int\limits_{\mathbf{A}_w(n)} \boldsymbol{B} \cdot \mathrm{d}\mathbf{A} \\ &= |\mathbf{A}_w(n)| \cdot |\boldsymbol{B}| + \mathscr{O}(h^{l+1}) \\ &\doteq |\mathbf{A}_w(n)| \cdot |\boldsymbol{B}|. \end{aligned} \tag{2.155}$$

Both Eqs. (2.154) and (2.155) contain the averaged magnetic flux density $|\boldsymbol{B}|$, which is unknown. Eliminating this unknown through inserting one equation into the other leads to

$$\widehat{\widehat{\mathbf{b}}}_w(n) = \underbrace{\bar{\mu}(n) \frac{|\mathbf{A}_w(n)|}{|\tilde{\mathbf{L}}_w(n)|}}_{=:\bar{\mu}_w(n)} \cdot \widehat{\mathbf{h}}_w(n) \ ,$$

finally arranging of these permeabilities as a matrix gives

$$\mathbf{M}_\mu := \mathrm{diag}\left(\bar{\mu}_x(1), \ldots, \bar{\mu}_x(N), \bar{\mu}_y(1), \ldots, \bar{\mu}_y(N), \bar{\mu}_z(1), \ldots, \bar{\mu}_z(N)\right) .$$

Similarly the other two material matrices are obtained, such that the laws can be given as

$$\widehat{\widehat{\mathbf{j}}} = \mathbf{M}_\sigma \widehat{\mathbf{e}} \ , \qquad \widehat{\widehat{\mathbf{d}}} = \mathbf{M}_\varepsilon \widehat{\mathbf{e}} \ , \qquad \widehat{\mathbf{h}} = \mathbf{M}_\nu \widehat{\widehat{\mathbf{b}}} \ ,$$

where $\mathbf{M}_\sigma$, $\mathbf{M}_\varepsilon$ and $\mathbf{M}_\mu$ are the (diagonal) matrices of conductivities, permittivities and reluctivities. As before in the continuous case the first two matrices are assumed to be constant, and the reluctivity matrix $\mathbf{M}_\nu = \mathbf{M}_\nu(\widehat{\widehat{\mathbf{b}}})$ may depend nonlinearly on the magnetic flux. Furthermore the matrices of permittivities and reluctivities (for all $\widehat{\widehat{\mathbf{b}}}$) are positive definite, while the conductivity matrix is only positive semi-definite, due to vanishing conductances in electrical insulators.

2.2.3.4 Discrete Vector Potential Formulation

Now having obtained a discrete version of Maxwell's Equations, we can deduce the discrete curl-curl equation with the same steps we used to derive the continuous formulation. The PDE (2.146) becomes the following space-discrete DAE

$$\mathbf{M}_\sigma \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} + \tilde{\mathbf{C}}\mathbf{M}_\nu \mathbf{C}\widehat{\mathbf{a}} = \mathbf{Q}_{\mathrm{str}}\mathbf{i}_{\mathrm{str}} + \mathbf{M}_\sigma \mathbf{Q}_{\mathrm{sol}}\mathbf{v}_{\mathrm{sol}}\ , \tag{2.156a}$$

$$\mathbf{Q}_{\mathrm{str}}^\top \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} = \mathbf{v}_{\mathrm{str}} - \mathbf{R}_{\mathrm{str}}\mathbf{i}_{\mathrm{str}}\ , \tag{2.156b}$$

$$\mathbf{Q}_{\mathrm{sol}}^\top \mathbf{M}_\sigma \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} = \mathbf{G}_{\mathrm{sol}}\mathbf{v}_{\mathrm{sol}} - \mathbf{i}_{\mathrm{sol}}\ , \tag{2.156c}$$

where the matrix $\mathbf{Q} = [\mathbf{Q}_{\mathrm{sol}}, \mathbf{Q}_{\mathrm{str}}]$ is the discrete analogue to the characteristic functions $\boldsymbol{\chi}$ in the continuous model: each column of this matrix corresponds to a conductor model and imposes currents/voltages onto edges of the grid, while each row in the transposed matrix $\mathbf{Q}^\top$ corresponds to the integration of the vector potential over the domain Ω in system (2.146). The conductor domains shall not overlap and we assume this to be true even after the spatial discretization, which affects the coupling matrix as follows

$$(\mathbf{Q})_{k,m}\,(\mathbf{Q})_{m,l} = 0 \text{ for all } m \text{ and } k \neq l. \tag{2.157}$$

Additionally we find especially for the stranded conductor coupling matrix

$$(\mathbf{M}_\sigma)_{k,m}\,(\mathbf{Q}_{\mathrm{str}})_{m,l} = 0 \text{ for all } m, k \text{ and } l, \tag{2.158}$$

which is a consequence of the disregard of eddy currents in stranded conductors, see Eq. (2.144). The matrices of lumped resistances and conductivities are extracted from the model, as explained in Eqs. (2.143c) and (2.145c) and they read in their discrete form as

$$\mathbf{R}_{\mathrm{str}} := \mathbf{Q}_{\mathrm{str}}^\top \mathbf{M}_{\sigma,\mathrm{str}}^+ \mathbf{Q}_{\mathrm{str}} \qquad \text{and} \qquad \mathbf{G}_{\mathrm{sol}} := \mathbf{Q}_{\mathrm{sol}}^\top \mathbf{M}_\sigma \mathbf{Q}_{\mathrm{sol}}\ , \tag{2.159}$$

where $\mathbf{M}_{\sigma,\mathrm{str}}^+$ is the pseudo-inverse of the conductivity matrix with conductivities only in the stranded conductor domains, hence

$$(\mathbf{M}_{\sigma,\mathrm{str}})_{k,m}\,(\mathbf{Q}_{\mathrm{sol}})_{m,l} = 0 \text{ for all } m, k, l \qquad \text{and} \qquad \mathbf{M}_\sigma \mathbf{M}_{\sigma,\mathrm{str}}^+ = \mathbf{0}\ , \tag{2.160}$$

where $\mathbf{M}_{\sigma,\mathrm{str}}^+$ is the pseudo-inverse of $\mathbf{M}_{\sigma,\mathrm{str}}$.

2.2.3.5 Gauging of the Curl-Curl Equation

In 3D the curl-curl equation (2.156a) has no unique solution since both the conductivity matrix $\mathbf{M}_\sigma$ and the curl operator $\mathbf{C}$ have non-trivial nullspaces, and thus the matrix pencil

$$\lambda \mathbf{M}_\sigma + \tilde{\mathbf{C}} \mathbf{M}_\nu \mathbf{C} \text{ for } \lambda \in \mathbb{R}$$

is in general only positive semi-definite, but a gauging, can enforce positive definiteness. For example a special Coulomb gauging, see (2.142), which applies only to the non-conducting parts of the problem, is proposed in [18]

$$\tilde{\mathbf{S}} \mathbf{M}_{\hat{\sigma}} \widehat{\mathbf{a}} = \mathbf{0} \; ,$$

where $\mathbf{M}_{\hat{\sigma}}$ is a special material matrix with artificial conductivities on the diagonal, if the entry corresponds to a non-conducting material, such that all its columns are in the nullspace of $\mathbf{M}_\sigma$. Using a Schur complement the restriction can by integrated into the curl-curl matrix, which becomes for example

$$\mathbf{K}_\nu := \tilde{\mathbf{C}} \mathbf{M}_\nu \mathbf{C} - \mathbf{M}_{\hat{\sigma}} \tilde{\mathbf{S}}^\top \mathbf{N} \tilde{\mathbf{S}} \mathbf{M}_{\hat{\sigma}}$$

and gives the grad-div regularization, [19]. Finally the matrix pencil $\lambda \mathbf{M}_\sigma + \mathbf{K}_\nu$ is positive definite for a simply connected domain Ω (without cavities), if the matrix $\mathbf{N}$ is negative definite, [18].

The positive definiteness of the gauged matrix pencil can still be enforced, if nonlinear reluctivities are considered, i.e., $\mathbf{M}_\nu = \mathbf{M}_\nu(\widehat{\widehat{\mathbf{b}}})$. The structure and hence the kernel of the nonlinear curl-curl matrix remain unchanged, as the following derivative shows

$$\frac{\mathrm{d}}{\mathrm{d}\widehat{\mathbf{a}}} \left(\tilde{\mathbf{C}} \mathbf{M}_\nu(\widehat{\widehat{\mathbf{b}}}) \mathbf{C} \widehat{\mathbf{a}} \right) = \tilde{\mathbf{C}} \frac{\mathrm{d}}{\mathrm{d}\widehat{\widehat{\mathbf{b}}}} \left(\mathbf{M}_\nu(\widehat{\widehat{\mathbf{b}}}) \mathbf{C} \widehat{\mathbf{a}} \right) \frac{\mathrm{d}\widehat{\widehat{\mathbf{b}}}}{\mathrm{d}\widehat{\mathbf{a}}} = \tilde{\mathbf{C}} \frac{\mathrm{d}}{\mathrm{d}\widehat{\widehat{\mathbf{b}}}} \left(\mathbf{M}_\nu(\widehat{\widehat{\mathbf{b}}}) \widehat{\widehat{\mathbf{b}}} \right) \mathbf{C} = \tilde{\mathbf{C}} \frac{\mathrm{d}\widehat{\mathbf{h}}}{\mathrm{d}\widehat{\widehat{\mathbf{b}}}} \mathbf{C} \; ,$$

where both the reluctivity matrix $\mathbf{M}_\nu(\widehat{\widehat{\mathbf{b}}})$ and the differential reluctivity matrix $\mathbf{M}_{\nu,\mathrm{d}} := \mathrm{d}\widehat{\mathbf{h}}/\mathrm{d}\widehat{\widehat{\mathbf{b}}}$ are still positive definite, [33]. In any case only the (constant) nullspace of the curl-operator has to be covered by the gauging and thus it is assumed in the following that

$$\lambda \mathbf{M}_\sigma + \mathbf{K}_\nu(\widehat{\widehat{\mathbf{b}}}) \qquad \text{and} \qquad \lambda \mathbf{M}_\sigma + \frac{\mathrm{d}}{\mathrm{d}\widehat{\mathbf{a}}} \left(\mathbf{K}_\nu(\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} \right)$$

are positive definite for a $\lambda \in \mathbb{R}$.

2.2.3.6 Structure of the Coupled System

Having now transformed the field PDE into a uniquely solvable DAE, we discuss in the following the coupling of the subproblems using a more abstract formulation.

Lemma 2.1 *The field system* (2.156) *is equivalent to the semi-explicit initial value problem*

$$
\begin{aligned}
\dot{\mathbf{y}}_2 &= \mathbf{f}_2(\mathbf{y}_2, \mathbf{z}_{2a}, \mathbf{v}_1), \quad \textit{with} \quad \mathbf{y}_2(0) = \mathbf{y}_{2,0}, \\
\mathbf{0} &= \mathbf{g}_{2a}(\mathbf{y}_2, \mathbf{z}_{2a}), \\
\mathbf{0} &= \mathbf{g}_{2b}(\mathbf{y}_2, \mathbf{z}_{2a}, \mathbf{z}_{2b}),
\end{aligned}
\tag{2.161}
$$

where $\mathbf{y}_2 := \mathscr{P}_\sigma \widehat{\mathbf{a}}$, $\mathbf{z}_{2a} := \mathscr{Q}_\sigma \widehat{\mathbf{a}}$, $\mathbf{z}_{2b} := (\mathbf{i}_{\mathrm{str}}, \mathbf{i}_{\mathrm{sol}})^\top$*and* $\mathbf{v}_1 := (\mathbf{v}_{\mathrm{str}}, \mathbf{v}_{\mathrm{sol}})^\top$

Proof In a first step system (2.156) is reformulated, such that there are no dependencies on derivatives in the two solid and stranded conductor coupling equations (2.156c) and (2.156b). In a second step the curl-curl equation (2.156a) is split into equations coming from conductive materials and non-conductive materials, since only the first materials did yield a differential term $\frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}}$.

Equation (2.156b) is left-multiplied by $\mathbf{Q}_{\mathrm{str}}\mathbf{R}_{\mathrm{str}}^{-1}$ and added to Eq. (2.156a), which yields

$$
\left(\mathbf{M}_\sigma + \mathbf{Q}_{\mathrm{str}}\mathbf{R}_{\mathrm{str}}^{-1}\mathbf{Q}_{\mathrm{str}}^\top\right) \frac{\mathrm{d}}{\mathrm{d}t}\widehat{\mathbf{a}} + \mathbf{K}_\nu(\widehat{\widehat{\mathbf{b}}})\widehat{\mathbf{a}} = \mathbf{Q}_{\mathrm{str}}\mathbf{R}_{\mathrm{str}}^{-1}\mathbf{v}_{\mathrm{str}} + \mathbf{M}_\sigma\mathbf{Q}_{\mathrm{sol}}\mathbf{v}_{\mathrm{sol}} \ , \tag{2.162a}
$$

where the new mass matrix $\mathbf{M}_\sigma + \mathbf{Q}_{\mathrm{str}}\mathbf{R}_{\mathrm{str}}^{-1}\mathbf{Q}_{\mathrm{str}}^\top$ is still symmetric positive semi-definite and can be interpreted as a special conductivity matrix, but it is obviously less sparse.

Left-multiplying Eq. (2.162a) by $\mathbf{Q}_{\mathrm{str}}^\top\mathbf{M}_{\sigma,\mathrm{str}}^+$ and adding to Eq. (2.156b) gives

$$
\mathbf{i}_{\mathrm{str}} - \mathbf{R}_{\mathrm{str}}^{-1}\mathbf{Q}_{\mathrm{str}}^\top\mathbf{M}_{\sigma,\mathrm{str}}^+\mathbf{K}_\nu(\widehat{\widehat{\mathbf{b}}})\widehat{\mathbf{a}} = \mathbf{0} \ , \tag{2.162b}
$$

because the conductors do not overlap $\mathbf{M}_{\sigma,\mathrm{str}}^+\mathbf{M}_\sigma = \mathbf{0}$, see Eq. (2.160) and due to the definition of the lumped resistance matrix for stranded conductors $\mathbf{Q}_{\mathrm{str}}^\top\mathbf{M}_{\sigma,\mathrm{str}}^+\mathbf{Q}_{\mathrm{str}} = \mathbf{R}_{\mathrm{str}}$ in Eq. (2.145c). Similarly a left-multiplication of Eq. (2.156a) by $\mathbf{Q}_{\mathrm{sol}}^\top$ added to Eq. (2.156c) gives

$$
\mathbf{i}_{\mathrm{sol}} - \mathbf{Q}_{\mathrm{sol}}^\top\mathbf{K}_\nu(\widehat{\widehat{\mathbf{b}}})\widehat{\mathbf{a}} = \mathbf{0} \ , \tag{2.162c}
$$

because of the definition of the lumped solid conductor conductances $\mathbf{G}_{\mathrm{sol}} = \mathbf{Q}_{\mathrm{sol}}^\top\mathbf{M}_\sigma\mathbf{Q}_{\mathrm{sol}}$.

Let us now split the curl-curl Equation (2.162a) according to the conductivity of the materials. The symmetric positive semi-definiteness of the mass matrix guarantees an orthogonal matrix $\mathscr{T}$ that transforms the mass matrix into its Jordan Normal Form

$$
\mathscr{T}\left(\mathbf{M}_\sigma + \mathbf{Q}_{\mathrm{str}}\mathbf{R}_{\mathrm{str}}^{-1}\mathbf{Q}_{\mathrm{str}}^\top\right)\mathscr{T}^\top = \begin{pmatrix} \mathbf{J}_\sigma & \\ & \mathbf{0} \end{pmatrix},
$$

where $\mathbf{J}_\sigma$ is a diagonal matrix consisting of the (positive) eigenvalues of $\mathbf{M}_\sigma$ and $\mathbf{Q}_{\text{str}}\mathbf{R}_{\text{str}}^{-1}\mathbf{Q}_{\text{str}}^\top$. This transformation depends only on the topology, there is neither a dependence on the vector potential nor on the time. Thus its application to the whole Eq. (2.162a) gives automatically a splitting of the vector potential $\widehat{\mathbf{a}}$ into differential and algebraic parts, that is constant in time

$$\mathbf{y}_2 := \mathscr{P}_\sigma \widehat{\mathbf{a}} := \big(\mathbf{I}\ \mathbf{0}\big)\,\mathscr{T}\widehat{\mathbf{a}} \qquad \text{and} \qquad \mathbf{z}_{2a} := \mathscr{Q}_\sigma \widehat{\mathbf{a}} := \big(\mathbf{0}\ \mathbf{I}\big)\,\mathscr{T}\widehat{\mathbf{a}}\ ,$$

such that $\widehat{\mathbf{a}} = \mathscr{P}_\sigma^\top \mathbf{y}_2 + \mathscr{Q}_\sigma^\top \mathbf{z}_{2a}$, while the currents are just collected in an additional algebraic variable

$$\mathbf{z}_{2b} := (\mathbf{i}_{\text{str}}, \mathbf{i}_{\text{sol}})^\top\ .$$

The application of $\mathscr{T}$ to the right hand side of (2.162a) yields

$$\begin{aligned}
&\mathscr{T}\left(\mathbf{Q}_{\text{str}}\mathbf{R}_{\text{str}}^{-1}\mathbf{v}_{\text{str}} + \mathbf{M}_\sigma \mathbf{Q}_{\text{sol}}\mathbf{v}_{\text{sol}}\right)\\
=&\mathscr{T}\left(\mathbf{M}_\sigma + \mathbf{Q}_{\text{str}}\mathbf{R}_{\text{str}}^{-1}\mathbf{Q}_{\text{str}}^\top\right)\mathscr{T}^\top \mathscr{T}\left(\mathbf{Q}_{\text{str}}(\mathbf{Q}_{\text{str}}^\top\mathbf{Q}_{\text{str}})^{-1}\mathbf{v}_{\text{str}} + \mathbf{Q}_{\text{sol}}\mathbf{v}_{\text{sol}}\right)\\
=&\begin{pmatrix}\mathbf{J}_\sigma & \\ & \mathbf{0}\end{pmatrix}\mathscr{T}\left(\mathbf{Q}_{\text{str}}(\mathbf{Q}_{\text{str}}^\top\mathbf{Q}_{\text{str}})^{-1}\mathbf{v}_{\text{str}} + \mathbf{Q}_{\text{sol}}\mathbf{v}_{\text{sol}}\right)
\end{aligned}$$

by just utilizing the properties (2.157)–(2.159) and thus the transformation of Eq. (2.162a) using the new variables read

$$\begin{aligned}
\mathbf{J}_\sigma \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{y} &= -\mathscr{P}_\sigma \mathbf{K}_\nu \mathscr{P}_\sigma^\top \mathbf{y}_2 - \mathscr{P}_\sigma \mathbf{K}_\nu \mathscr{Q}_\sigma^\top \mathbf{z}_{2a} + \mathscr{P}_\sigma\left(\mathbf{Q}_{\text{str}}\mathbf{R}_{\text{str}}^{-1}\mathbf{v}_{\text{str}} + \mathbf{M}_\sigma \mathbf{Q}_{\text{sol}}\mathbf{v}_{\text{sol}}\right)\\
\mathbf{0} &= \mathscr{Q}_\sigma \mathbf{K}_\nu \mathscr{P}_\sigma^\top \mathbf{y}_2 + \mathscr{Q}_\sigma \mathbf{K}_\nu \mathscr{Q}_\sigma^\top \mathbf{z}_{2a}\ . \qquad (2.163)
\end{aligned}$$

The first equation defines the function $\mathbf{f}_2$ after a left-multiplication by the inverse $\mathbf{J}_\sigma^{-1}$ of the Jordan Normal Form, while the second Eq. (2.163) defines the first algebraic constraint $\mathbf{g}_{2a}$. Finally the definition of the additional algebraic constraint $\mathbf{g}_{2b}$ follows immediately from Eqs. (2.162b) and (2.162c).

Using the new abstract notation, the field/circuit coupled problem consists of the two subsystems (2.138) and (2.161), i.e.,

$$\begin{aligned}
\dot{\mathbf{y}}_1 &= \mathbf{f}_1(\mathbf{y}_1, \mathbf{z}_1, \boxed{\mathbf{z}_2}), &\qquad \text{and} \qquad \dot{\mathbf{y}}_2 &= \mathbf{f}_2(\mathbf{y}_2, \mathbf{z}_2, \boxed{\mathbf{z}_1}),\\
\mathbf{0} &= \mathbf{g}_1(\mathbf{y}_1, \mathbf{z}_1, \boxed{\mathbf{z}_2}), & \mathbf{0} &= \mathbf{g}_2(\mathbf{y}_2, \mathbf{z}_2),
\end{aligned}$$

where the coupling terms are highlighted by boxes. Note, that the notation was abused slightly, since the algebraic variables $\mathbf{z}_1$ and $\mathbf{z}_2$ contain more than the actually needed node potentials $\mathbf{u}$ and the currents $\mathbf{i}_{\text{sol}}$ and $\mathbf{i}_{\text{str}}$ through solid and stranded conductors.

The coupled problems from electromagnetics are considered again in Chap. 2.

2.2.4 Thermal and Quantum Effects in Semiconductors

In semiconductor technology, the miniaturization of devices is more and more progressing. As a consequence, the simulation of the today nanoscale semiconductor devices requires advanced transport models that take into account also quantum effects and the heating of the crystal. These effects are not very relevant in micrometer devices, but they are crucial for the electric performance in the nanoscale case.

At semiclassical kinetic level the thermal effects are modeled by describing the energy transport in solids with a phonon gas obeying the Peierls kinetic equation while the charge transport is described by the Boltzmann equation, coupled to the Poisson equation for the electric potential. However such a complex system is very difficult to face from a numerical point of view and the simulations require long CPU times not suitable for CAD purposes in electrical engineering. For this reason simpler macroscopic models are warranted in order to use them in the design of electrical devices. A physically sound way for getting macroscopic models is to consider the moment system associated with the transport equations and obtain the closure relations with the maximum entropy principle (hereafter MEP) [9, 53, 54].

Concerning the quantum effects, the typical physical situation we want to describe is the case when the main contribution to the charge transport is semiclassical while the quantum effects enter as small perturbations. For example, this is reasonable for MOSFETs of characteristic length in the range of 10–20 nanometers under strong electric field. Now the semiclassical Boltzmann equation for electrons is replaced with the Wigner equation and a singular perturbation approach is used with a Chapman-Enskog expansion in the high field scaling.

What follows is based on [56] and [57].

2.2.4.1 The Electron-Phonon System

At semiclassical kinetic level, the transport of energy inside the crystal is modeled through quasi-particles called phonons (Fig. 2.6).

The electron-phonon system is described by the Boltzmann-Peierls equations for the distribution functions of electrons and phonons, coupled to the Poisson equation for the electric potential

$$\frac{\partial f}{\partial t} + \mathbf{v}(\mathbf{k}) \cdot \nabla_x f - \frac{e\,\mathbf{E}}{\hbar} \cdot \nabla_k f = C[f, g^{(ac)}, g^{(np)}], \tag{2.164}$$

$$\frac{\partial g^{(ac)}}{\partial t} + \frac{\partial \omega^{(ac)}}{\partial q^i}\frac{\partial g^{(ac)}}{\partial x^i} = S^{(ac)}[g^{(ac)}, g^{(np)}, f], \tag{2.165}$$

$$\frac{\partial g^{(np)}}{\partial t} = S^{(np)}[g^{(np)}, g^{(ac)}, f], \tag{2.166}$$

$$\mathbf{E} = -\nabla_x \Phi, \quad \nabla(\epsilon \nabla \Phi) = -e(C_D(x) - n), \tag{2.167}$$

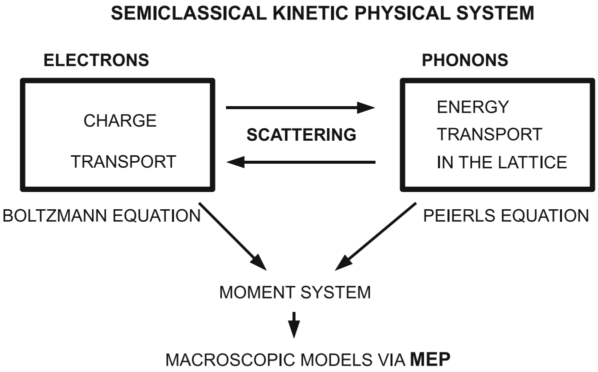


Fig. 2.6 Schematic representation of the electron-phonon system describing the coupling between the charge transport and the crystal thermal effect in a semiconductor and general strategy for getting macroscopic models

where *ac* and *np* stand for acoustic and non-polar optical phonons. $f(\mathbf{x}, \mathbf{k}, t)$ is the electron distribution function which depends on the position $\mathbf{x} \in \mathbb{R}^3$, time t and wave vector $\mathbf{k}$. $C[f, g^{(ac)}, g^{(np)}]$ is the collision operator of electrons with phonons and impurities. We will neglect electron-electron interaction because it is relevant only at very high density, usually not reached in the most common electron devices. e represents the absolute value of the elementary charge. ∇_x and ∇_k denote the nabla operator with respect to $\mathbf{x}$ and $\mathbf{k}$, respectively.

We assume that the conduction bands of semiconductor are described by Kane's dispersion relation

$$\mathcal{E}(k) = \frac{1}{2\alpha}\left[-1 + \sqrt{1 + 4\alpha\frac{\hbar^2 k^2}{2m^*}}\right], \qquad \mathbf{k} \in \mathbb{R}^3,$$

with $\mathcal{E}(k)$ the energy measured from the valley minimum, m^* the effective electron mass, $\hbar\mathbf{k}$ the crystal momentum and α the non parabolicity parameter (for Silicon $\alpha = 0.5eV^{-1}$). Consequently, according to the quantum relation $\mathbf{v} = \frac{1}{\hbar}\nabla_{\mathbf{k}}\mathcal{E}(k)$, the electron velocity is given by the relation $\mathbf{v} = \dfrac{\hbar\mathbf{k}}{m^*\sqrt{1 + 4\alpha\frac{\hbar^2 k^2}{2m^*}}}$.

$g^{(A)} \equiv g(\mathbf{x}, t, \mathbf{q}^{(A)})$ is the phonon distribution of type A (acoustic or non polar optical) which depends on the position $\mathbf{x}$, time t and the wave vector $\mathbf{q}^{(A)}$. $S^{(A)}[g^{(ac)}, g^{(np)}, f]$ is the collision operator of phonons with electrons. The phonon-phonon interaction is described by the relaxation time approximation.

The phonon energy $\hbar\omega^{(A)}$ is related to $\mathbf{q}^{(A)}$ by the dispersion relation. Here we will consider a simplified isotropic model $\omega = \omega(q)$, q being the modulus of $\mathbf{q}$. In particular in the acoustic branch the Debye approximation $\omega = c\,q$ will be adopted with c the Debye velocity, while in the optical branch the Einstein approximation $\omega = \text{const}$ will be used. Moreover we assume that the non-polar optical phonons are described by the Bose-Einstein distribution.

$C_D(x)$ is the doping density, considered as a known function of the position, ϵ is the dielectric constant and $n(x,t)$ the electron number density

$$n(x,t) = \int_{\mathbb{R}^3} f \, \mathrm{d}^3 \mathbf{k}.$$

The direct solution of the system (2.164), (2.167) is very expensive from a computational point of view (for a deterministic numerical solution see [16, 30]) and not practical for electron device design. An alternative approach is to replace the transport equations with a macroscopic model deduced as moment equations of (2.164)–(2.166). These are obtained by multiplying (2.164) with a weight function $\psi(\mathbf{k})$, (2.165) and (2.166) with a weight function $\eta(\mathbf{q})$ and integrating over the first Brillouin zone.

We will consider the 8-moment electron system comprising the balance equations for the electron density, average crystal momentum, energy and energy-flux

$$\frac{\partial n}{\partial t} + \frac{\partial \left(nV^i\right)}{\partial x^i} = 0, \tag{2.168}$$

$$\frac{\partial \left(nP^i\right)}{\partial t} + \frac{\partial \left(nU^{ij}\right)}{\partial x^i} + neE^i = nC_p^i, \tag{2.169}$$

$$\frac{\partial (nW)}{\partial t} + \frac{\partial \left(nS^j\right)}{\partial x^i} + neV_k E^k = nC_W, \tag{2.170}$$

$$\frac{\partial \left(nS^i\right)}{\partial t} + \frac{\partial \left(nF^{ij}\right)}{\partial x^i} + neE_j G^{ij} = nC_W^i, \tag{2.171}$$

coupled to the 9-moment phonon system comprising the balance equation for the phonon energy, average momentum density and the deviatoric part of its flux

$$\frac{\partial u}{\partial t} + Q_k = P_u, \tag{2.172}$$

$$\frac{\partial p_i}{\partial t} + \frac{1}{3}\frac{\partial u}{\partial x_i} + \frac{\partial N_{\langle jk \rangle}}{\partial x_k} = P_i, \tag{2.173}$$

$$\frac{\partial N_{\langle ij \rangle}}{\partial t} + \frac{\partial M_{\langle ij \rangle k}}{\partial x_k} = P_{\langle ij \rangle}. \tag{2.174}$$

The basic quantities entering the electron equations are defined in the kinetic framework as follows (the density has been already defined above)

$$
\begin{aligned}
V^i &= \frac{1}{n}\int_{\mathbb{R}^3} f v^i \mathrm{d}^3\,\mathbf{k} \quad \text{is the average electron velocity,}\\
W &= \frac{1}{n}\int_{\mathbb{R}^3} \mathscr{E}(k)\, f\, \mathrm{d}^3\,\mathbf{k} \quad \text{is the average electron energy,}\\
S^i &= \frac{1}{n}\int_{\mathbb{R}^3} f v^i \mathscr{E}(k) \mathrm{d}^3\,\mathbf{k} \quad \text{is the energy flux,}\\
P^i &= \frac{1}{n}\int_{\mathbb{R}^3} f \hbar k^i \mathrm{d}^3\,\mathbf{k} = m^* \left(V^i + 2\alpha S^i\right) \text{ is the average crystal momentum.}
\end{aligned}
$$

The other electron quantities including production terms are given by

$$
\begin{aligned}
U^{ij} &= \frac{1}{n}\int_{\mathbb{R}^3} f v^i \hbar k^j \mathrm{d}^3\,\mathbf{k}, \quad G^{ij} = \frac{1}{n}\int_{\mathbb{R}^3} \frac{1}{\hbar} f \frac{\partial}{\partial k_j}(\mathscr{E} v_i)\mathrm{d}^3\,\mathbf{k},\\
F^{ij} &= \frac{1}{n}\int_{\mathbb{R}^3} f v^i v^j \mathscr{E}(k) \mathrm{d}^3\,\mathbf{k}\\
nC_W &= \int_{\mathbb{R}^3} C\,[f, g^{(ac)}, g^{(np)}]\mathscr{E}\,(k)\,\mathrm{d}^3\mathbf{k}, \quad nC_p^i = \int_{\mathbb{R}^3} C\,[f, g^{(ac)}, g^{(np)}]\hbar k^i\,\mathrm{d}^3\mathbf{k},\\
nC_W^i &= \int_{\mathbb{R}^3} C\,[f, g^{(ac)}, g^{(np)}] v^i \mathscr{E}\,(k)\,\mathrm{d}^3\mathbf{k}.
\end{aligned}
$$

The basic quantities entering the phonon equations are defined as follows

$$
\begin{aligned}
u &= \int_{\mathbb{R}^3} \hbar\omega\, g\, \mathrm{d}^3\,\mathbf{q} \, \text{is the phonon energy density,}\\
Q_k &= \int_{\mathbb{R}^3} \hbar\omega \frac{\partial \omega}{\partial q_k}\, g\, \mathrm{d}^3\,\mathbf{q} \, \text{is the phonon energy density flux,}\\
p_i &= \hbar \int_{\mathbb{R}^3} q_i\, g\, \mathrm{d}^3\,\mathbf{q} \, \text{is the phonon momentum density,}\\
N_{ik} &= \int_{\mathbb{R}^3} \frac{\hbar\omega}{q^2} q_i q_k\, g\, \mathrm{d}^3\,\mathbf{q} \, \text{is the momentum flux density.}
\end{aligned}
$$

Phonon momentum flux can be decomposed into an isotropic part and a deviatoric part $N_{ik} = \frac{u}{3}\,\delta_{ik} + N_{\langle ik \rangle}$. The deviatoric part of the momentum flux $N_{\langle ik \rangle}$, and its flux are represented by

$$
N_{\langle ij \rangle} = \int_{\mathbb{R}^3} \frac{\hbar\omega}{q^2} q_{<i} q_{j>}\, g\, \mathrm{d}^3\,\mathbf{q}, \quad M_{\langle ij \rangle k} = \int_{\mathbb{R}^3} \frac{\hbar\omega^2}{q^4} q_{<i} q_{j>} q_k\, g\, \mathrm{d}^3\,\mathbf{q}.
$$

The phonon production terms are given by

$$P_u = \int_{\mathbb{R}^3} \hbar\omega \, S[g^{(ac)}, g^{(np)}, f] \, \mathrm{d}^3 \mathbf{q}, \quad P_i = \int_{\mathbb{R}^3} \hbar q_i \, S[g^{(ac)}, g^{(np)}, f] \, \mathrm{d}^3 \mathbf{q},$$

$$P_{\langle ij\rangle} = \int_{\mathbb{R}^3} \frac{\hbar\omega}{q^2} q_{<i} q_{j>} \, S[g^{(ac)}, g^{(np)}, f] \, \mathrm{d}^3 \mathbf{q}.$$

2.2.4.2 The Maximum Entropy Principle

The set of balance equations (2.168)–(2.174) does not form a closed system since more unknowns appear than the number of equations. Therefore the problem of prescribing suitable closure relations arises.

The maximum entropy principle (hereafter MEP) gives a systematic way for obtaining constitutive relations. In the information theory framework the principle has been formalized by Shannon [61]. In statistical physics, it has been introduced in [22, 41] (see also [65] for a review). In [7–9, 49, 53, 54] the approach has been applied to charge transport in semiconductors considering the phonons as a thermal bath. Here the phonons are no longer supposed to be at equilibrium and therefore one has to maximize the phonon distribution.

In the case under investigation, since it is assumed that the non-polar optical phonons are described by Bose-Einstein distribution

$$g_{BE} = \left[\exp\left(\frac{\hbar\omega^{(op)}}{k_B T_L} \right) - 1 \right]^{-1},$$

with T_L the lattice temperature, MEP can be formulated as follows. If a given number of moments $M_A^{(f)}$, $A = 1, \ldots, N$ of f as well as a given number of moments $M_B^{(g)}$, $B = 1, \ldots, M$ of $g = g^{(ac)}$ are known, the distribution functions which can be used to evaluate the unknown moments of f and g, correspond to the maximum, (f_{ME}, g_{ME}), of the entropy functional

$$s(f, g) = -k_B \left[\int_{\mathbb{R}^3} f(\log f - 1) \mathrm{d}^3 \mathbf{k} + \int_{\mathbb{R}^3} \left(g \ln \frac{g}{y} - (y + g) \ln\left(1 + \frac{g}{y}\right) \right) \mathrm{d}^3 \mathbf{q} \right]$$

under the constraints

$$\int_{\mathbb{R}^3} \Psi_A^{(e)}(\mathbf{k}) f_{ME} \mathrm{d}^3 \mathbf{k} = M_A^{(f)}, \quad \int_{\mathbb{R}^3} \Psi_B^{(p)}(\mathbf{q}) g_{ME} \mathrm{d}^3 \mathbf{q} = M_B^{(g)},$$

where $\Psi_A^{(e)}(\boldsymbol{k})$ and $\Psi_B^{(p)}(\boldsymbol{q})$ are electrons and phonons weight functions, respectively, relative to the basic moments $M_A^{(f)}$ and $M_B^{(g)}$. k_B is Boltzmann constant and $y = \dfrac{3}{8\pi^3}$.

From a statistical point of view, f_{ME} and g_{ME} represent the least biased estimators of f and g that can be obtained using only the knowledge of a finite number of moments of f and g. Assuming as fundamental variables for electrons, the density n, the velocity $\mathbf{V}$, the energy W and the energy-flux $\mathbf{S}$, this procedure leads for electrons to the non-equilibrium distribution (see [9, 53])

$$f_{ME} = \exp\left(-\frac{\lambda}{k_B} - \lambda^W \mathscr{E}(k)\right)[1-\chi],$$

with[1] $\chi = \lambda_i^V v^i + \lambda_i^S v_i\ \mathscr{E}(k)$ where Lagrange multipliers associated with the density, the momentum and the energy flux have the expressions

$$\frac{\lambda}{k_B} = -\log\frac{n\hbar^3}{4\pi m^*\sqrt{2m^*}d_0},\ \lambda_i^V = b_{11}V_i + b_{12}S_i,\ \lambda_i^S = b_{12}V_i + b_{22}S_i$$

while λ^W is the Lagrange multiplier related to the energy. It depends on W and it is obtained by inverting the relation

$$W = \frac{\int_0^\infty \mathscr{E}\sqrt{\mathscr{E}(1+\alpha\mathscr{E})}\,(1+2\alpha\mathscr{E})\exp\left(-\lambda^W\mathscr{E}\right)\mathrm{d}\mathscr{E}}{\int_0^\infty \sqrt{\mathscr{E}(1+\alpha\mathscr{E})}\,(1+2\alpha\mathscr{E})\exp\left(-\lambda^W\mathscr{E}\right)\mathrm{d}\mathscr{E}}.$$

The coefficients b_{ij} are given by $b_{11} = \frac{a_{22}}{\Delta}$, $b_{12} = -\frac{a_{12}}{\Delta}$, $b_{22} = \frac{a_{11}}{\Delta}$ with

$$a_{11} = -\frac{2p_0}{3m^*d_0},\ a_{12} = -\frac{2p_1}{3m^*d_0},\ a_{22} = -\frac{2p_2}{3m^*d_0}, \Delta = a_{11}a_{22} - a_{12}^2,$$

including d_k and p_k defined by

$$d_k = \int_0^\infty \mathscr{E}^k\sqrt{\mathscr{E}(1+\alpha\mathscr{E})}\,(1+2\alpha\mathscr{E})\exp\left(-\lambda^W\mathscr{E}\right)\mathrm{d}\mathscr{E},$$

$$p_k = \int_0^\infty \frac{[\mathscr{E}(1+\alpha\mathscr{E})]^{3/2}\,\mathscr{E}^k}{1+2\alpha\mathscr{E}}\exp\left(-\lambda^W\mathscr{E}\right)\mathrm{d}\mathscr{E}.$$

For acoustic phonons, assuming as fundamental variables the energy u, the momentum $\mathbf{p}$ and the deviatoric part of the momentum flux $N_{\langle ij\rangle}$, the following phonon non-equilibrium distribution has been deduced as in [23]

$$g_{ME} \equiv g_{ME}^{(ac)} = g_{BE} + g_{BE}^{+}\left[\frac{3c^2\hbar q}{4uk_BT_L}p_il_i + \frac{15\hbar cq}{8uk_BT_L}\left(N_{ij}l_il_j - \frac{u}{3}\right)\right],$$

[1] Einstein's convention is used: summation with respect repeated dummy indices is understood.

where

$$g_{BE}^{+} = \frac{\exp\left(\frac{\hbar c q}{k_B T_L}\right)}{\left(\exp\left(\frac{\hbar c q}{k_B T_L}\right) - 1\right)^2},$$

and $\mathbf{l} = (l_1, l_2, l_3)$ belongs to S^2, the unit sphere of $\mathbb{R}^3$. We assume as definition of T_L, the Debye relation $u = \sigma\, T_L^4$.

The previous acoustic phonon distribution is valid up to first order in the deviation from the equilibrium.

Putting f_{ME} and g_{ME} into the kinetic definition of the variables appearing in the balance equations (2.168)–(2.174), one gets the desired closure relation in terms of the fundamental variables n, $\mathbf{V}$, W, $\mathbf{S}$, u, $\mathbf{p}$ and $N_{\langle ij\rangle}$.

2.2.4.3 Closure Relations: Phonon Subsystem

Each term is given by the sum of two contributions: one due to the acoustic and another due to the non-polar optical phonons. The details can be found in [56]. Concerning the energy-flux one has $Q_k^{(ac)} = c^2\, p_k^{(ac)}$, $Q_k^{(np)} = 0$ wherefrom

$$Q_k = c^2\, p_k,$$

since $p_k^{(np)} = 0$. Similarly, concerning the divergence of the deviatoric part, one has

$$\frac{\partial M_{\langle ij\rangle k}}{\partial x_k} = c^2 \frac{2}{5} \frac{\partial p_{\langle i}}{\partial x_{j\rangle}}.$$

The production of the energy and the production of the deviatoric part of the momentum flux due to interaction between acoustic phonons and electrons vanishes $P_u^{(ac)} = 0$, $P_{\langle ij\rangle}^{(ac)} = 0$ while the production of momentum for this scattering mechanism is given by

$$\begin{aligned} P_i^{(ac)} &= -n I\, V_i \frac{4\pi\hbar}{3} \int_0^\infty g_{BE}(q)\, (A_0(q)\, b_{11}(W) + A_1(q)\, b_{12}(W))\; q^4\, \mathrm{d}q \\ &\quad - n I\, S_i \frac{4\pi\hbar}{3} \int_0^\infty g_{BE}(q)\, (A_0(q)\, b_{12}(W) + A_1(q)\, b_{22}(W))\; q^4\, \mathrm{d}q, \end{aligned} \tag{2.175}$$

where $I = \dfrac{D_A^2 \hbar^2}{16\pi^2 \rho_S v_s \sqrt{2}(m^*)^{3/2} d_0}$, with D_A^2 the deformation potential, ρ_S the silicon density, v_s the longitudinal sound speed and

$$A_0(q) = \int_{\frac{q}{2}}^{\infty} k \exp\left(-\lambda^W \mathscr{E}\right) \mathrm{d}k, \quad A_1(q) = \int_{\frac{q}{2}}^{\infty} k\mathscr{E} \exp\left(-\lambda^W \mathscr{E}\right) \mathrm{d}k.$$

Since the non-polar optical phonons are described by the Bose-Einstein, the production of momentum is zero along with the deviatoric part of the momentum flux: $P_i^{(np)} = 0, \quad P_{\langle ij\rangle}^{(np)} = 0$.

The energy production can be easily obtained by taking into account that the total energy of the electron-phonon system must be conserved. Since the energy production vanishes in the case of acoustic phonons, we have $P_u^{(np)} = P_u = -nC_W$, where C_W is the electron energy production.

The production terms of energy, momentum and the deviatoric part of the momentum flux arising from the phonon-phonon (ph) acoustic interaction are given by

$$P_u^{(ph)} = 0, \quad P_i^{(ph)} = -\frac{1}{\tau_R} p_i, \quad P_{\langle ij\rangle}^{(ph)} = -\frac{1}{\tau} N_{\langle ij\rangle},$$

where τ_R is the relaxation time for resistive processes and τ is total relaxation time.

Summing up the above relations the production terms read as follows. The production of energy, momentum and deviatoric part of the momentum flux read as

$$\begin{aligned} P_u &= -nC_W, \\ P_i &= n\, c_{11}^{(p)}(W, T_L)\ V_i + n\, c_{12}^{(p)}(W, T_L)\ S_i - \frac{p_i}{\tau_R}, \\ P_{\langle ij\rangle} &= -\frac{N_{\langle ij\rangle}}{\tau}, \end{aligned}$$

where the coefficients $c_{11}^{(p)}(W, T_L)$ and $c_{12}^{(p)}(W, T_L)$ originate from Eq. (2.175).

2.2.4.4 Closure Relations for Electrons

The general expression of the production term for acoustic phonons based on f_{ME} reads as

$$C_{\psi^{(e)}}^{(ac)} =$$
$$I \int_{\mathbb{R}^3} \int_0^{2k} \psi^{(e)}(\mathbf{k}) \left[2g_{BE} + 1\right] \exp\left(-\lambda^W \mathscr{E}\right) \frac{q^4}{2k^2} \left(\lambda_i^V l^i + \lambda_i^S l_i\ \mathscr{E}(k)\right) \mathrm{d}q\, \mathrm{d}^3\mathbf{k}.$$

The production of the energy is zero since the scattering is considered in the elastic approximation $C_W^{(ac)} = 0$. The production of the crystal momentum is given by

$$C_p^{i\ (ac)} = \frac{2\pi I}{3} V_i \int_0^\infty \hbar k \ C\,(k) \exp\left(-\lambda^W \mathscr{E}\right) (b_{11}(W) + \mathscr{E} b_{12}(W))\,\mathrm{d}k$$

$$+\frac{2\pi I}{3} S_i \int_0^\infty \hbar k \ C\,(k) \exp\left(-\lambda^W \mathscr{E}\right) (b_{12}(W) + \mathscr{E} b_{22}(W))\,\mathrm{d}k, \qquad (2.176)$$

where $C\,(k) = \int_0^{2k} q^4\,(2g_{BE} + 1)\,\mathrm{d}q$.

The production of the energy flux has the same structure

$$C_W^{i\ (ac)} = \frac{2\pi I}{3} V_i \int_0^\infty \frac{\hbar k}{m^*} \frac{\mathscr{E}\ C\,(k) \exp\left(-\lambda^W \mathscr{E}\right)}{\sqrt{1 + 4\alpha \frac{\hbar^2 k^2}{2m^*}}} (b_{11}(W) + \mathscr{E} b_{12}(W))\,\mathrm{d}k$$

$$+\frac{2\pi I}{3} S_i \int_0^\infty \frac{\hbar k}{m^*} \frac{\mathscr{E}\ C\,(k) \exp\left(-\lambda^W \mathscr{E}\right)}{\sqrt{1 + 4\alpha \frac{\hbar^2 k^2}{2m^*}}} (b_{12}(W) + \mathscr{E} b_{22}(W))\,\mathrm{d}k. \qquad (2.177)$$

In the case of electron–non-polar optical phonon scattering we have the same expressions already found in [9, 53] but with the lattice temperature which is no longer constant.

Summing up the above results, the production terms in the electron moment system can be written in general forms as the sum of terms due to productions of acoustic and non-polar phonon–electron scattering (electron-electron scattering is negligible). In particular, the production of energy, momentum and energy-flux read

$$\begin{aligned} C_W &= C_W^{(e)}, \\ C_p^i &= c_{11}^{(e)}(W, T_L) V_i + c_{12}^{(e)}(W, T_L) S_i, \\ C_W^i &= c_{21}^{(e)}(W, T_L) V_i + c_{22}^{(e)}(W, T_L) S_i. \end{aligned}$$

where the coefficients $c_{11}^{(e)}(W, T_L)$, $c_{12}^{(e)}(W, T_L)$, $c_{21}^{(e)}(W, T_L)$ and $c_{22}^{(e)}(W, T_L)$ originate from Eqs. (2.176) and (2.177).

2.2.4.5 Limiting Energy Transport and Lattice Heating Model

Under an appropriate scaling, an energy-transport model for electrons coupled to the crystal energy balance equation can be derived. Such a model comprises three balance equations: one for the electron density, one for the electron energy density and one for the crystal temperature. This allows a comparison with the existing models, already known in the literature, for the lattice heating in presence of a charge flow. We assume long time and diffusion scaling, that is with spatial variation on large scale,

$$t = \mathscr{O}\left(\frac{1}{\delta^2}\right), \; x_k = \mathscr{O}\left(\frac{1}{\delta}\right),$$

and that the variables vanishing at equilibrium are of first order

$$\mathbf{V} = \mathscr{O}(\delta), \; \mathbf{S} = \mathscr{O}(\delta), \; \mathbf{p} = \mathscr{O}(\delta), \; \mathbf{N}_{\langle ij \rangle} = \mathscr{O}(\delta),$$

δ being a formal small parameter which is related to the anisotropic part of f_{ME} (see [53]). Moreover we suppose that

$$C_W = \mathscr{O}\left(\frac{1}{\delta^2}\right) \quad \text{and} \quad \tau = \mathscr{O}\left(\frac{1}{\delta^2}\right). \tag{2.178}$$

The last assumptions have the following meaning. If we introduce the energy relaxation time τ_W, one can write $C_W = -\dfrac{W - \frac{3}{2}k_B T_L}{\tau_W}$. Therefore relation (2.178)$_1$ is equivalent to require a long energy relaxation time. Since the experimental data indicates $\tau \geq \tau_W$, it is quite natural to assume also (2.178)$_2$.

By proceedings formally as in [53], we write

$$t = \delta^2 \tilde{t}, \; x = \delta \tilde{x}, \; \mathbf{V} = \delta\,\tilde{\mathbf{V}}, \, \mathbf{S} = \delta\,\tilde{\mathbf{S}}, \, \mathbf{p} = \delta\,\tilde{\mathbf{p}}, \, \mathbf{N}_{\langle ij \rangle} = \delta\,\tilde{\mathbf{N}}_{\langle ij \rangle},$$

and substitute into relations (2.168)–(2.174).

By eliminating the tilde for simplifying the notation, observing that C_P^i and C_W^i are of order δ and by putting equal to zero the coefficients of the various powers of δ in the previous system, one gets again the balance equations (3.72) and (3.74) of density and energy, and moreover

$$\begin{aligned}
&\frac{\partial}{\partial t} n V^i = 0, \quad \frac{\partial}{\partial t} n S^i = 0, \\
&\frac{1}{n}\frac{\partial}{\partial x^j}\, nU^{(0)} = -eE^i + c_{11}^{(e)} V^i + c_{12}^{(e)} S^i, \\
&\frac{1}{n}\frac{\partial}{\partial x^j} nF^{(0)}\delta_{ij} = -eE^i G^{(0)} + c_{21}^{(e)} V^i + c_{22}^{(e)} S^i.
\end{aligned}$$

The last two relations allow to express $\mathbf{V}$ and $\mathbf{S}$ as functions of n, W, T_L and ϕ.

Concerning the phonon part, solving the previous compatibility conditions at each order in δ gives

$$\frac{\partial u}{\partial t} + \frac{\partial c^2 p_k}{\partial x_k} = -n\, C_W, \tag{2.179}$$

$$p_i = -\frac{1}{3}\,\tau_R\,\frac{\partial u}{\partial x_i} + \tau_R\,\left(n\, c_{11}^{(p)}\, V_i + n\, c_{12}^{(p)}\, S_i\right), \tag{2.180}$$

$$\frac{\partial N_{\langle ik \rangle}}{\partial t} = -\frac{\partial N_{\langle ik \rangle}}{\tau}, \tag{2.181}$$

$$\frac{\partial p_{\langle i}}{\partial x_{j\rangle}} = 0. \tag{2.182}$$

We remark that, as expected in a diffusive regime, only the resistive processes are relevant and that neglecting the convective part due to the electron flow $\tau_R\,\left(n\, c_{11}^{(p)}\, V_i + n\, c_{12}^{(p)}\, S_i\right)$ in (2.180) leads to the well known Peierls relation $Q_k = -\frac{1}{3}\, c^2\,\tau_R\,\frac{\partial u}{\partial x_k}$.

Collecting all the previous results, the following *energy transport model for electrons coupled to the lattice energy equation* is obtained

$$\frac{\partial n}{\partial t} + \operatorname{div}(n\mathbf{V}) = 0, \tag{2.183}$$

$$\frac{\partial\,(nW)}{\partial t} + \operatorname{div}(n\mathbf{S}) - ne\mathbf{V}\cdot\nabla\phi = nC_W, \tag{2.184}$$

$$\rho c_V\,\frac{\partial T_L}{\partial t} - \operatorname{div}\left[k(T_L)\nabla T_L\right] = H, \tag{2.185}$$

where $\rho\, c_V = \frac{\partial u}{\partial T_L}$ with c_V specific heat in Silicon at constant volume, $k(T_L) = \frac{1}{3}\rho\, c_V\, c^2\,\tau_R$ is the thermal conductivity and

$$H = -nC_W - c^2 \operatorname{div}\left(\tau_R\, n c_{11}^{(p)}\mathbf{V} + \tau_R n c_{12}^{(p)}\mathbf{S}\right) \tag{2.186}$$

is the crystal energy production.

The electron velocity and energy-flux have the same expression as in [54] but with a lattice temperature which is not kept at equilibrium

$$\mathbf{V} = D_{11}(W, T_L)\,\nabla\log n + D_{12}(W, T_L)\,\nabla W + D_{13}(W, T_L)\,\nabla\phi, \tag{2.187}$$

$$\mathbf{S} = D_{21}(W, T_L)\,\nabla\log n + D_{22}(W, T_L)\nabla\, W + D_{23}(W, T_L)\,\nabla\phi, \tag{2.188}$$

where

$$D_{11}(W,T_L) = D_V\left[c_{12}^{(e)}F - c_{22}^{(e)}U\right], \quad D_{12}(W,T_L) = D_V\left[c_{12}^{(e)}F' - c_{22}^{(e)}U'\right],$$

$$D_{13}(W,T_L) = D_V\left[c_{22}^{(e)}e - c_{12}^{(e)}eG\right], \quad D_V(W,T_L) = \frac{1}{c_{12}^{(e)}c_{21}^{(e)} - c_{22}^{(e)}c_{11}^{(e)}},$$

$$D_{21}(W,T_L) = D_S\left[c_{11}^{(e)}F - c_{21}^{(e)}U\right], \quad D_{22}(W,T_L) = D_S\left[c_{11}^{(e)}F' - c_{21}^{(e)}U'\right],$$

$$D_{23}(W,T_L) = D_S\left[c_{21}^{(e)}e - c_{11}^{(e)}eG\right], \quad D_S(W,T_L) = \frac{1}{c_{22}^{(e)}c_{11}^{(e)} - c_{12}^{(e)}c_{21}^{(e)}}.$$

The explicit form of the coefficients can be easily obtained when taking into the account expressions reported in [9, 53].

In the literature several expressions of H have been proposed (see for more details [60]). In [32] only the Joule effect has been included $H = -e\,n\,\mathbf{V}\cdot\mathbf{E}$, while in [1] the following formulation was suggested $H = -\mathrm{div}\,(E_C\,n\,\mathbf{V})$, with E_C the conduction band edge energy. A different model has been given in [17] $H = -e\,n\,\mathbf{V}\cdot\nabla\phi_n$, with ϕ_n the quasi-Fermi electron potential. It is evident that the previous models can cover only part of the effects present in (2.186).

In order to compare our results with those reported in [62], we sum up Eqs. (3.74) and (3.75), obtaining the balance equation for the total energy

$$\frac{\partial\,(nW)}{\partial t} + \rho c_V\,\frac{\partial T_L}{\partial t} + \mathrm{div}\,(n\mathbf{S} - k(T_L)\nabla T_L) = \\ -\mathbf{J}\cdot\mathbf{E} - c^2\mathrm{div}\left(\tau_R\,nc_{11}^{(p)}\mathbf{V} + \tau_R nc_{12}^{(p)}\mathbf{S}\right), \quad (2.189)$$

where $\mathbf{J} = -en\mathbf{V}$ is the current density. The production terms in Eq. (2.189) are given by a Joule heating term and a divergence term. The argument of the divergence operator can be written as

$$-P_n\,\mathbf{J} - P_S\,n\,\mathbf{S},$$

with $P_n = \dfrac{c^2\,\tau_R\,c_{11}^{(p)}}{e}$ and $P_S = -c^2\,\tau_R\,c_{12}^{(p)}$ a kind of thermoelectric power coefficients. The main difference with [62] (eq. 31 therein, without holes and recombination-generation term), is that $n\,\mathbf{S}$ is not neglected. Moreover, P_n and P_S have an explicit expression directly related to the scattering parameters, and both electrons and lattice have different temperatures.

2.2.4.6 Quantum Corrections

Besides the crystal heating, also quantum effects must been included in the simulation of nanoscale devices. What follows is based on [57]. The starting point is the single particle Wigner-Poisson system in the effective mass approximation which represents the quantum analogous of the semiclassical Boltzmann-Poisson system. In the following the explicit dependence on the lattice temperature will be not written since the results does not change with respect to T_L.

In the effective mass approximation the Wigner-Poisson system reads as

$$\frac{\partial w}{\partial t} + \mathbf{v} \cdot \nabla_x w + \frac{e}{m^*} \Theta[\Phi] w = \mathscr{C}[w], \tag{2.190}$$

$$\text{div}\,(\epsilon \nabla \Phi_S) = -e(C_D(x) - n). \tag{2.191}$$

where the unknown function $w(\mathbf{x}, \mathbf{v}, t)$, depending on the position, velocity and time, is the Wigner quasi distribution, defined as

$$\begin{aligned} w(\mathbf{x}, \mathbf{v}, t) = \; & \mathscr{F}^{-1}[\rho(\mathbf{x} + \frac{\hbar}{2m^*}\boldsymbol{\eta}, \mathbf{x} - \frac{\hbar}{2m^*}\boldsymbol{\eta}, t)](\mathbf{v}) = \\ & \frac{1}{(2\pi)^3} \int_{\mathbb{R}^3} \rho(\mathbf{x} + \frac{\hbar}{2m^*}\boldsymbol{\eta}, \mathbf{x} - \frac{\hbar}{2m^*}\boldsymbol{\eta}, t)\; e^{i\mathbf{v}\cdot\boldsymbol{\eta}}\, \mathrm{d}^3\, \boldsymbol{\eta}. \end{aligned}$$

Here $\rho(\mathbf{x}, \mathbf{y})$ is the density matrix, which is related to the wave function $\psi(\mathbf{x}, t)$ by

$$\rho(\mathbf{x}, \mathbf{y}) = \overline{\psi(\mathbf{x}, t)}\, \psi(\mathbf{y}, t).$$

$\mathscr{F}$ denotes the Fourier transform, given for function $g(\mathbf{v}) \in L^1(\mathbb{R}^3)$ by

$$\mathscr{F}[g(\mathbf{v})](\eta) = \int_{\mathbb{R}^3_v} g(\mathbf{v})\, \mathrm{e}^{-i\mathbf{v}\cdot\eta}\, \mathrm{d}^3\, \mathbf{v},$$

and $\mathscr{F}^{-1}$ the inverse Fourier transform

$$\mathscr{F}^{-1}[h(\boldsymbol{\eta})] = \frac{1}{(2\,\pi)^3} \int_{\mathbb{R}^3_\eta} h(\boldsymbol{\eta})\, \mathrm{e}^{i\mathbf{v}\cdot\boldsymbol{\eta}}\, \mathrm{d}^3\, \boldsymbol{\eta}.$$

The potential Φ is usually given by the sum of a self-consistent term Φ_S, solution of the Poisson equation (2.191), and an additional term Φ_B which models the potential barriers in hetero-junctions and is a prescribed function of the position.

As well known, $w(\mathbf{x}, \mathbf{v}, t)$ is not in general positive definite. However it is possible to calculate the macroscopic quantities of interest as expectation values (moments) of $w(\mathbf{x}, \mathbf{v}, t)$ in the same way of the semiclassical case, e.g.

$$\begin{aligned}
&\text{density} \quad n(\mathbf{x},t) = \int_{\mathbb{R}^3} w(\mathbf{x},\mathbf{v},t)\, \mathrm{d}^3\,\mathbf{v}, \\
&\text{velocity} \quad V(\mathbf{x},t) = \frac{1}{n(\mathbf{x},t)} \int_{\mathbb{R}^3} \mathbf{v}\, w(\mathbf{x},\mathbf{v},t)\, \mathrm{d}^3\,\mathbf{v}, \\
&\text{energy} \quad W(\mathbf{x},t) = \frac{1}{n(\mathbf{x},t)} \int_{\mathbb{R}^3} \frac{1}{2}\, m^*\ v^2\, w(\mathbf{x},\mathbf{v},t)\, \mathrm{d}^3\,\mathbf{v}, \\
&\text{energy-flux} \quad S(\mathbf{x},t) = \frac{1}{n(\mathbf{x},t)} \int_{\mathbb{R}^3} \frac{1}{2}\, m^*\ \mathbf{v}\, v^2\, w(\mathbf{x},\mathbf{v},t)\, \mathrm{d}^3\,\mathbf{v}.
\end{aligned}$$

It is worth to mention that the previous definition of energy and energy flux are valid only in the parabolic band, consistently with the effective mass approximation.

$\Theta[\Phi]$ represents the pseudo-differential operator

$$\begin{aligned}
\Theta[\Phi] w(\mathbf{x},\mathbf{v},t) = \frac{i m^*}{\hbar (2\pi)^3} \int_{\mathbb{R}^3_\eta \times \mathbb{R}^3_{v'}} &\left[\Phi\left(\mathbf{x} + \frac{\hbar}{2m^*}\boldsymbol{\eta},\, t\right) - \Phi\left(\mathbf{x} - \frac{\hbar}{2m^*}\boldsymbol{\eta},\, t\right)\right] \\
&w(\mathbf{x},\mathbf{v}',t)\, \mathrm{e}^{-i(\mathbf{v}'-\mathbf{v})\cdot\boldsymbol{\eta}}\, \mathrm{d}^3\,\mathbf{v}'\, \mathrm{d}^3\,\boldsymbol{\eta}.
\end{aligned}$$

$\mathscr{C}[w]$ is the quantum collision term. Its formulation is itself an open problem. Some attempts can be found in [10, 28], but a derivation suitable for application in electron devices is still lacking. Here we propose an expression which is a perturbation of the semiclassical collision term, useful for the formulation of macroscopic models.

As general guideline $\mathscr{C}[w]$ should drive the system towards the equilibrium. If we consider the electrons in a thermal bath at the lattice temperature $T_L = 1/k_B\beta$, the equilibrium Wigner function w_{eq} has been found in [64].

For our purposes we *locally* parameterize the equilibrium Wigner function in terms of the electron density. Up to first order in $\hbar^2$ on has

$$\begin{aligned}
w_{eq} = w_{eq}^{(0)} + \hbar^2\, w_{eq}^{(1)} + \mathscr{O}(\hbar^4) = n(\mathbf{x},t) \left(\frac{m^*\beta}{2\,\pi}\right)^{3/2} \mathrm{e}^{-\beta\,\mathscr{E}} \times \\
\left\{ 1 + \frac{\hbar^2 \beta^2 e}{24} \left[\frac{\Delta\,\Phi}{m^*} - \beta\ v_r\ v_s\ \frac{\partial^2 \Phi}{\partial\, x_r\, \partial\, x_s} \right] \right\} + \mathscr{O}(\hbar^4),
\end{aligned}$$

where

$$w_{eq}^{(0)} = n(\mathbf{x},t) \left(\frac{m^*\beta}{2\,\pi}\right)^{3/2} \mathrm{e}^{-\beta\,\mathscr{E}}$$

is the classical Maxwellian.

We suppose that the expansion

$$w = w^{(0)} + \hbar^2 w^{(1)} + \mathscr{O}(\hbar^4)$$

holds. By proceedings in a formal way, as $\hbar \mapsto 0$ the Wigner equation gives the semiclassical Boltzmann equation in the parabolic band approximation

$$\frac{\partial w^{(0)}}{\partial t} + \mathbf{v} \cdot \nabla_x w^{(0)} + \frac{e}{m^*} \nabla_x \Phi \cdot \nabla_v w^{(0)} = \mathscr{C}^{(0)}[w^{(0)}]. \tag{2.192}$$

At first order in $\hbar^2$ we have

$$\frac{\partial w^{(1)}}{\partial t} + \mathbf{v} \cdot \nabla_x w^{(1)} + \frac{e}{m^*} \nabla_x \Phi \cdot \nabla_v w^{(1)} - \frac{e}{24 m^3} \frac{\partial^3 \Phi}{\partial x_i\, x_j\, x_k} \frac{\partial^3 w^{(0)}}{\partial v_i\, v_j\, v_k} = \mathscr{C}^{(1)}, \tag{2.193}$$

with $\mathscr{C}^{(1)}$ to be modeled.

Since $w^{(0)}$ must be positive, being a solution of the semiclassical Boltzmann equation, we make the following first assumption

$$\mathscr{C}[w] = \mathscr{C}^{(0)}[w^{(0)}] + \hbar^2 \mathscr{C}^{(1)}[w^{(1)}] = \mathscr{C}_C[w^{(0)}] - \hbar^2 \nu \left(w^{(1)} - w^{(1)}_{eq}\right) + \mathscr{O}(\hbar^4) \tag{2.194}$$

with $\mathscr{C}_C[w^{(0)}]$ semiclassical collision operator $(w^{(0)} > 0!)$

and $\nu > 0$ quantum collision frequency.

Remark 2.3 At variance with other approaches, only the $\hbar^2$ correction to the collision term has a relaxation form. This assures that as $\hbar \mapsto 0$ one gets the semiclassical scattering of electrons with phonons and impurities.

The value of the quantum collision frequency ν is a fitting parameter that can be determined comparing the results with the experimental data.

We require that $\mathscr{C}[w]$ conserves the electron density (second assumption)

$$\int_{\mathbb{R}^3} \mathscr{C}[w]\, \mathrm{d}^3 \mathbf{v} = 0.$$

Proposition 2.1 *The collision operator $\mathscr{C}[w]$ of the form (2.194) satisfies up to terms $\mathscr{O}(\hbar^4)$ the following properties:*

1. Ker $(\mathscr{C}[w])$ is given by the quantum Maxwellian

$$w_{(eq)} = w^{(0)}_{eq} + \hbar^2 w^{(1)}_{eq},$$

with $w^{(0)}_{eq}$ the classical Maxwellian.

2.

$$-k_B \int_{\mathbb{R}^3} \mathscr{C}^{(0)}[w^{(0)}] \ln \frac{w^{(0)}}{\exp(-\frac{\beta m^* v^2}{2})} \mathrm{d}^3 \boldsymbol{v} = \\ -k_B \int_{\mathbb{R}^3} \left[\ln w^{(0)} + \frac{\beta m^* v^2}{2} \right] \mathscr{C}^{(0)} \mathrm{d}^3 \boldsymbol{v} \geq 0,$$

3.

$$-\frac{1}{2} \mathscr{C}^{(1)}[w^{(1)}] \left(w^{(1)} - w_{eq}^{(1)} \right) \geq 0.$$

Moreover the equality holds if and only if w is the quantum Maxwellian, defined above.

Properties 1 and 3 are straightforward. Property 2 is based on the proof in [45–47] valid in the classical case.

2.2.4.7 Quantum Corrections in the High Field Approximation

In the case of high electric fields, it is possible to get an approximation for $w^{(1)}$ by a suitable Chapman-Enskog expansion. Let us introduce the dimensionless variables $\tilde{\mathbf{x}} = \frac{\mathbf{x}}{l_0}$, $\tilde{t} = \frac{t}{t_0}$, $\tilde{\mathbf{v}} = \frac{\mathbf{v}}{v_0}$, with l_0, t_0 and $v_0 = l_0/t_0$ typical length, time and velocity. Let l_Φ be the characteristic length of the electrical potential and $1/t_C$ the characteristic collision frequency. After scaling the collision frequency according to $\tilde{\nu} = t_C\, \nu$, Eq. (2.193) can be rewritten as

$$\frac{1}{t_0} \frac{\partial w^{(1)}}{\partial t} + \frac{v_0}{l_0} \mathbf{v} \cdot \nabla_x w^{(1)} + \frac{v_0}{l_\Phi} \left[\frac{e}{m^*} \nabla_x \Phi \cdot \nabla_v w^{(1)} \right. \\ \left. - \frac{e}{24 m^3} \frac{\partial^3 \Phi}{\partial x_i\, x_j\, x_k} \frac{\partial^3 w^{(0)}}{\partial v_i\, v_j\, v_k} \right] = -\frac{1}{t_C} \nu \left(w^{(1)} - w_{eq}^{(1)} \right).$$

We will continue to denoted the scaled variables as the unscaled ones for simplifying the notation. Note that the scaling of $w^{(1)}$ is unimportant.

Let us introduce the characteristic length associated with the quantum correction of the collision term (a kind of mean free path in a semiclassical context) $l_C = v_0\, t_C$ We assume that the quantum effects occur in the high field and collision dominated regime, where drift and collision mechanisms have the same characteristic length. Therefore we set formally $\frac{l_C}{l_\Phi} = 1$ and observe that in the high frequency regime the Knudsen number $\alpha = \dfrac{l_C}{l_0}$ is a small parameter. Substituting in the previous equation, we get

$$\alpha \frac{\partial w^{(1)}}{\partial t} + \alpha \mathbf{v} \cdot \nabla_x w^{(1)} + \frac{e}{m^*} \nabla_x \Phi \cdot \nabla_v w^{(1)}$$
$$- \frac{e}{24 m^3} \frac{\partial^3 \Phi}{\partial x_i \; x_j \; x_k} \frac{\partial^3 w^{(0)}}{\partial v_i \; v_j \; v_k} = -\nu \left(w^{(1)} - w^{(1)}_{eq} \right).$$

The zero order in α gives

$$\frac{q}{m^*} \nabla_x \Phi \cdot \nabla_v w^{(1)} - \frac{e}{24 m^3} \frac{\partial^3 \Phi}{\partial x_i \; x_j \; x_k} \frac{\partial^3 w^{(0)}}{\partial v_i \; v_j \; v_k} = -\nu \left(w^{(1)} - w^{(1)}_{eq} \right)$$

and by Fourier transforming one has

$$w^{(1)}(\mathbf{x}, \mathbf{v}, t) = \mathscr{F}^{-1} \left\{ \frac{1}{\nu + \frac{ie}{m^*} \boldsymbol{\eta} \cdot \nabla_x \Phi} \left[- \frac{ie}{24 m^{*3}} \frac{\partial^3 \Phi}{\partial x_i \; x_j \; x_k} \eta_i \eta_j \eta_k \mathscr{F} w^{(0)}(\boldsymbol{\eta}) \right. \right.$$
$$\left. \left. + \nu \mathscr{F} w^{(1)}_{eq}(\boldsymbol{\eta}) \right] \right\} (\mathbf{x}, \mathbf{v}, t).$$

Approximating $w^{(0)}$ with f_{ME}, we obtain

$$w(\mathbf{x}, \mathbf{v}, t) \approx f_{ME}(\mathbf{x}, \mathbf{v}, t) + \hbar^2 w^{(1)}(\mathbf{x}, \mathbf{v}, t), \tag{2.195}$$

which will be used in the next section for evaluating the unknown quantities in the moment system, associated with the Wigner equation.

In analogy with the semiclassical case, multiplying (2.190) by suitable weight functions ψ, depending in the physical relevant cases on the velocity $\mathbf{v}$, and integrating over the velocity, one has the balance equation for the macroscopic quantities of interest

$$\frac{\partial}{\partial t} \int_{\mathbb{R}^3} w(\mathbf{x}, \mathbf{v}, t) \;\; \psi(\mathbf{v}) \; \mathrm{d}^3 \, \mathbf{v} + \nabla_x \int_{\mathbb{R}^3} \psi(\mathbf{v}) \mathbf{v} \cdot w \, \mathrm{d}^3 \, \mathbf{v}$$
$$+ \frac{q}{m^*} \int_{\mathbb{R}^3} \psi(\mathbf{v}) \; \Theta[\Phi] w \, \mathrm{d}^3 \, \mathbf{v} = \int_{\mathbb{R}^3} \psi(\mathbf{v}) \, \mathscr{C}[w] \, \mathrm{d}^3 \, \mathbf{v}. \tag{2.196}$$

In the 8-moment model the basic variables are the moments relative to the weight functions $1, m^* \, \mathbf{v}, \dfrac{1}{2} m^* \, v^2, \dfrac{1}{2} m^* \, v^2 \mathbf{v}$.

By evaluating (2.196) for $\psi = 1$, under the assumption that the necessary moments of $w^{(1)}(\mathbf{x}, \mathbf{v}, t)$ and $\dfrac{\partial^3 w^{(0)}}{\partial v_i\, v_j\, v_k}$ with respect to v exist, one has

$$\frac{q}{m^*}\int_{\mathbb{R}^3} \Theta[\Phi] w \,\mathrm{d}^3\,\mathbf{v} = \frac{e}{m^*}\nabla_x \cdot \int_{\mathbb{R}^3} \nabla_v w^{(0)}\,\mathrm{d}^3\,\mathbf{v}$$
$$+\hbar^2\left[\frac{e}{m^*}\nabla_x \Phi \cdot \int_{\mathbb{R}^3} \nabla_v w^{(1)}\,\mathrm{d}^3\,\mathbf{v} - \frac{e}{24m^3}\frac{\partial^3 \Phi}{\partial x_i\, x_j\, x_k}\int_{\mathbb{R}^3}\frac{\partial^3 w^{(0)}}{\partial v_i\, v_j\, v_k}\,\mathrm{d}^3\,\mathbf{v}\right] = 0,$$

obtaining the continuity equation

$$\frac{\partial}{\partial t}n + \frac{\partial (nV_i)}{\partial x^i} = 0. \tag{2.197}$$

In order to get other moment equations we observe that from (2.195) it follows

$$\frac{e}{m^*}\nabla_x \Phi \cdot \int_{\mathbb{R}^3} \psi(\mathbf{v})\, \nabla_v w^{(1)}\,\mathrm{d}^3\,\mathbf{v} \;-\frac{e}{24m^3}\frac{\partial^3 \Phi}{\partial x_i\, x_j\, x_k}\int_{\mathbb{R}^3}\psi(\mathbf{v})\,\frac{\partial^3 w^{(0)}}{\partial v_i\, v_j\, v_k}\,\mathrm{d}^3\,\mathbf{v}$$
$$+\nu\int_{\mathbb{R}^3}\psi(\mathbf{v})\,\left(w^{(1)} - w^{(1)}_{eq}\right)\,\mathrm{d}^3\,\mathbf{v} = 0, \tag{2.198}$$

for each weight function $\psi(\mathbf{v})$ such that the integrals exist.

By taking into account (2.198), multiplying Eq. (2.190) by the weight functions $m^*\,\mathbf{v}$, $\frac{1}{2}\,m^*\,v^2$, $\frac{1}{2}\,m^*\,v^2\mathbf{v}$, after integration one finds the balance equations for momentum, energy and energy-flux

$$\frac{\partial}{\partial t}(nV_i) + \frac{\partial(nU_{ij})}{\partial x^j} + n\,e\,E_i = nC^i_{p^,}\left(W^{(0)}, V_i^{(0)}S_i^{(0)}\right), \tag{2.199}$$
$$\frac{\partial}{\partial t}(nW) + \frac{\partial(nS_j)}{\partial x^j} + neV_k^{(0)}E^k = nC_W(W^{(0)}), \tag{2.200}$$
$$\frac{\partial}{\partial t}(nS_i) + \frac{\partial(nF_{ij})}{\partial x^j} + \frac{5}{3}n\frac{e}{m^*}E_i W^{(0)} = nnC^i_{W^,}\left(W^{(0)}, V_i^{(0)}, S_i^{(0)}\right). \tag{2.201}$$

Here $V_i^{(0)}$, $W^{(0)}$ and $S_i^{(0)}$ are the zero order components of the average velocity, energy and energy-flux. Also for other quantities, the superscript (0) will mean zero order with respect to $\hbar^2$. The components of the flux of momentum and the flux of energy-flux are defined as

$$U_{ij} = \frac{1}{n(x,t)}\int_{\mathbb{R}^3} m^*\,v_i\,v_j\,w(x,v,t)\,\mathrm{d}^3\,\mathbf{v},$$
$$F_{ij} = \frac{1}{n(x,t)}\int_{\mathbb{R}^3}\frac{1}{2}m^* v_i\,v_j\,v^2\,w(x,v,t)\,\mathrm{d}^3\,\mathbf{v}.$$

The production terms are defined as

$$
\begin{aligned}
n\, C_{p}^{i}, &= \int_{\mathbb{R}^3} m^*\, v_i\, \mathscr{C}[w]\, \mathrm{d}^3\, \mathbf{v},\\
n\, C_W &= \int_{\mathbb{R}^3} \frac{1}{2}\, m^* v^2\, \mathscr{C}[w]\, \mathrm{d}^3\, \mathbf{v},\\
n\, n C_{W}^{i}, &= \int_{\mathbb{R}^3} \frac{1}{2}\, m^* v^2\, v_i\, \mathscr{C}[w]\, \mathrm{d}^3\, \mathbf{v}.
\end{aligned}
$$

Remark 2.4 The quantum corrections affect only the free streaming part, while the drift and production terms appear only at the zero order.

Therefore $C_W(W^{(0)})$, $C_{p}^{i},\left(W^{(0)}, V_i^{(0)}, S_i^{(0)}\right)$ and $C_{W}^{i},\left(W^{(0)}, V_i^{(0)}, S_i^{(0)}\right)$ are as in the semiclassical case.

The system (2.197), (2.199)–(2.201) is not closed because of the presence of the unknown quantities U_{ij}, F_{ij}, C_p^i, C_W and C_W^i,. We solve the closure problem with the approximation (2.195), assuming a collision dominated high field regime for the quantum effects. The results are given by the following proposition

Proposition 2.2 *In the high field approximation one has*

$$
\begin{aligned}
J_i &= n\, V_i = n\, V_i^{(0)} + \mathscr{O}(\hbar^4),\\
W &= W^{(0)} - \frac{\hbar^2 \beta\, e}{24 m^*}\, \Delta\, \Phi + \mathscr{O}(\hbar^4),\\
U_{ij} &= U_{ij}^{(0)} - \frac{\hbar^2 \beta\, e}{12 m^*}\, \frac{\partial^2 \Phi}{\partial x_i\, \partial x_j} + \mathscr{O}(\hbar^4),\\
S_i &= S_i^{(0)} - \frac{\hbar^2 \beta^2\, e^2}{24 m^{*2} \nu} \left(2 \frac{\partial^2 \Phi}{\partial x_i\, \partial x_r} \frac{\partial\, \Phi}{\partial\, x_r} + \frac{\partial\, \Phi}{\partial\, x_i} \Delta \Phi \right)\\
&\quad - \frac{\hbar^2\, e}{8 m^{*2} \nu} \frac{\partial}{\partial\, x_i}\, \Delta\, \Phi + \mathscr{O}(\hbar^4),\\
F_{ij} &= F_{ij}^{(0)} - \frac{\hbar^2 \beta\, e^3}{3 m^{*3} \nu^2}\, \frac{\partial\, \Phi}{\partial\, x_{(i}} \frac{\partial^2 \Phi}{\partial x_{j)} \partial x_r}\, \frac{\partial \Phi}{\partial x_r} - \frac{\hbar^2\, e^2}{4 m^{*3} \nu^2}\, \frac{\partial^3 \Phi}{\partial x_i\, \partial x_j\, \partial x_r}\, \frac{\partial \Phi}{\partial x_r}\\
&\quad - \frac{\hbar^2\, \beta\, e^3}{12 m^{*3} \nu^2} \left(\frac{\partial \Phi}{\partial\, x_i}\, \frac{\partial \Phi}{\partial\, x_j}\, \Delta\, \Phi + |\nabla\, \Phi|^2 \frac{\partial^2 \Phi}{\partial x_i\, \partial x_j} \right)\\
&\quad - \frac{\hbar^2\, e^2}{4 m^{*3} \nu^2}\, \frac{\partial \Delta \Phi}{\partial x_{(i}}\, \frac{\partial\, \Phi}{\partial x_{j)}} - \frac{\hbar^2\, e}{24 m^{*2}}\, \left(\Delta \Phi\, \delta_{ij} + 5 \frac{\partial^2 \Phi}{\partial x_i\, \partial x_j} \right)\\
&\quad - \frac{\hbar^2\, e}{4 m^{*2} \nu}\, \left(\frac{\partial \Delta \Phi}{\partial x_{(i}}\, V_{j)} + \frac{\partial^3 \Phi}{\partial x_{(i}\; x_j\; x_{k)}} V_k \right) + \mathscr{O}(\hbar^4).
\end{aligned}
$$

In the previous relationships round brackets indicate symmetrization, e.g.

$$\begin{aligned}
A_{i(jk)} &= \frac{1}{2}\left(A_{ijk} + A_{ikj}\right), \\
A_{(ijk)} &= \frac{1}{3!}\left(A_{ijk} + A_{ikj} + A_{jik} + A_{jki} + A_{kij} + A_{kji}\right).
\end{aligned}$$

Remark 2.5 In the limit of high frequency $\nu \to \infty$ one has the simplified model

$$\begin{aligned}
J_i &= n\,V_i = n\,V_i^{(0)} + \mathscr{O}(\hbar^4), \\
W &= W^{(0)} - \frac{\hbar^2 \beta\, e}{24m^*}\,\Delta\,\Phi + \mathscr{O}(\hbar^4), \\
U_{ij} &= U_{ij}^{(0)}\,\delta_{ij} - \frac{\hbar^2 \beta\, e}{12m^*}\,\frac{\partial^2 \Phi}{\partial x_i\,\partial x_j} + \mathscr{O}(\hbar^4), \\
S_i &= S_i^{(0)} + \mathscr{O}(\hbar^4), \\
F_{ij} &= F_{ij}^{(0)} - \frac{\hbar^2\, e}{24m^{*2}}\left(\Delta\Phi\,\delta_{ij} + 5\frac{\partial^2 \Phi}{\partial x_i\,\partial x_j}\right).
\end{aligned}$$

From Eq. (2.195) one sees that in the limit $\nu \to \infty$, $w^{(1)}$ reduces to the quantum correction of the equilibrium Wigner function $w_{eq}^{(1)}$. The resulting quantum corrections to the tensor U_{ij} are the same as those obtained in [31] by using a shifted Wigner function, but with the semiclassical contribution which contains also a heat flux, not added *ad hoc*.

2.2.4.8 Quantum Corrected Energy-Transport and Crystal Heating Model

Assuming the same scaling of Sect. 2.2.4.4, one gets (formally) the energy-transport equations (3.72) and (3.74) with the closure relations

$$\begin{aligned}
V_i = \frac{1}{\Delta}\, &\left\{ c_{22}^{(e)}\left[\frac{U_{ik}}{n}\frac{\partial n}{\partial x_k} + \frac{\partial U_{ik}}{\partial x_k} - e\frac{\partial \Phi}{\partial x_i}\right]\right. \\
&\left. -c_{12}^{(e)}\left[\frac{F_{ik}}{n}\frac{\partial n}{\partial x_k} + \frac{\partial F_{ik}}{\partial x_k} - \frac{5e}{3m^*}W^{(0)}\frac{\partial \Phi}{\partial x_i}\right]\right\}, \\
S_i = \frac{1}{\Delta}\, &\left\{ c_{11}^{(e)}\left[\frac{F_{ik}}{n}\frac{\partial n}{\partial x_k} + \frac{\partial F_{ik}}{\partial x_k} - \frac{5e}{3m^*}W^{(0)}\frac{\partial \Phi}{\partial x_i}\right]\right. \\
&\left. -c_{21}^{(e)}\left[\frac{U_{ik}}{n}\frac{\partial n}{\partial x_k} + \frac{\partial U_{ik}}{\partial x_k} - e\frac{\partial \Phi}{\partial x_i}\right]\right\},
\end{aligned}$$

where

$$\Delta(W^{(0)}) = c_{11}^{(e)} c_{22}^{(e)} - c_{12}^{(e)} c_{21}^{(e)}.$$

If also the effect of the crystal heating need to be included, the lattice temperature is no longer constant and one has to take into the account equation (3.75) as well.

The zero order terms are strictly valid in the parabolic band case ($\alpha = 0$), in particular the $c_{ij}^{(e)}$'s. A simple way to extend the results in the case of Kane dispersion relation is to consider for the semiclassical part of **V** and **S** the relations (2.187)–(2.188), but including the quantum corrections for U_{ik} and F_{ik} according to the proposition 2.2.

For example, in the case $\nu \to 0$ the complete model reads as

$$\begin{aligned}
\frac{\partial n}{\partial t} + \frac{\partial (n V^i)}{\partial x^i} &= 0, \\
\frac{\partial (n W)}{\partial t} + \frac{\partial (n S^j)}{\partial x^j} + neV_k E^k &= n C_W, \\
\rho c_V \, \frac{\partial T_L}{\partial t} - \operatorname{div}\left[k(T_L)\nabla T_L\right] &= H, \\
\mathbf{E} &= -\nabla_x \Phi, \\
\epsilon \Delta \Phi_S &= -e(N_D - N_A - n),
\end{aligned}$$

along with the constitutive relations

$$\begin{aligned}
V_i &= D_{11}(W^{(0)}, T_L)\, \frac{\partial \log n}{\partial x_i} + D_{12}(W^{(0)}, T_L)\, \frac{\partial W}{\partial x_i} + D_{13}(W^{(0)}, T_L)\, \frac{\partial \phi}{\partial x_i} \\
&+ \frac{1}{\Delta}\left[\left(-c_{22}^{(e)} \frac{\hbar^2 \beta\, e}{12 m^*}\, \frac{\partial^2 \Phi}{\partial x_i \partial x_k} + c_{12}^{(e)} \frac{\hbar^2\, e}{24 m^{*2}} \left(\Delta \Phi\, \delta_{ik} + 5 \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\right)\right) \frac{\partial \log n}{\partial x_k} \right.\\
&\left. - c_{22}^{(e)} \frac{\partial}{\partial x_k}\left(\frac{\hbar^2 \beta\, e}{12 m^*}\, \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\right) + c_{12}^{(e)} \frac{\partial}{\partial x_k}\left(\frac{\hbar^2\, e}{24 m^{*2}}\left(\Delta \Phi\, \delta_{ik} + 5 \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\right)\right)\right],
\end{aligned}$$

$$\begin{aligned}
S_i &= D_{21}(W^{(0)}, T_L)\, \frac{\partial \log n}{\partial x_i} + D_{22}(W^{(0)}, T_L)\, \frac{\partial W}{\partial x_i} + D_{23}(W^{(0)}, T_L)\, \frac{\partial \phi}{\partial x_i} \\
&+ \frac{1}{\Delta}\left[\left(c_{21}^{(e)} \frac{\hbar^2 \beta\, e}{12 m^*}\, \frac{\partial^2 \Phi}{\partial x_i \partial x_k} - c_{11}^{(e)} \frac{\hbar^2\, e}{24 m^{*2}} \left(\Delta \Phi\, \delta_{ik} + 5 \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\right)\right) \frac{\partial \log n}{\partial x_k} \right.\\
&\left. + c_{21}^{(e)} \frac{\partial}{\partial x_k}\left(\frac{\hbar^2 \beta\, e}{12 m^*}\, \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\right) - c_{11}^{(e)} \frac{\partial}{\partial x_k}\left(\frac{\hbar^2\, e}{24 m^{*2}}\left(\Delta \Phi\, \delta_{ik} + 5 \frac{\partial^2 \Phi}{\partial x_i \partial x_k}\right)\right)\right].
\end{aligned}$$

If one introduces the equation of state

$$W^{(0)} = \frac{3}{2} k_B T, \tag{2.202}$$

the previous energy-transport model can be written using the electron density and temperature T, besides the electrical potential, as variables. However, it is crucial to remark that (2.202) is valid only in the parabolic band case (in analogy with the monatomic gas dynamics) and it is not justified in the non parabolic case, e.g. the Kane dispersion relation. In this latter case it is more appropriate to retain the energy W as fundamental variable.

References

1. Adler, M.: Accurate calculations of the forward drop and power dissipation in thyristors. IEEE Trans. Electron Dev. **ED-25**, 16–22 (1979)
2. Alì, G.: PDAE models of integrated circuits. Math. Comput. Mod. **51**, 915–926 (2010)
3. Alì, G., Bartel, A., Culpo, M., de Falco, C.: Analysis of a PDE thermal element model for electrothermal circuit simulation. In: Roos, J., Costa, L.R.J. (eds.) Proceedings of Scientific Computing in Electrical Engineering SCEE 2008, Espoo. Mathematics in Industry, vol. 14, pp. 273–280. Springer, Heidelberg (2010)
4. Alì, G., Bartel, A., Günther, M., Tischendorf, C.: Elliptic partial differential-algebraic multiphysics models in electrical network design. Math. Models Methods Appl. Sci. **13**(9), 1261–1278 (2003)
5. Alì, G., Bartel, A., Günther, M.: Parabolic differential-algebraic models in electric network design. SIAM J. MMS **4**(3), 813–838 (2005)
6. Alì, G., Mascali, G., Pulch, R.: Hyperbolic PDAEs for semiconductor devices coupled with circuits. In: Roos, J., Costa, L.R.J. (eds.) Proceedings of Scientific Computing in Electrical Engineering SCEE 2008, Espoo. Mathematics in Industry, vol. 14, pp. 305–312. Springer, Heidelberg (2010)
7. Anile, A., Mascali, G., Romano, V.: Recent developments in hydrodynamical modeling of semiconductors. In: Mathematical Problems in Semiconductor Physics. Lecture Notes in Mathematics, vol. 1832, pp. 1–56. Springer, Berlin/Heidelberg (2003)
8. Anile, A., Romano, V., Russo, G.: Extended hydrodynamical model of carrier transport in semiconductors. SIAM J. Appl. Math. **61**, 74–101 (2000)
9. Anile, A., Romano, V.: Non parabolic band transport in semiconductors: closure of the moment equations. Contin. Mech. Thermodyn. **11**, 307–325 (1999)
10. Barker, J., Ferry, D.: Self-scattering path-variable formulation of high-field, time-dependent, quantum kinetic equations for semiconductor transport in the finite-collision-duration regime. Phys. Rev. Lett. **42**, 1779–1781 (1979)
11. Bartel, A.: Partial differential-algebraic models in chip design – thermal and semiconductor problems. Ph.D. thesis, Bergische Universität Wuppertal (2003)
12. Bartel, A., Pulch, R.: A concept for classification of partial differential algebraic equations in nanoelectronics. In: Bonilla, L., Moscoso, M., Platero, G., Vega, J. (eds.) Progress in Industrial Mathematics at ECMI 2006. Mathematics in Industry, vol. 12, pp. 506–511. Springer, Berlin (2007)
13. Bedrosian, G.: A new method for coupling finite element field solutions with external circuits and kinematics. IEEE Trans. Magn. **29**(2), 1664–1668 (1993)

14. Bíró, O., Preis, K.: On the use of the magnetic vector potential in the finite-element analysis of three-dimensional eddy currents. IEEE Trans. Magn. **25**(4), 3145–3159 (1989)
15. Bossavit, A., Kettunen, L.: Yee-like schemes on staggered cellular grids: a synthesis between FIT and FEM approaches. IEEE Trans. Magn. **36**(4), 861–867 (2000)
16. Carrillo, J., Gamba, I., Majorana, A., Shu, C.W.: A Weno-solver for the transients of boltzmann-poisson system for semiconductor devices: performance and comparisons with Monte Carlo methods. J. Comput. Phys. **184**, 498–525 (2003)
17. Chryssafis, A., Love, W.: A computer-aided analysis of one dimensional thermal transient in n-p-n power transistors. Solid-State Electron. **22**, 249–256 (1978)
18. Clemens, M., Weiland, T.: Regularization of eddy-current formulations using discrete grad-div operators. IEEE Trans. Magn. **38**(2), 569–572 (2002)
19. Clemens, M.: Large systems of equations in a discrete electromagnetism: formulations and numerical algorithms. IEE Proc. Sci. Meas. Technol. **152**(2), 50–72 (2005)
20. Culpo, M.: Numerical algorithms for system level electro-thermal simulation. Ph.D. thesis, Bergische Universität Wuppertal (2009)
21. Culpo, M., de Falco, C.: Dynamical iteration schemes for coupled simulation in nanoelectronics. Proc. Appl. Math. Mech. **8**, 10,065–10,068 (2008)
22. Dreyer, W.: Maximisation of the entropy in non-equilibrium. J. Phys. A: Math. Gen. **20**, 6505–6517 (1987)
23. Dreyer, W., Struchtrup, H.: Heat pulse experiment revisited. Contin. Mech. Thermodyn. **5**, 3–50 (1993)
24. Estévez Schwarz, D., Tischendorf, C.: Structural analysis of electric circuits and consequences for MNA. Int. J. Circuit Theory Appl. **28**(2), 131–162 (2000)
25. de Falco, C., Culpo, M.: Dynamical iteration schemes for multiscale simulation in nanoelectronics. Proc. Appl. Math. Mech. **8**, 10,061–10,064 (2008)
26. Feldmann, U., Günther, M.: CAD-based electric-circuit modeling in industry I: mathematical structure and index of network equations. Surv. Math. Ind. **8**(2), 97–129 (1999)
27. Franz, A.F., Franz, G.A., Selberherr, S., Ringhofer, C., Markowich, P.: Finite boxes—a generalization of the finite-difference method suitable for semiconductor device simulation. IEEE Trans. Electron Devices **ED-30**, 1070–1082 (1983)
28. Fromlet, F., Markowich, P., Ringhofer, C.: A wignerfunction approach to phonon scattering. VLSI Des. **9**, 339–350 (1999)
29. Fukahori, K.: Computer simulation of monolithic circuit performance in the presence of electro-thermal interactions. Ph.D. thesis, University of California, Berkeley (1977)
30. Galler, M., Schürrer, F.: A deterministic solution method for the coupled system of transport equations for the electrons and phonons in polar semiconductors. J. Phys. A: Math. Gen. **37**, 1479–1497 (2004)
31. Gardner, C.: The quantum hydrodynamic model for semiconductors devices. SIAM J. Appl. Math. **54**, 409–427 (1994)
32. Gaur, S., Navon, D.: Two-dimensional carrier flow in a transistor structure under nonisothermal conditions. IEEE Trans. Electron Devices **ED-23**, 50–57 (1976)
33. De Gersem, H., Munteanu, I., Weiland, T.: Construction of differential material matrices for the orthogonal finite-integration technique with nonlinear materials. IEEE Trans. Magn. **44**(6), 710–713 (2008)
34. Glowinski, R., He, J., Lozinski, A., Rappaz, J., Wagner, J.: Finite element approximation of multi-scale elliptic problems using patches of elements. Numer. Math. **101**(4), 663–687 (2005)
35. Griepentrog, E., März, R.: Differential-Algebraic Equations and Their Numerical Treatment. Teubner, Leipzig (1986)
36. Günther, M.: A joint DAE/PDE model for interconnected electrical networks. Math. Comput. Model. Dyn. Syst. **6**, 114–128 (2000)
37. Günther, M., Wagner, Y.: Index concepts for linear mixed systems of differential-algebraic and hyperbolic-type equations. SIAM J. Sci. Comput. **22**(5), 1610–1629 (2000)
38. Haas, H., Schmellebeck, F.: Approximation of nonlinear anisotropic magnetization characteristics. IEEE Trans. Magn. **28**(2), 1255–1258 (1992)

39. Haus, H.A., Melcher, J.R.: Electromagnetic Fields and Energy. Prentice Hall, Englewood Cliffs (1989)
40. Ho, C.W., Ruehli, A.E., Brennan, P.A.: The modified nodal approach to network analysis. IEEE Trans. Circuits Syst. CAS **22**, 505–509 (1975)
41. Janes, E.: Information theory and statistical mechanics. Phys. Rev. **106**, 620–630 (1957)
42. Kosaku, Y.: Functional Analysis. Springer, Berlin/New York (1980)
43. Lions, J.L., Magenes, E.: Problèmes aux limites non Homogènes et Applications, vol. 1. Dunod, Paris (1968)
44. Lucht, W., Strehmel, K., Eichler-Liebenow, C.: Indexes and special discretization methods for linear partial differential algebraic equations. BIT **39**(3), 484–512 (1999)
45. Majorana, A.: Space homogeneous solutions of the Boltzmann equation describing electron-phonon interactions in semiconductors. Transp. Theory Stat. Phys. **20**, 261–279 (1991)
46. Majorana, A.: Conservation laws from the Boltzmann equation describing electron-phonon interactions in semiconductors. Transp. Theory Stat. Phys. **22**, 849–859 (1993)
47. Majorana, A.: Equilibrium solutions of the non-linear Boltzmann equation for an electron gas in a semiconductors. Il Nuovo Cimento **108B**, 871–877 (1993)
48. Marrocco, A., Anile, A., Romano, V., Sellier, J.: 2d numerical simulation of the mep energy-transport model with a mixed finite elements scheme. J. Comput. Electron. **4**, 231–259 (2005)
49. Mascali, G., Romano, V.: Hydrodynamical model of charge transport in GAAs based on the maximum entropy principle. Contin. Mech. Thermodyn. **14**, 405–423 (2002)
50. McCalla, W.J.: Fundamentals of computer aided circuit simulation. Kluwer Academic, Boston (1988)
51. Pulch, R., Günther, M., Knorr, S.: Multirate partial differential algebraic equations for simulating radio frequency signals. Eur. J. Appl. Math. **18**, 709–743 (2007)
52. Quarteroni, A., Valli, A.: Numerical Approximation of Partial Differential Equations. Computational Mathematics. Springer, Berlin/New York (1997)
53. Romano, V.: Non parabolic band transport in semiconductors: closure of the production terms in the moment equations. Contin. Mech. Thermodyn. **12**, 31–51 (2000)
54. Romano, V.: Non parabolic band hydrodynamical model of silicon semiconductors and simulation of electron devices. Math. Methods Appl. Sci. **24**, 439–471 (2001)
55. Romano, V.: 2d numerical simulation of the mep energy-transport model with a finite difference scheme. J. Comput. Phys. **221**, 439–468 (2007)
56. Romano, V., Zwierz, M.: Electron-phonon hydrodynamical model for semiconductors. ZAMP **61**, 1111–1131 (2010)
57. Romano, V.: Quantum corrections to the semiclassical hydrodynamical model of semiconductors based on the maximum entropy principle. J. Math. Phys. **48**, 123504 (2007)
58. Romano, V., Scordia, C.: Simulations of an electron-phonon hydrodynamical model based on the maximum entropy principle. In: Roos, J., Costa, L.R.J. (eds.) Proceedings of Scientific Computing in Electrical Engineering SCEE 2008, Espoo. Mathematics in Industry, vol. 14, pp. 289–296. Springer, Heidelberg (2010)
59. Schöps, S., Bartel, A., de Gersem, H., Günther, M.: DAE-index and convergence analysis of lumped electric circuits refined by 3-D magnetoquasistatic conductor models. In: Roos, J., Costa, L.R.J. (eds.) Proceedings of Scientific Computing in Electrical Engineering SCEE 2008, Espoo. Mathematics in Industry, vol. 14, pp. 341–348. Springer, Heidelberg (2010)
60. Selberherr, S.: Analysis and Simulation of Semiconductor Devices. Springer, Wien/New York (1984)
61. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423, 623–656 (1948)
62. Wachutka, G.: Rigorous thermodynamic treatment of heat generation and conduction in semiconductor device modeling. IEEE Trans. Comput. Aided Des. **9**, 1141–1149 (1990)
63. Weiland, T.: A discretization model for the solution of Maxwell's equations for six-component fields. Int. J. Electron. Commun. **31**, 116–120 (1977)

64. Wigner, E.: On the quantum correction for thermodynamic equilibrium. Phys. Rev. **40**, 749–759 (1932)
65. Wu, N.: The Maximum Entropy Method. Springer, New York (1997)
66. Yee, K.S.: Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Trans. Antennas Propag. **14**(3), 302–307 (1966)

Chapter 3
Simulation of Coupled PDAEs: Dynamic Iteration and Multirate Simulation

Giuseppe Alì, Andreas Bartel, Michael Günther, Vittorio Romano, and Sebastian Schöps

Abstract This chapter investigates the error transport in dynamic iteration schemes for coupled DAE systems. The essential theory is developed in detail. Then the results are applied to various coupled systems stemming from applications in electrical engineering.

3.1 Aim and Outline

In practice, we often have to deal with multiphysical descriptions of mathematical models and as well with systems which exhibit widely separated time scales. A common approach for multiphysical systems is the application of dynamic iteration (or co-simulation), which allows to treat each subsystem with a dedicated solver, and also an according discretization. Furthermore, so-called multirate techniques can be applied to specifically exploit different time scales.

G. Alì
Department of Physics, University of Calabria via Pietro Bucci 30/B, 87036 Arcavacata di Rende, Cosenza, Italy
e-mail: giuseppe.ali@unical.it

A. Bartel (✉) • M. Günther
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Gaußstraße 20, D-42119 Wuppertal, Germany
e-mail: bartel@math.uni-wuppertal.de; guenther@math.uni-wuppertal.de

S. Schöps
TU Darmstadt, Graduate School of Excellence Computational Engineering, Dolivostraße 15, 64293 Darmstadt, Germany
e-mail: schoeps@gsc.tu-darmstadt.de

V. Romano
Università di Catania, Dipartimento di Matematica e informatica, Viale A. Doria no, 95125 Catania, Italy
e-mail: romano@dmi.unict.it

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_3

To reflect this, the aim of this chapter is twofold. First we address dynamic iteration of spatially discretized PDAE systems, which are in fact coupled DAE systems. We demonstrate the crucial differences between coupled DAE and coupled ODE systems by investigating the splitting error of these coupled systems theoretically. Then we apply the obtained knowledge to coupled systems from Chap. 2. Secondly, a multirate strategy is discussed and studied numerically.

To this end, this chapter is organized as follows. It starts with the detailed theory of dynamic iteration schemes for coupled DAEs. First we consider a single window and proof an error recursion for any investigated window. Then we treat multiple windows and generalize the results. In the following section, we apply our results to some of the DAE models introduced in Chap. 2: refined network models, electric networks and Maxwell's magnetostatic equations. Finally, a multirate method for the coupled simulation of thermal effects in silicon devices is investigated.

3.2 Theory of Dynamic Iteration Schemes for Coupled DAEs

Here we address the time-domain solution of PDAEs by means of dynamical iteration schemes. To explain the basic concept, let us suppose that we want to solve an initial value problem for a system of PDAEs, on a time interval $[0, t_e]$. To this end, the time interval $[0, t_e]$ is split in windows $[t_n, t_{n+1}]$ with so-called synchronization points t_n, which satisfy: $0 = t_0 < t_1 < \cdots < t_N = t_e$. The windows are treated sequentially and in each window the subsystems are solved iteratively. Mathematically speaking, this leads to apply a dynamic iteration scheme.

Coupled systems as our PDAEs, see Chap. 2, can be treated with coupled simulators, each designed and tailored to the respective subsystem's structure. This is called simulator-coupling, co-simulation or distributed (time-)integration. Compared to monolithic approaches, where the overall system is treated by any standard integration the distributed computation offers potential w.r.t. parallelization and incorporates adapted step sizes and orders to every subsystem automatically.

Although we have in mind applications to PDAEs, we will develop the theory of dynamic iteration schemes for DAEs. For practical applications, all the results presented in this Chapter can be extended to PDAE after performing suitable spatial discretizations. A detailed example of this approach is given for PDAEs arising in refined network modeling.

Iteration schemes were first applied to coupled ODE systems, including multirate, multi-order, multi-method and dynamic iteration. For the latter, which is our focus, convergence is unconditional (see [10]) if the windowing technique is applied. However, the situation changes, when this methods are applied to DAEs. Here instabilities may occur and solutions can explode even if a windowing technique is in use. Here convergence, that is, contraction of the corresponding fixed point operator, can be guaranteed by fulfilling additional stability constraints. This dates back to Lelarasmee [24] and was applied for single window convergence [3, 22] and specially coupled systems for multiple windows in [1, 4]. We note that the stability restrictions where also encountered in the numerical analysis of DAEs, see [16, 21] and [1, 23].

Here we follow [4] with some more details to derive a general representation of the error recursion for coupled systems. The preceding steps, e.g. for convergence result, are as in [1]. Furthermore we aim at extracting the underlying principle: algebraic to algebraic coupling is to be excluded or damped.

3.2.1 Description of Coupled Systems

After applying a suitable space discretization to the PDAE problems discussed in the first chapter, we are faced with the following simulation problem: solve an initial-value problems of semi-explicit differential-algebraic equations

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{z}), \tag{3.1a}$$

$$0 = \mathbf{g}(\mathbf{y}, \mathbf{z}), \tag{3.1b}$$

where the dot denotes differentiation with respect to time. In this formulation we do not distinguish between different subsystems, but all subsystems are comprised within one system. As we will see, this is enough to treat dynamic iteration schemes. It is specially well-applicable for linear PDE-parts, where space and time discretization can be easily separated. Also a non-autonomous system can be casted in this form, by introducing an additional equation: $\dot{t} = 1$. We assume that this problem, equipped with initial values

$$\mathbf{y}(0) = \mathbf{y}_0\ , \quad \mathbf{z}(0) = \mathbf{z}_0, \tag{3.2}$$

has a unique solution $\mathbf{y} : [0, t_e] \to \mathbb{R}^{n_y}$, $\mathbf{z} : [0, t_e] \to \mathbb{R}^{n_z}$ on the finite time interval $[0, t_e]$. In a neighborhood of this solution the functions $\mathbf{f}$ and $\mathbf{g}$ are supposed to be sufficiently often differentiable. Furthermore, it is supposed that

$$\text{the Jacobian } \partial\mathbf{g}/\partial\mathbf{z} \text{ is non-singular,} \tag{3.3}$$

in the neighborhood of the solution. Hence system (3.1) has index-1. Moreover, the initial values (3.2) have to be consistent, that is for our semi-explicit index-1 system (3.1), the explicit algebraic constraint (3.1b) is fulfilled for the initial data.

Next we discuss the representation of coupled systems. In multiphysics problems, system (3.1) is often directly given as a coupled system of r DAE subsystems

$$\dot{\mathbf{y}}_i = \mathbf{f}_i(\mathbf{y}, \mathbf{z}), \tag{3.4a}$$

$$0 = \mathbf{g}_i(\mathbf{y}, \mathbf{z}) \tag{3.4b}$$

for $i = 1, \ldots, r$, with $\mathbf{y}^\top = (\mathbf{y}_1^\top, \ldots, \mathbf{y}_r^\top)$, $\mathbf{z}^\top = (\mathbf{z}_1^\top, \ldots, \mathbf{z}_r^\top)$, $\mathbf{f}^\top = (\mathbf{f}_1^\top, \ldots, \mathbf{f}_r^\top)$, $\mathbf{g}^\top = (\mathbf{g}_1^\top, \ldots, \mathbf{g}_r^\top)$. In addition to the index-one assumption (3.3) for the whole system (3.1), we now assume that

$$\partial \mathbf{g}_i / \partial \mathbf{z}_i \text{ is non-singular for all } i = 1, \ldots, r, \tag{3.5}$$

so that the equations $\mathbf{g}_i(\mathbf{y}, \mathbf{z}) = 0$ are locally uniquely solvable with respect to $\mathbf{z}_i$, with other words: system (3.4) defines an index-1 system for unknown functions $\mathbf{y}_i, \mathbf{z}_i$ assuming that all other variables $\mathbf{y}_j, \mathbf{z}_j$ $(j \neq i)$ are given as time-dependent functions.

Sometimes system (3.1) may be given as r coupled ODE systems linked to only one algebraic equation:

$$\dot{\mathbf{y}}_i = \mathbf{f}_i(\mathbf{y}, \mathbf{z}), \tag{3.6a}$$

$$0 = \mathbf{g}(\mathbf{y}, \mathbf{z}), \tag{3.6b}$$

for $i = 1, \ldots, r$. The index-1 assumption now again reads as in (3.3), that is, we assume that $\partial \mathbf{g}/\partial \mathbf{z}$ is non-singular in a neighborhood of the solution.

Sometimes a separation in subsystems is not a priori fixed by a simple partition (e.g. (3.6)). This leads to the following notation, where some quantities are assigned to several subsystems.

Overlapping modeling The structure (3.6) gives more freedom in a dynamic iteration scheme by applying appropriate overlapping strategies [2]. For such a strategy, the system is replaced by a number of overlapping subsystems, defined by means of splitting matrices. As splitting matrices we introduce $\mathbf{P}_i \in \mathbb{R}^{n_z \times l_i}$ with $1 \leq l_i \leq n_z$ and $\operatorname{rank}(P_i) = l_i$ for $i = 1, \ldots, r$, such that the matrix

$$(\mathbf{P}_1 \ldots \mathbf{P}_r) \in \mathbb{R}^{n_z \times (\sum_i l_i)} \text{ has full rank } n_z \tag{3.7}$$

(thus we implicitly require $\sum_i l_i \geq n_z$). In this way, arbitrary parts $\mathbf{P}_i^\top \mathbf{g}$ of the algebraic equation (3.6b) can be extracted, since it holds:

$$(\mathbf{P}_1, \ldots, \mathbf{P}_r)^\top \mathbf{g} = 0 \quad \text{if and only if} \quad \mathbf{g} = 0.$$

Next, we assign the extracted components to the i-th ODE subsystem to define r overlapping DAE systems:

$$\dot{\mathbf{y}}_i = \mathbf{f}_i(\mathbf{y}, \mathbf{w}_i), \tag{3.8a}$$

$$0 = \mathbf{P}_i^\top \mathbf{g}(\mathbf{y}, \mathbf{w}_i), \tag{3.8b}$$

substituting $\mathbf{z}$ by $\mathbf{w}_i$ (for $i = 1, \ldots, r$). Also $\mathbf{z}$ is split into further components $\bar{\mathbf{z}}_i := \mathbf{P}_i^\top \mathbf{z}$, such that it holds

$$\mathbf{w}_i = \mathbf{z} = (\mathbf{I} - \mathbf{P}_i \mathbf{P}_i^\top)\mathbf{z} + \mathbf{P}_i \bar{\mathbf{z}}_i. \tag{3.9}$$

This splitting is crucial for any modular time integration to come. Adding the coupling equation (3.9) to the rth system, we obtain in fact:

$$\dot{\mathbf{y}}_i = \tilde{\mathbf{f}}_i(\mathbf{y}, \mathbf{z}, \bar{\mathbf{z}}_i), \tag{3.10a}$$

$$0 = \tilde{\mathbf{g}}_i(\mathbf{y}, \mathbf{z}, \bar{\mathbf{z}}_i), \tag{3.10b}$$

for $i = 1, \ldots, r$, with

$$\begin{aligned}
\tilde{\mathbf{f}}_i(\mathbf{y}, \mathbf{z}, \bar{\mathbf{z}}_i) &:= \mathbf{f}_i(\mathbf{y}, (\mathbf{I} - \mathbf{P}_i\mathbf{P}_i^\top)\mathbf{z} + \mathbf{P}_i\bar{\mathbf{z}}_i), \quad i = 1, \ldots, r,\\
\tilde{\mathbf{g}}_i(\mathbf{y}, \mathbf{z}, \bar{\mathbf{z}}_i) &:= \mathbf{P}_i^\top \mathbf{g}(\mathbf{y}, (\mathbf{I} - \mathbf{P}_i\mathbf{P}_i^\top)\mathbf{z} + \mathbf{P}_i\bar{\mathbf{z}}_i) \quad i = 1, \ldots, r-1,\\
\tilde{\mathbf{g}}_r(\mathbf{y}, \mathbf{z}, \bar{\mathbf{z}}) &:= \begin{pmatrix} \mathbf{P}_r^\top \mathbf{g}(\mathbf{y}, (\mathbf{I} - \mathbf{P}_r\mathbf{P}_r^\top)\mathbf{z} + \mathbf{P}_r\bar{\mathbf{z}}_r) \\ \mathbf{z} - \left[(\mathbf{I} - \sum_{j=1}^{r} \mathbf{P}_j\mathbf{P}_j^\top)\mathbf{z} + \sum_{j=1}^{r} \mathbf{P}_j\bar{\mathbf{z}}_j\right] \end{pmatrix}.
\end{aligned}$$

If the original system (3.6) has index-1, then also system (3.10) has index-1. In fact, the index-1 conditions for system (3.10) are:

$$\mathbf{P}_i^\top(\partial\mathbf{g}/\partial\mathbf{z})\mathbf{P}_i \text{ regular,}$$

$$\sum_{j=1}^{r} \mathbf{P}_j\mathbf{P}_j^\top \text{ regular,}$$

which are ensured by the index-1 condition (3.3), and by the definition of our matrices $\mathbf{P}_j$, which satisfy condition (3.7).

Lastly, we notice: (a) according to our system (3.6), we have only overlapping in the algebraic system; of course, more general situations are conceivable; (b) the case of additional coupling equations can be also retrieved within the above discussed case.

Next, we discuss several types of iteration schemes, which we can identify with splitting functions.

3.2.2 Iteration Schemes for Coupled DAE Systems

The idea of our dynamic iteration schemes is now to work directly on the splitting structure of system (3.1) given by either (3.4) or (3.6) to exploit the varying properties of the subsystems via multirate and multimethod approaches.

Before going into the details of exploiting the special structure, we define a generic dynamic iteration scheme in the following. In a first step we split the whole integration interval $[0, t_e]$ into windows $[t_n, t_{n+1}] \subset [0, t_e]$ ($n = 0, 1, \ldots, N-1$ with $t_0 = 0$ and $t_N = t_e$), of size $H_n := t_{n+1} - t_n$. As already mentioned, this windowing technique guarantees convergence in the case of purely coupled ODE systems and for DAE systems additional stability restrictions to be discussed play an important role (for convergence and fast numerical computation of solutions).

Let us now consider a window $[t_n, t_{n+1}]$ and suppose that the numerical solution

$$(\tilde{\mathbf{y}}, \tilde{\mathbf{z}})^\top : [0, t_e] \to \mathbb{R}^{n_y} \times \mathbb{R}^{n_z}$$

has already been computed for $t \in [0, t_n]$. To get a numerical approximation in the next window $[t_n, t_{n+1}]$,

$$\tilde{\mathbf{y}}|_{(t_n, t_{n+1}]}, \qquad \tilde{\mathbf{z}}|_{(t_n, t_{n+1}]},$$

we proceed as follows:

- *Extrapolation step*: the iteration starts with

$$\begin{pmatrix} \tilde{\mathbf{y}}_n^{(0)} \\ \tilde{\mathbf{z}}_n^{(0)} \end{pmatrix} := \Phi_n \begin{pmatrix} \tilde{\mathbf{y}}|_{(t_{n-1}, t_n]} \\ \tilde{\mathbf{z}}|_{(t_{n-1}, t_n]} \end{pmatrix} \qquad \text{with } \Phi_n = \begin{pmatrix} \Phi_{\mathbf{y},n} \\ \Phi_{\mathbf{z},n} \end{pmatrix}, \tag{3.11}$$

where $\Phi_n : \bar{C}_{n-1}^{1,0} \to C_n^{1,0}$ denotes an operator that extrapolates $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ continuously from $(t_{n-1}, t_n]$ to $[t_n, t_{n+1}]$ with corresponding spaces

$$\begin{aligned} \bar{C}_n^{1,0} &:= \left\{ (\mathbf{y}, \mathbf{z})|_{(t_n, t_{n+1}]} : (\mathbf{y}, \mathbf{z}) \in C_n^{1,0} \right\}, \\ C_n^{1,0} &:= C^1([t_n, t_{n+1}], \mathbb{R}^{n_y}) \times C([t_n, t_{n+1}], \mathbb{R}^{n_z}). \end{aligned}$$

The most simple initial guesses are constant functions

$$\tilde{\mathbf{y}}_n^{(0)}(t) = \tilde{\mathbf{y}}(t_n), \quad \mathbf{z}_n^{(0)}(t) = \tilde{\mathbf{z}}(t_n) \quad (\text{f.a. } t \in [t_n, t_{n+1}])$$

which results in approximation errors proportional to the window size H_n. Approximations of higher order may be obtained by using higher degree polynomials. In any case, these extrapolation operators satisfy uniform Lipschitz conditions independent of the window size (see [1]).

- *Iteration step*: the k-th iteration step in the dynamic iteration scheme (with $k = 1, \ldots, k_n$) defines a mapping

$$\begin{pmatrix} \tilde{\mathbf{y}}_n^{(k-1)} \\ \tilde{\mathbf{z}}_n^{(k-1)} \end{pmatrix} \to \begin{pmatrix} \tilde{\mathbf{y}}_n^{(k)} \\ \tilde{\mathbf{z}}_n^{(k)} \end{pmatrix} := \Psi_n \begin{pmatrix} \tilde{\mathbf{y}}_n^{(k-1)} \\ \tilde{\mathbf{z}}_n^{(k-1)} \end{pmatrix} \qquad \text{with } \Psi_n = \begin{pmatrix} \Psi_{\mathbf{y},n} \\ \Psi_{\mathbf{z},n} \end{pmatrix}, \tag{3.12}$$

$\Psi_n : C_n^{1,0} \to C_n^{1,0}$. Here we assume k_n to denote the finite number of iterations to be performed in the n-th window ($[t_n, t_{n+1}]$). Regarding the general setting (3.1), the iteration operator Ψ_n is implicitly defined via splitting functions $\mathbf{F}$ and $\mathbf{G}$ by solving the initial value problem

$$\dot{\tilde{\mathbf{y}}}_n^{(k)} = \mathbf{F}(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) \tag{3.13a}$$

$$0 = \mathbf{G}(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) \tag{3.13b}$$

with initial value

$$\tilde{\mathbf{y}}_n^{(k)}(t_n) = \tilde{\mathbf{y}}_n^{(k-1)}(t_n). \tag{3.13c}$$

The splitting functions $\mathbf{F}$ and $\mathbf{G}$ can be chosen as arbitrarily smooth functions provided that they are related to the right-hand-sides $\mathbf{f}$ and $\mathbf{g}$ of the DAE system (3.1) by the compatibility conditions

$$\mathbf{F}(\mathbf{y}, \mathbf{y}, \mathbf{z}, \mathbf{z}) = \mathbf{f}(\mathbf{y}, \mathbf{z}), \qquad \mathbf{G}(\mathbf{y}, \mathbf{y}, \mathbf{z}, \mathbf{z}) = \mathbf{g}(\mathbf{y}, \mathbf{z}). \tag{3.14}$$

As $\mathbf{f}$, $\mathbf{g}$ are assumed to be sufficiently often differentiable, this is also assumed for $\mathbf{F}$ and $\mathbf{G}$.

Remark 3.4 Notice, that the analytic solution $(\mathbf{y}, \mathbf{z})$ is a fixed-point of the iteration operator Ψ_n due to the compatibility conditions (3.14).

With these notations the dynamic iteration step for window $[t_n, t_{n+1}]$ may be written as composition of the above introduced operators:

$$\begin{pmatrix} \tilde{\mathbf{y}}|_{(t_n, t_{n+1}]} \\ \tilde{\mathbf{z}}|_{(t_n, t_{n+1}]} \end{pmatrix} = (\Psi_n^{k_n} \circ \Phi_n)\left(\begin{pmatrix} \tilde{\mathbf{y}}|_{(t_{n-1}, t_n]} \\ \tilde{\mathbf{z}}|_{(t_{n-1}, t_n]} \end{pmatrix}\right). \tag{3.15}$$

We now come back to the question how to exploit the given structure of the coupled DAE system. If the DAE system is given in partitioned form (3.4), we are looking for numerical approximations

$$\tilde{\mathbf{y}}_n = (\mathbf{y}_{1,n}, \ldots, \mathbf{y}_{r,n})^\top, \qquad \tilde{\mathbf{z}}_n = (\mathbf{z}_{1,n}, \ldots, \mathbf{z}_{r,n})^\top$$

in split form. Now the iteration operator Ψ_n should reflect this partitioning. Instead of (3.13), Ψ_n is now implicitly defined by the r initial-value problems

$$\dot{\tilde{\mathbf{y}}}_{i,n}^{(k)} = \mathbf{F}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}), \tag{3.16a}$$

$$0 = \mathbf{G}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}), \tag{3.16b}$$

for $i = 1, \ldots, r$, with initial value

$$\tilde{\mathbf{y}}_{i,n}^{(k)}(t_n) = \tilde{\mathbf{y}}_{i,n}^{(k-1)}(t_n). \tag{3.16c}$$

Again, all splitting functions $\mathbf{F}_i$ and $\mathbf{G}_i$ are related to the right-hand-sides $\mathbf{f}_i$ and $\mathbf{g}_i$ of the DAE system (3.4) by the compatibility conditions

$$\mathbf{F}_i(\mathbf{y},\mathbf{y},\mathbf{z},\mathbf{z}) = \mathbf{f}_i(\mathbf{y},\mathbf{z}), \qquad \mathbf{G}_i(\mathbf{y},\mathbf{y},\mathbf{z},\mathbf{z}) = \mathbf{g}_i(\mathbf{y},\mathbf{z}).$$

And it holds:

$$\mathbf{F}^\top = (\mathbf{F}_1^\top, \ldots, \mathbf{F}_r^\top) \quad \text{and} \quad \mathbf{G}^\top = (\mathbf{G}_1^\top, \ldots, \mathbf{G}_r^\top).$$

In the notation of splitting functions, the following important classes of dynamic iterations schemes for the coupled system (3.4) read as:

$$\mathbf{F}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) = \mathbf{f}_i(\mathbf{Y}_{i,n}^{(k)}, \mathbf{Z}_{i,n}^{(k)}), \tag{3.17a}$$

$$\mathbf{G}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) = \mathbf{g}_i(\mathbf{Y}_{i,n}^{(k)}, \mathbf{Z}_{i,n}^{(k)}), \tag{3.17b}$$

for $i = 1, \ldots, r$, with:

- *Picard iteration*:

$$\mathbf{Y}_{i,n}^{(k)} = \tilde{\mathbf{y}}_n^{(k-1)},$$
$$\mathbf{Z}_{i,n}^{(k)} = \tilde{\mathbf{z}}_n^{(k-1)},$$

- *Jacobi iteration:*

$$\mathbf{Y}_{i,n}^{(k)} = (\tilde{\mathbf{y}}_{1,n}^{(k-1)}, \ldots, \tilde{\mathbf{y}}_{i-1,n}^{(k-1)}, \tilde{\mathbf{y}}_{i,n}^{(k)}, \tilde{\mathbf{y}}_{i+1,n}^{(k-1)}, \ldots, \tilde{\mathbf{y}}_{r,n}^{(k-1)})^\top,$$
$$\mathbf{Z}_{i,n}^{(k)} = (\tilde{\mathbf{z}}_{1,n}^{(k-1)}, \ldots, \tilde{\mathbf{z}}_{i-1,n}^{(k-1)}, \tilde{\mathbf{z}}_{i,n}^{(k)}, \tilde{\mathbf{z}}_{i+1,n}^{(k-1)}, \ldots, \tilde{\mathbf{z}}_{r,n}^{(k-1)})^\top,$$

- *Gauss-Seidel iteration:*

$$\mathbf{Y}_{i,n}^{(k)} = (\tilde{\mathbf{y}}_{1,n}^{(k)}, \ldots, \tilde{\mathbf{y}}_{i,n}^{(k)}, \tilde{\mathbf{y}}_{i+1,n}^{(k-1)}, \ldots, \tilde{\mathbf{y}}_{r,n}^{(k-1)})^\top,$$
$$\mathbf{Z}_{i,n}^{(k)} = (\tilde{\mathbf{z}}_{1,n}^{(k)}, \ldots, \tilde{\mathbf{z}}_{i,n}^{(k)}, \tilde{\mathbf{z}}_{i+1,n}^{(k-1)}, \ldots, \tilde{\mathbf{z}}_{r,n}^{(k-1)})^\top.$$

These techniques can be applied to the system derived from overlapping (3.8). The involved multiple computation of certain quantities, enables higher flexibility with respect to stability, as we will see. In the following we discuss a variant of the Gauss-Seidel scheme.

Overlapping technique For a DAE system given in form (3.6) (with an overall algebraic equation), overlapping was introduced in (3.10) with dynamic iteration as the method of choice [2]. For a Gauss-Seidel-like scheme, this overlapping modular time integration reads as follows. First, each subsystem

$$\dot{\mathbf{y}}_i = \mathbf{f}_i(\mathbf{y}_1, \ldots, \mathbf{y}_r, \mathbf{W}_i), \tag{3.18a}$$

$$0 = \mathbf{P}_i^\top \mathbf{g}(\mathbf{y}_1, \ldots, \mathbf{y}_r, \mathbf{W}_i), \tag{3.18b}$$

for $i = 1, \ldots, r$, is equipped with the relation

$$\mathbf{W}_i = (\mathbf{I} - \mathbf{P}_i \mathbf{P}_i^\top)\mathbf{z}_n^{(k-1)} + \mathbf{P}_i \mathbf{P}_i^\top \mathbf{Z}_i^{(k)},$$

introducing an additional stage vector $\mathbf{Z}_i^{(k)}$, which serves as an intermediate approximation for components of $\mathbf{z}$. Translated into splitting functions (and adding the Gauss-Seidel scheme), this leads to system (3.16), with

$$\begin{aligned}
&\mathbf{F}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) = \mathbf{f}_i(\mathbf{Y}_{i,n}^{(k)}, \mathbf{W}_{i,n}^{(k)}), \quad i = 1, \ldots, r,\\
&\mathbf{G}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) = \mathbf{P}_i^\top \mathbf{g}_i(\mathbf{Y}_{i,n}^{(k)}, \mathbf{W}_{i,n}^{(k)}), \quad i = 1, \ldots, r-1,\\
&\mathbf{G}_r(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)})\\
&\quad = \begin{pmatrix} \mathbf{P}_r^\top \mathbf{g}_r(\mathbf{Y}_{i,n}^{(k)}, \mathbf{W}_{i,n}^{(k)}) \\ \tilde{\mathbf{z}}_n^{(k)} - \left(\mathbf{I} - \sum_{j=1}^r \mathbf{A}_j \mathbf{P}_j \mathbf{P}_j^\top\right) \tilde{\mathbf{z}}_n^{(k-1)} - \sum_{j=1}^r \mathbf{A}_j \mathbf{P}_j \mathbf{Z}_j^{(k)} \end{pmatrix},
\end{aligned}$$

where we have posed

$$\begin{aligned}
\mathbf{Y}_{i,n}^{(k)} &= (\tilde{\mathbf{y}}_{1,n}^{(k)}, \ldots, \tilde{\mathbf{y}}_{i,n}^{(k)}, \tilde{\mathbf{y}}_{i+1,n}^{(k-1)}, \ldots, \tilde{\mathbf{y}}_{r,n}^{(k-1)})^\top,\\
\mathbf{W}_{i,n}^{(k)} &= (\mathbf{I} - \mathbf{P}_i \mathbf{P}_i^\top)\tilde{\mathbf{z}}_n^{(k-1)} + \mathbf{P}_i \mathbf{P}_i^\top \mathbf{Z}_i^{(k)}.
\end{aligned}$$

Thereby in the last algebraic constraint, we have introduced additional matrices

$$\mathbf{A}_j \in \mathbb{R}^{n_z \times n_z} \qquad (j = 1, \ldots, r)$$

as free parameters for enforcing better stability properties. Notice that the special choice $\mathbf{P}_i = e_i^\top$, $\mathbf{A}_i = \mathbf{I}$ $(i = 1, \ldots, r)$ leads back to system (3.4), solved by the Jacobi-like iteration scheme, while regarding the algebraic part only. Last, the index-1 hypothesis, leads to the assumption that

$$\text{the matrix } \sum_{j=1}^r \mathbf{A}_j \mathbf{P}_j \mathbf{P}_j^\top \text{ is regular.} \tag{3.19}$$

This is the case, if $(\mathbf{A}_1 \mathbf{P}_1, \ldots, \mathbf{A}_r \mathbf{P}_r)$ has full rank.

The discussed method corresponds to a dynamic iteration for the overlapping DAE systems (3.10), with slight generalization with respect to the free parameter matrices.

Applying Gauss-Seidel, Jacobi or Picard like dynamic iteration schemes, as well as overlapping modular time integration, to coupled ODEs convergence may always be achieved using sufficiently small window sizes. In the application to coupled differential-algebraic equations, however, two additional contractivity conditions have to be satisfied to achieve

- Convergence within one window, and
- A stable error propagation in the algebraic components z from one window to another.

This will be the topic of the next sections, where we generalize corresponding results of [1] obtained for a special coupled system to the general case of system (3.1).

3.2.3 *Convergence and Stability*

In the following we address the convergence of the above defined dynamic iteration schemes. That is, we want to deal with (a) the error within one window, and (b) the transport and amplification of error from window to window. To this end, we introduce the related error notations. First, we derive the error recursions for the error within one window, and prove convergence within each single window under certain stability requirements. Secondly, we treat a finite number of windows and prove the convergence under the related requirements.

We consider an analytic error recursion, thus error due to time integration are not considered explicitly, here. We follow basically [1], but put everything in a more general context as already started in[3]. Thus in fact, only Lemma 3.1 and the exact definition of α differ from the preceding work. Here we adopt a more general viewpoint, to reveal the most prominent structural properties.

3.2.3.1 Error Recursion

Following standard procedures in error analysis, e.g. [21], we define the global error $\boldsymbol{\epsilon}_{\mathbf{y},n}(t)$, $\boldsymbol{\epsilon}_{\mathbf{z},n}(t)$ on the n-th time window ($t \in [t_n, t_{n+1}]$) as the difference of the numerical approximation $\tilde{\mathbf{y}}(t)$, $\tilde{\mathbf{z}}(t)$ and the exact solutions $\mathbf{y}(t)$, $\mathbf{z}(t)$, where the unknowns and hence the errors are split into algebraic and differential components:

$$\begin{pmatrix}\boldsymbol{\epsilon}_{\mathbf{y},n}\\ \boldsymbol{\epsilon}_{\mathbf{z},n}\end{pmatrix} := \begin{pmatrix}(\tilde{\mathbf{y}}-\mathbf{y})\,|_{(t_n,t_{n+1}]}\\ (\tilde{\mathbf{z}}-\mathbf{z})\,|_{(t_n,t_{n+1}]}\end{pmatrix} = \left(\Psi_n^{k_n}\circ\Phi_n\right)\begin{pmatrix}\tilde{\mathbf{y}}|_{(t_{n-1},t_n]}\\ \tilde{\mathbf{z}}|_{(t_{n-1},t_n]}\end{pmatrix} - \begin{pmatrix}\mathbf{y}|_{[t_n,t_{n+1}]}\\ \mathbf{z}|_{[t_n,t_{n+1}]}\end{pmatrix}.$$

Here the numerical approximation on the current time window is given by an approximation on the previous time window, which is extrapolated by Φ_n and then k_n-times iterated by the dynamic iteration operator (e.g. using the Gauss-Seidel scheme).

Classically, the global error is split into contributions from previous windows due to error propagation $\mathbf{e}_{\mathbf{y},n}$, $\mathbf{e}_{\mathbf{z},n}$ and into the errors from the current window $\mathbf{d}_{\mathbf{y},n}$, $\mathbf{d}_{\mathbf{z},n}$, i.e.,

$$\begin{aligned}\boldsymbol{\epsilon}_{\mathbf{y},n} &= \mathbf{e}_{\mathbf{y},n} + \mathbf{d}_{\mathbf{y},n}\\ \boldsymbol{\epsilon}_{\mathbf{z},n} &= \mathbf{e}_{\mathbf{z},n} + \mathbf{d}_{\mathbf{z},n},\end{aligned} \tag{3.20}$$

where the propagated errors are described by

$$\begin{pmatrix}\mathbf{e}_{\mathbf{y},n}\\ \mathbf{e}_{\mathbf{z},n}\end{pmatrix} := \left(\Psi_n^{k_n} \circ \Phi_n\right)\begin{pmatrix}\tilde{\mathbf{y}}|_{(t_n-1,t_n]}\\ \tilde{\mathbf{z}}|_{(t_n-1,t_n]}\end{pmatrix} - \left(\Psi_n^{k_n} \circ \Phi_n\right)\begin{pmatrix}\mathbf{y}|_{(t_{n-1},t_n]}\\ \mathbf{z}|_{(t_{n-1},t_n]}\end{pmatrix} \tag{3.21}$$

and the local error contributions by

$$\begin{pmatrix}\mathbf{d}_{\mathbf{y},n}\\ \mathbf{d}_{\mathbf{z},n}\end{pmatrix} := \left(\Psi_n^{k_n} \circ \Phi_n\right)\begin{pmatrix}\mathbf{y}|_{(t_{n-1},t_n]}\\ \mathbf{z}|_{(t_{n-1},t_n]}\end{pmatrix} - \Psi_n^{k_n}\begin{pmatrix}\mathbf{y}|_{[t_n,t_{n+1}]}\\ \mathbf{z}|_{[t_n,t_{n+1}]}\end{pmatrix}. \tag{3.22}$$

The sum gives indeed global error, since the exact solution $(\mathbf{y}, \mathbf{z})$ is a fixed point of Ψ_n.

To investigate the convergence of the dynamic iteration scheme applied to system (3.1), we introduce a neighborhood $\mathscr{U}_{d,n}$ of the exact solution $\mathbf{x}|_{[t_n,t_{n+1}]} := (\mathbf{y}, \mathbf{z})|_{[t_n,t_{n+1}]}$, defined for any given $d > 0$ by

$$\mathscr{U}_{d,n} = \Big\{(\mathbf{Y}, \mathbf{Z}) \in C_n^{1,0} \;:\; \big|\big|\mathbf{Y} - \mathbf{y}|_{[t_n,t_{n+1}]}\big|\big|_{2,\infty},\; \big|\big|\mathbf{Z} - \mathbf{z}|_{[t_n,t_{n+1}]}\big|\big|_{2,\infty} \le d\Big\},$$

with $||\mathbf{v}||_{2,\infty} = \max_t |\mathbf{v}(t)|$, where the maximum is taken on the interval of definition of the vector function $\mathbf{v}(t)$, and $|\cdot|$ denotes the vector 2-norm, that is, the Euclidean norm. Furthermore, we assume:

Assumption 3.1 *For our problem, there exists $d_0 > 0$ such that*

- *The splitting function $\mathbf{F}$ is Lipschitz-continuous in all its coordinates on $\mathscr{U}_{d_0,n}$ with constant $L_{\mathbf{F}} > 0$,* (3.23)
- *The splitting function $\mathbf{G}$ is totally differentiable, and its derivatives are Lipschitz-continuous on $\mathscr{U}_{d_0,n}$,* (3.24)
- *The partial derivative $\mathbf{G}_{\mathbf{z}^{(k)}}$ is invertible on $\mathscr{U}_{d_0,n}$.* (3.25)

The Lipschitz continuity means: for any fixed time t and for any set of vectors $\mathbf{Y}_i, \tilde{\mathbf{Y}}_i \in \mathbb{R}^{n_y}$, $\mathbf{Z}_i, \tilde{\mathbf{Z}}_i \in \mathbb{R}^{n_z}$, $i = 1, 2$, that satisfy $|\mathbf{Y}_i - \mathbf{y}(t)|$, $|\mathbf{Z}_i - \mathbf{z}(t)|$,

$|\tilde{\mathbf{Y}}_i - \mathbf{y}(t)|$, $|\tilde{\mathbf{Z}}_i - \mathbf{z}(t)| \le d_0$, it holds

$$\begin{aligned}&\left|\mathbf{F}(\mathbf{Y}_1, \tilde{\mathbf{Y}}_1, \mathbf{Z}_1, \tilde{\mathbf{Z}}_1) - \mathbf{F}(\mathbf{Y}_2, \tilde{\mathbf{Y}}_2, \mathbf{Z}_2, \tilde{\mathbf{Z}}_2)\right| \\ &\qquad\qquad \le L_{\mathbf{F}}(|\mathbf{Y}_1 - \mathbf{Y}_2| + |\tilde{\mathbf{Y}}_1 - \tilde{\mathbf{Y}}_2| + |\mathbf{Z}_1 - \mathbf{Z}_2| + |\tilde{\mathbf{Z}}_1 - \tilde{\mathbf{Z}}_2|)\end{aligned}$$

To have a well-defined solution to (3.13), we have the second and third assumption; it is analogous to the index-1 condition.

For $0 < d < d_0$, let us consider arbitrary functions $\mathbf{X} := (\mathbf{Y}, \mathbf{Z})^\top$ and $\tilde{\mathbf{X}} := (\tilde{\mathbf{Y}}, \tilde{\mathbf{Z}})^\top \in \mathscr{U}_{d,n}$, and denote their image after k dynamic iterations by

$$\begin{aligned}\mathbf{Y}_n^k &:= \Psi_{\mathbf{y},n}^k \mathbf{X}, \quad & \mathbf{Z}_n^k &:= \Psi_{\mathbf{z},n}^k \mathbf{X}, \\ \tilde{\mathbf{Y}}_n^k &:= \Psi_{\mathbf{y},n}^k \tilde{\mathbf{X}}, \quad & \tilde{\mathbf{Z}}_n^k &:= \Psi_{\mathbf{z},n}^k \tilde{\mathbf{X}}.\end{aligned} \tag{3.26}$$

Do not confuse the above definition (3.26) with the notation in (3.17).

Let us denote distances of the y-component after k dynamic iteration by

$$\begin{aligned}\Delta_{\mathbf{y},n}^k(\mathbf{X}, \tilde{\mathbf{X}})(t) &:= \mathbf{Y}_n^k(t) - \tilde{\mathbf{Y}}_n^k(t), \\ \Delta_{\mathbf{z},n}^k(\mathbf{X}, \tilde{\mathbf{X}})(t) &:= \mathbf{Z}_n^k(t) - \tilde{\mathbf{Z}}_n^k(t), \\ \delta_{\mathbf{y},n}^k(\mathbf{X}, \tilde{\mathbf{X}}) &:= ||\Delta_{\mathbf{y},n}^k(\mathbf{X}, \tilde{\mathbf{X}})||_{2,\infty}, \\ \delta_{\mathbf{y},n}^k(\mathbf{X}, \tilde{\mathbf{X}}) &:= ||\Delta_{\mathbf{y},n}^k(\mathbf{X}, \tilde{\mathbf{X}})||_{2,\infty}.\end{aligned} \tag{3.27}$$

Now, we deduce an estimate for the error when the dynamic iteration is applied to the functions in $\mathscr{U}_{d,n}$. As in [3], we have

Lemma 3.1 (Error recursion) *Given a DAE* (3.1) *– with initial conditions* (3.2) *– and a dynamic iteration* (3.13) *with consistent splitting functions* **F**, **G**. *For the current time window* $[t_n, t_{n+1}]$ *let Assumption 3.1 hold true. Then there are constants* $C, \tilde{c} > 0$, *such that for* $d < \min\{d_0/C,\ 1/(2\tilde{c})\}$, $H < H_0 := 1/C$, *and*

$$\Psi_n^{k-1}\mathbf{X},\ \Psi_n^{k-1}\tilde{\mathbf{X}} \in \mathscr{U}_{d,n}$$

implies

$$\begin{pmatrix}\delta_{\mathbf{y},n}^k(\mathbf{X}, \tilde{\mathbf{X}}) \\ \delta_{\mathbf{z},n}^k(\mathbf{X}, \tilde{\mathbf{X}})\end{pmatrix} \le \mathbf{K}\begin{pmatrix}\delta_{\mathbf{y},n}^{k-1}(\mathbf{X}, \tilde{\mathbf{X}}) \\ \delta_{\mathbf{z},n}^{k-1}(\mathbf{X}, \tilde{\mathbf{X}})\end{pmatrix} + \begin{pmatrix}1 \\ 0\end{pmatrix} |\Delta_{\mathbf{y},n}^{k-1}(\mathbf{X}, \tilde{\mathbf{X}})(t_n)| \tag{3.28}$$

with

$$\mathbf{K} := \begin{pmatrix} CH & CH \\ C & CH + \alpha_n \end{pmatrix}, \tag{3.29}$$

$$\alpha_n := (1 + \tilde{c}\, d)\, ||\mathbf{G}_{\mathbf{z}^{(k)}}^{-1}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_{2,\infty} + Cd. \tag{3.30}$$

Notice $\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n) = \Delta^{0}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n)$ denotes the offset due to differing initial values at the beginning of the n-th time window.

Proof We apply the technique used in [1, 3]. First we show

$$\Psi^{k-1}_n\mathbf{X},\ \Psi^{k-1}_n\tilde{\mathbf{X}} \in \mathscr{U}_{d,n} \implies \delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}),\ \delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}) \le Cd \tag{3.31}$$

thus $\delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}),\ \delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}) \in \mathscr{U}_{d_0,n}$. On the one hand, we investigate the differential part (3.13a). To this end, we write this equation for any two sets of functions $\tilde{\mathbf{X}} = (\tilde{\mathbf{Y}},\tilde{\mathbf{Z}})^\top$ and $\mathbf{X} = (\mathbf{Y},\mathbf{Z})^\top$ from $\mathscr{U}_{d,n}$, which approximate the solution at the start of the dynamic iteration. Here we take the difference, and time integrate over the interval $[t_n,\tau]$, with $t_n < \tau \le t_{n+1}$. This gives for the k-th iterate, with $k > 0$,

$$\begin{aligned} |\Delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(\tau)| \le{}& |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n)| \\ &+ L_F\int_{t_n}^{\tau}\Big\{|\Delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})| + |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})| \\ &\qquad + |\Delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})| + |\Delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})|\Big\}\,\mathrm{d}t, \end{aligned} \tag{3.32}$$

using Lipschitz-continuity and consistency of $\mathbf{F}$, and observing that the initial value does not change in the iterations

$$\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n) = \Delta^{k}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n).$$

On the other hand, the algebraic part (3.13b) can be solved for variable $\mathbf{Z}^{(k)} = \hat{\boldsymbol{\phi}}(\mathbf{Y}^{(k)},\mathbf{Y}^{(k-1)},\mathbf{Z}^{(k-1)})$ due to Assumption 3.1. The Lipschitz continuity of $\hat{\boldsymbol{\phi}}$ (due to the implicit function theorem on $\mathscr{U}_{d_0,n}$) leads to

$$\begin{aligned} |\Delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})| &= |\hat{\boldsymbol{\phi}}(\mathbf{Y}^{(k)},\mathbf{Y}^{(k-1)},\mathbf{Z}^{(k-1)}) - \hat{\boldsymbol{\phi}}(\tilde{\mathbf{Y}}^{(k)},\tilde{\mathbf{Y}}^{(k-1)},\tilde{\mathbf{Z}}^{(k-1)})| \\ &\le L_{\hat{\phi}}\Big(|\Delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})| + |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})| + |\Delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})|\Big) \end{aligned} \tag{3.33}$$

for some $L_{\hat{\phi}} > 0$. Plugging this estimate into (3.32), we obtain

$$\begin{aligned} \delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) \le{}& |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n)| \\ &+ L_0\,H\Big(\delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) + \delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) + \delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})\Big), \end{aligned}$$

where $L_0 := L_F(1 + L_{\hat{\phi}})$. Now solving for $\delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})$ gives

$$\begin{aligned} \delta^k_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) \le{}& \left(1 + \frac{L_0}{1 - L_0\,H}\,H\right)|\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n)| \\ &+ \frac{L_0}{1 - L_0\,H}\,H\Big(\delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) + \delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})\Big). \end{aligned}$$

The smallness of H, i.e., $H < H_0 = C^{-1}$, implies for $C > L_0$

$$H\, L_0 < H_0\, L_0 < 1.$$

This motivates the definition $c_y := 2L_0/(1 - L_0 H_0)$ from which follows

$$\begin{aligned}\delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \;&\le \Big(1 + \frac{c_y}{2} H\Big)\, |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})(t_n)| \\ &\qquad\qquad + \frac{c_y}{2} H \big(\delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}})\big) \\ &\le |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})(t_n)| + c_y H \big(\delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}})\big),\end{aligned} \tag{3.34}$$

because the initial error at time t_n is smaller than the maximal error on the whole interval

$$|\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})(t_n)| \le \delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}).$$

Estimate (3.34) controls the error propagation for the differential variables, and it is the first line of the estimate (3.28) with the global constant $C > \max\big\{c_y, L_0\big\} = c_y$ (so far).

From the estimates (3.34) and (3.33), it is immediate to prove (3.31). In fact, by hypothesis, the $(k-1)$th iterates differ at most by $2d$, so we have

$$\begin{aligned}\delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) &\le 2(1 + 2c_y H_0)\, d, \\ \delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) &\le 2L_{\hat{\phi}}(3 + 2c_y H_0)\, d.\end{aligned} \tag{3.35}$$

Thus (3.31) holds with

$$C > \max\Big\{c_y,\; 2(1 + 2c_y H_0)d,\; 2L_{\hat{\phi}}(3 + 2c_y H_0)\, d\Big\}.$$

The error recursion estimate for the algebraic component, in the second line of estimate (3.28), can be deduced from the following homotopy of the kth iterates: let $\theta \in [0, 1]$, and let us put

$$\begin{aligned}\mathbf{Y}^{(k),\theta}(t) &:= \theta \tilde{\mathbf{Y}}^k_n(t) + (1 - \theta)\mathbf{Y}^k_n(t). \\ \mathbf{Z}^{(k),\theta}(t) &:= \theta \tilde{\mathbf{Z}}^k_n(t) + (1 - \theta)\mathbf{Z}^k_n(t).\end{aligned}$$

For the splitting function of the algebraic part, we use the short notation

$$\mathbf{G}(\theta) := \mathbf{G}\big(\mathbf{Y}^{(k),\theta}, \mathbf{Y}^{(k-1),\theta}, \mathbf{Z}^{(k),\theta}, \mathbf{Z}^{(k-1),\theta}\big) \qquad \text{and} \qquad \mathbf{G}_{\mathbf{u}}(\theta) := \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\theta),$$

for any argument $\mathbf{u}$ of $\mathbf{G}$. Notice that $\mathbf{G}(0) = \mathbf{G}(1) = 0$. Thus a version of the fundamental theorem of calculus yields:

$$\begin{aligned} 0 &= \mathbf{G}(1) - \mathbf{G}(0) \\ &= \int_0^1 \Big(\mathbf{G}_{\mathbf{y}^{(k)}}(\theta) \Delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) + \mathbf{G}_{\mathbf{y}^{(k-1)}}(\theta) \Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ &\qquad + \mathbf{G}_{\mathbf{z}^{(k)}}(\theta)\, \Delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) + \mathbf{G}_{\mathbf{z}^{(k-1)}}(\theta)\, \Delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \Big) \mathrm{d}\theta, \end{aligned} \tag{3.36}$$

since $\frac{\partial}{\partial \theta} \mathbf{Y}^{(k),\theta} = \Delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})$, and so forth. The upper bound of d in terms of d_0, i.e.,

$$Cd \le d_0$$

allows us to use the Lipschitz continuity of $\mathbf{G}_{\mathbf{z}^{(k)}}$ on $\mathscr{U}_{d_0,n}$ (inside the integral of (3.36)). We denote the corresponding constant by L'_G. Together with the above estimate (3.31), we obtain for any time $t \in [t_n, t_{n+1}]$

$$\begin{aligned} |\mathbf{G}_{\mathbf{u}}(\theta) - \mathbf{G}_{\mathbf{u}}(\hat{\theta})| \le\; & L_{G'} \Big(\big| \theta \tilde{\mathbf{Y}}^k_n(t) + (1-\theta) \mathbf{Y}^k_n(t) \\ & \qquad - \hat{\theta} \tilde{\mathbf{Y}}^k_n(t) - (1-\hat{\theta}) \mathbf{Y}^k_n(t) \big| \\ & \qquad + \cdots + \big| \theta \tilde{\mathbf{Z}}^{k-1}_n(t) + (1-\theta) \mathbf{Z}^{k-1}_n(t) \\ & \qquad - \hat{\theta} \tilde{\mathbf{Z}}^{k-1}_n(t) - (1-\hat{\theta}) \mathbf{Z}^{k-1}_n(t) \big| \Big) \\ =\; & L_{G'} |\theta - \hat{\theta}| \Big(|\Delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})| + |\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})| \\ & \qquad + |\Delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}})| + |\Delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}})| \Big) \le c_g d. \end{aligned} \tag{3.37}$$

(This defines c_g in the obvious way.) The operator $\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}(0)$ exists due to Assumption 3.1. Left-multiplication of (3.36) by $\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}(0)$ yields:

$$\begin{aligned} 0 = \int_0^1 \mathbf{G}^{-1}_{\mathbf{z}^{(k)}}(0) \Big(& \big(\mathbf{G}_{\mathbf{z}^{(k)}}(0) + \big[\mathbf{G}_{\mathbf{z}^{(k)}}(\theta) - \mathbf{G}_{\mathbf{z}^{(k)}}(0) \big] \big) \Delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ & + \big(\mathbf{G}_{\mathbf{z}^{(k-1)}}(0) + \big[\mathbf{G}_{\mathbf{z}^{(k-1)}}(\theta) - \mathbf{G}_{\mathbf{z}^{(k-1)}}(0) \big] \big) \Delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ & + \big(\mathbf{G}_{\mathbf{y}^{(k)}}(0) + \big[\mathbf{G}_{\mathbf{y}^{(k)}}(\theta) - \mathbf{G}_{\mathbf{y}^{(k)}}(0) \big] \big) \Delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ & + \big(\mathbf{G}_{\mathbf{y}^{(k-1)}}(0) + \big[\mathbf{G}_{\mathbf{y}^{(k-1)}}(\theta) - \mathbf{G}_{\mathbf{y}^{(k-1)}}(0) \big] \big) \Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \Big)\, \mathrm{d}\theta. \end{aligned}$$

The matrices $\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}$, $\mathbf{G}_{\mathbf{z}^{(k-1)}}$, $\mathbf{G}_{\mathbf{y}^{(k)}}$, $\mathbf{G}_{\mathbf{y}^{(k-1)}}$ are uniformly bounded on $\mathscr{U}_{d_0,n}$. Let the constant be denoted by c'_g. Now, this equation is (partially) solved for the first bit of

$\Delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}})$. Using

$$\begin{aligned}||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_2 &= ||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_2(0) \\ &= \big|\big|\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}\big|\big|_2\big(\mathbf{Y}^k_n(t), \mathbf{Y}^k_n(t), \mathbf{Z}^k_n(t), \mathbf{Z}^k_n(t)\big)\end{aligned}$$

and applying the maximum norm in time as well as (3.37) gives

$$\begin{aligned}\delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \le &\Big(||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_{2,\infty} + \frac{\tilde{c}}{2} d\Big)\, \delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ &+ \frac{\tilde{c}}{2}\, d\, \delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) + c_h \left(\delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) + \delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})\right)\end{aligned}$$

with $c_h := (c_g d + c'_g)\, c'_g$ and $\tilde{c} := 2\, c_g c'_g$. Inserting the estimate for $\delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})$ (3.34), we deduce, having H and d small enough, such that $d < 1/(2\tilde{c})$, the estimate

$$\begin{aligned}\delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \le &(1 + \tilde{c}d)c_h\Big(|\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})(t_n)| + (1 + c_y H)\delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})\Big) \qquad (3.38) \\ &+ (1 + \tilde{c}d)\Big(c_h c_y H + ||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_{2,\infty} + \frac{\tilde{c}}{2} d\Big)\, \delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ \le &(1 + \tilde{c}d)c_h(2 + c_y H_0)\delta^{k-1}_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \qquad (3.39) \\ &+ (1 + \tilde{c}d)\Big(c_h c_y H + ||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_{2,\infty} + \frac{\tilde{c}}{2} d\Big)\, \delta^{k-1}_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}),\end{aligned}$$

$(H < H_0)$. Finally, summing up, the global constant C should be large enough to state (3.31) from (3.34), (3.35) and to obtain from estimate (3.39) the claim (3.28) with (3.29). Hence we conclude

$$\begin{aligned}C > \max\Big\{&c_y,\ 2(1 + 2c_y H_0)\, d,\ 2L_{\hat{\phi}}(3 + 2c_y H_0)\, d \\ &(1 + \tilde{c}d)c_h(2 + c_y H_0), (1 + \tilde{c}d)c_h c_y, \frac{\tilde{c}}{2}\Big\}.\end{aligned}$$

Then (3.34) and (3.39) yield the recursion (3.28), our claim. □

When iteratively applying Lemma 3.1, one can deduce the following rather technical result, which is proven for an analogous recursion in [1]:

Proposition 3.1 (Recursion Estimate) *Let the splitting functions fulfill the assumptions of Lemma 3.1 and $\alpha_n < 1$, $C > \alpha_n$, then there is a constant C_0*

such that

$$\begin{pmatrix} \delta^k_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ \delta^k_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \end{pmatrix} \leq \begin{pmatrix} C(4C+1)H\mu_n^{\max(0,k-2)} & 4CH\mu_n^{k-2} \\ 4C\mu_n^{k-1} & \mu_n^k + (\mu_n - \alpha_n)^k \end{pmatrix} \begin{pmatrix} \delta^0_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}}) \\ \delta^0_{\mathbf{z},n}(\mathbf{X}, \tilde{\mathbf{X}}) \end{pmatrix} + \begin{pmatrix} 1 + C_0 H \\ C_0 \end{pmatrix} \cdot \delta^0_{\mathbf{y},n}(\mathbf{X}, \tilde{\mathbf{X}})(t_n) \tag{3.40}$$

with

$$\mu_n = \mu(\alpha_n, H) := \alpha_n + \frac{2CH}{\frac{\alpha_n}{2C} + \sqrt{H}} \tag{3.41}$$

is satisfied for all $k \geq 1$ *and for all* $H \leq H_0$.

This result is proven for a similar setting in [1]. It is established using the same arguments as in the proof of Theorem 3.1 for the local error: the iteration error is determined by the powers of its matrix $\mathbf{K}$ as given in (3.29) and the computation of the eigenvalues and eigenvectors as in (3.44) proofs the claim.

Next, we will employ the above estimates to show that the mapping is indeed a fixed-point operator.

3.2.3.2 Contraction and Local Error

We consider in this section the local error as defined in Eq. (3.22) only, where the error of a single iteration starting from exact data is analyzed

$$\mathbf{d}_{\mathbf{y},n} = \Delta^{k_n}_{\mathbf{y},n}(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n \mathbf{x}|_{(t_{n-1},t_n]}),$$

and analogously for $\mathbf{d}_{\mathbf{z},n}$ with $\mathbf{x} := (\mathbf{y}, \mathbf{z})^\top$ in both cases. We follow [3] and the strategy from [1] to proof the following result, that is already predicted in [19]. It shows that the crucial point in the coupling lies in the algebraic-to-algebraic coupling, which is represented by the additional DAE-contraction factor α.

Theorem 3.1 (Contraction) *The splitting functions shall fulfill the assumptions of Lemma 3.1 including the index-1 assumption. Furthermore, let* $\mathbf{x}$ *denote our exact solution. Then for* d *and* $H < H_0$ *small enough the map*

$$\delta^{k-1}_n(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n \mathbf{x}|_{(t_{n-1},t_n]}) \mapsto \delta^k_n(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n \mathbf{x}|_{(t_{n-1},t_n]}) \tag{3.42}$$

is strongly contractive for all k *provided that*

$$||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}} \, \mathbf{G}_{\mathbf{z}^{(k-1)}}||_{2,\infty} < 1. \tag{3.43}$$

Proof We show contractivity for the constant extrapolation with $\tilde{y}_n^{(0)} = \tilde{y}(t_n)$, $z_n^{(0)} = \tilde{z}(t_n)$, from which the contraction for any higher order polynomial extrapolation follows automatically.

By induction we setup the error recursion (3.28) in $\mathscr{U}_{d,n}$: as induction basis, we have for $k = 0$ and $\tau \in [t_n, t_{n+1}]$

$$|\varDelta^0_{\mathbf{y},n}(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n\mathbf{x}|_{(t_{n-1},t_n]})|(\tau) = \Big|\int_{t_n}^{\tau} \mathbf{f}(\mathbf{y},\,\mathbf{z})\,\mathrm{d}t\,\Big| \leq c_f H,$$

where $c_f := ||\mathbf{f}(\mathbf{y}, \mathbf{z})||_{2,\infty}$. Then the index-1 assumption implies for $\mathbf{z}$

$$\begin{aligned}\big|\varDelta^0_{\mathbf{z},n}(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n\mathbf{x}|_{(t_{n-1},t_n]})\big|(\tau) &\leq \big|\boldsymbol{\phi}(\Phi_{\mathbf{y},n}\mathbf{x}|_{(t_{n-1},t_n]}) - \boldsymbol{\phi}(\mathbf{y})\big| \\ &\leq L_\phi \big|\Phi_{\mathbf{y},n}\mathbf{x}|_{(t_{n-1},t_n]} - \mathbf{y}\big| \\ &\leq c_f L_\phi H;\end{aligned}$$

thus choosing H sufficient small, such that $c_f\ (L_\phi + 1)\, H_0 < 1$ (and $H < H_0$), we obtain an extrapolation, which lies in the neighborhood of the solution: $\Phi_n\mathbf{x} \in \mathscr{U}_{d,n}$.

Recall the definition of the matrix $\mathbf{K}$ (3.29), which denotes an upper bound on the error recursion. Now, the mapping (3.42) is contractive if the spectral radius $\rho(\mathbf{K}) < 1$. The eigenvalues of $\mathbf{K}$ are

$$\lambda_{1,2}(\mathbf{K}) = \frac{1}{2}\Big(\alpha_n + 2\,CH \pm \sqrt{\alpha_n^2 + 4\,C^2H}\,\Big), \tag{3.44}$$

Therefore $\alpha_n < 1$ is sufficient for contraction provided that d and H_0 are small enough. Inspecting (3.30), this translates into:

$$||\mathbf{G}^{-1}_{\mathbf{z}^{(k)}}\,\mathbf{G}_{\mathbf{z}^{(k-1)}}||_{2,\infty} < 1.$$

Eventually applying Lemma 3.1 iteratively and using

$$\delta^0_{\mathbf{y},n}(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n\mathbf{x}|_{(t_{n-1},t_n]})(t_n) = 0$$

concludes the proof. □

Remark 3.5 (Convergence Order of Iteration) The eigenvalues of $\mathbf{K}$, defined in (3.29), suggest a certain order of convergence (i.e., for the asymptotics as $H \to 0$) for the dynamic iteration (3.13): For the rate of convergence, we use Taylor expansion of the square root term in $\lambda(\mathbf{K})$ (3.44) and find

$$\sqrt{\alpha_n^2 + 4\,C^2H} = \alpha_n\ (1 + 2\,C^2H/\alpha_n^2) + \mathscr{O}(H^2).$$

This suggests a order of $\alpha_n + \mathscr{O}(H)$, if α_n does not vanish and $4\,C^2H < \alpha_n^2$. For $\alpha_n = 0$, we have order $\sqrt{H}$.

We notice: convergence of the DAE-distributed time integration depends on the stability of the algebraic-to-algebraic component coupling (3.43) (and, of course, depends on the mentioned hypothesis). Thus modeling coupling is important for DAEs and should be organized if possible in a way, s.t. contractivity (stability) is directly given. The following important special case avoids these kinds of dependencies:

Corollary 3.1 (Simple Coupling) *Let the hypothesis of Lemma 3.1 be fulfilled.*

(i) If no algebraic constraint depends on an old algebraic variable, i.e.,

$$\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0 \tag{3.45}$$

then contraction is archived with $\alpha_n = 0$.

(ii) If no algebraic constraint depends on an old algebraic or a differential variable, i.e.,

$$\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0 \quad \textit{and} \quad \mathbf{G}_{\mathbf{y}^{(k-1)}} = 0 \tag{3.46}$$

then the contraction is archived with convergence order H.

Proof We discuss (3.36) for the special cases in which the given partial derivatives vanish.

(i) The assumption $\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0$ gives the following estimate for the algebraic part replacing (3.39)

$$\delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}) \;\le\; \le C\,\delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) + CH\,\delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}).$$

This is (3.28) with $\alpha_n = 0$. This result is in the spirit of the numerical DAE-theory (cf. [21]).

(ii) Analogously $\mathbf{G}_{\mathbf{y}^{(k-1)}} = \mathbf{G}_{\mathbf{z}^{(k-1)}} = 0$ yields

$$\begin{aligned}\delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}) \;&\le (1+\tilde{c}d)c_h\Big(|\Delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n)| + c_y H \delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})\Big)\\ &\quad + (1+\tilde{c}d)c_h c_y H\,\delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}})\end{aligned}$$

replacing (3.38). This give the error recursion

$$\delta^k_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}) \;\le CH\,\delta^{k-1}_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}}) + CH\,\delta^{k-1}_{\mathbf{z},n}(\mathbf{X},\tilde{\mathbf{X}}) + C\,|\Delta^0_{\mathbf{y},n}(\mathbf{X},\tilde{\mathbf{X}})(t_n)|$$

which unveils a contraction operator $\mathbf{K} = \mathscr{O}(H)$ and hence implies a convergence order of H, cf. Remark 3.5. Only the initial offset cannot be improved.

□

Now still following the strategy from [1], we prove estimates for the local and propagated errors, and conclude from those results the overall stability and convergence of the method for the nth time window.

Proposition 3.2 (Local Error) *Let the assumptions of Lemma 3.1 be fulfilled, then the recursion* (3.40) *with* μ_n (3.41) *of that Lemma holds. Moreover, then there is for a sufficiently small* $H < H_0$ *a constant* $C_{\mathbf{d}^\star}$*, independent of* H*,* α_n *and* k_n*, such that the local error is bounded by*

$$||\mathbf{d}_{\mathbf{y},n}|| + H||\mathbf{d}_{\mathbf{z},n}|| \leq C_{\mathbf{d}^\star} H\delta_n^0$$

where the right-hand-side is given in terms of the extrapolation errors

$$\delta_n^0 := \mu_n^{\max(0,k_n-2)} \delta_{\mathbf{y},n}^0(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n\mathbf{x}|_{(t_{n-1},t_n]}) \\ + \mu_n^{k_n-1}\delta_{\mathbf{z},n}^0(\mathbf{x}|_{[t_n,t_{n+1}]}, \Phi_n\mathbf{x}|_{(t_{n-1},t_n]})$$

Proof The proof of Theorem 3.1 showed that $\Phi_n\mathbf{x}|_{(t_{n-1},t_n]} \in \mathscr{U}_{d,n}$ for H sufficiently small. Therefore applying Proposition 3.1 to the specific functions

$$\mathbf{X} := \mathbf{x}|_{(t_n,t_{n+1}]} \quad \text{and} \quad \tilde{\mathbf{X}} := \Phi_n\mathbf{x}|_{(t_{n-1},t_n]}$$

where $\mathbf{x} = (\mathbf{y}, \mathbf{z})$ is the exact solution. Notice $\delta_{\mathbf{y},n}^0(\mathbf{X}, \tilde{\mathbf{X}})(t_n) = 0$ holds, since the initial values are equal. Summation of the two equations in (3.40) yields the claimed estimate. □

This proves convergence for one window (for $k_n \to \infty$), since $\mu_n < 1$ for H sufficiently small. Next, we have to address the error transport, since the iteration is stopped after a finite number of iterations and we are not performing $k_n \to \infty$ in the numerical treatment.

3.2.3.3 Stability and Convergence for Windowing Technique

To obtain convergence and stability of the method on multiple windows it is crucial to control the error propagation from the previous window to the current one, hence we need to inspect

$$\mathbf{e}_{\mathbf{y},n} = \Delta_{\mathbf{y},n}^{k_n}(\Phi_n\mathbf{x}|_{(t_{n-1},t_n]}, \Phi_n\tilde{\mathbf{x}}|_{(t_{n-1},t_n]})$$

and the similar expression for $\mathbf{e}_{\mathbf{z},n}$ (here $\mathbf{x}$ denotes the analytic solution and $\tilde{\mathbf{x}}$ an approximation). The following result is again a consequence of Proposition 3.1 (cf. [1]):

Proposition 3.3 (Propagation Error) *Let an continuous extrapolation* (3.11) *be given, that is of accuracy* $\mathscr{O}(H)$ *and satisfies a uniform Lipschitz condition (with* *constant L_Φ) and a dynamic iteration* (3.13)*, which fulfill the assumptions of Proposition 3.1 with $\mu_n < 1$, then there is a constant $C_{\mathbf{e}^\star} > 0$, such that the propagation error is bounded by*

$$\begin{pmatrix} ||\mathbf{e}_{\mathbf{y},n}|| \\ ||\mathbf{e}_{\mathbf{z},n}|| \end{pmatrix} \le \begin{pmatrix} 1 + C_{\mathbf{e}^\star} & C_{\mathbf{e}^\star} H \\ C_{\mathbf{e}^\star} & \alpha_{n^\star} \end{pmatrix} \cdot \begin{pmatrix} ||\boldsymbol{\epsilon}_{\mathbf{y},n-1}|| \\ ||\boldsymbol{\epsilon}_{\mathbf{z},n-1}|| \end{pmatrix} \tag{3.47}$$

with $\alpha_n^\star$ depending on the Lipschitz constant L_Φ of the extrapolation operator

$$\alpha_n^\star := L_\Phi(\mu_n^{k_n} + (\mu_n - \alpha_n)^{k_n}) \tag{3.48}$$

Proof When applying Proposition 3.1 to the extrapolation of exact and erroneous functions of the previous time window

$$\mathbf{X} := \Phi_n \mathbf{x}|_{(t_{n-1},t_n]} \quad \textit{and} \quad \tilde{\mathbf{X}} := \Phi_n \tilde{\mathbf{x}}|_{(t_{n-1},t_n]}\ ,$$

we will have an offset in the initial values (at t_n), which is bounded by the total error on the interval

$$||\varDelta^0_{\mathbf{y},n}(\Phi_n \mathbf{x}|_{(t_{n-1},t_n]}, \Phi_n \tilde{\mathbf{x}}|_{(t_{n-1},t_n]})(t_n)|| \le ||\mathbf{y} - \tilde{\mathbf{y}}||_{(t_{n-1},t_n]}.$$

Furthermore the extrapolation operator is a uniformly Lipschitz continuous mapping with Lipschitz constant L_Φ, hence we have

$$\begin{aligned} \delta^0_n(\Phi_n \mathbf{x}|_{(t_{n-1},t_n]}, \Phi_n \tilde{\mathbf{x}}|_{(t_{n-1},t_n]}) &\le L_\Phi \begin{pmatrix} ||\mathbf{y} - \tilde{\mathbf{y}}||_{(t_{n-1},t_n]} \\ ||\mathbf{z} - \tilde{\mathbf{z}}||_{(t_{n-1},t_n]} \end{pmatrix} \\ &= L_\Phi \begin{pmatrix} ||\mathbf{e}_{\mathbf{y},n-1}|| \\ ||\mathbf{e}_{\mathbf{z},n-1}|| \end{pmatrix}, \end{aligned}$$

that completes together with Eq. (3.40) of Proposition 3.1 the proof. □

Now bringing all pieces together, we obtain the following result on stability and convergence

Theorem 3.2 (Stability) *Let a continuous extrapolation Φ* (3.11) *be given, that is of accuracy order $\mathcal{O}(H)$ and satisfies a uniform Lipschitz condition (L_Φ), further a dynamic iteration* (3.13)*, where the splitting functions* **F**, **G** *are consistent and for the current time window $[t_n, t_{n+1}]$ let Assumption 3.1 hold true, furthermore the contractivity constant is bounded*

$$\alpha_n \le \bar{\alpha} < 1 \qquad \textit{and} \qquad L_\Phi \alpha_n^{k_n} \le \bar{\alpha}$$

and the numerical solution remains close to the exact solution

$$||\boldsymbol{\epsilon}_{\mathbf{y},m}|| + ||\boldsymbol{\epsilon}_{\mathbf{z},m}|| \le d \quad \textit{for } 0 \le m < n,$$

then there is a constant $C^\star > 0$, *independent of* n *and* H, *such that the total error on the time window* $[t_n, t_{n+1}]$ *is bounded by*

$$||\boldsymbol{\epsilon}_{\mathbf{y},n}|| + ||\boldsymbol{\epsilon}_{\mathbf{z},n}|| \leq C^\star \max_{0\leq m<n} \delta_m^0 \leq d \tag{3.49}$$

all for a sufficiently small step size $0 < H < H_0$.

Proof According to Eq. (3.41) we have $\mu_n = \alpha_n + \mathscr{O}(H)$ and by assumption $L_\Phi \alpha_n^{k_n} \leq \bar{\alpha}$, hence

$$\alpha_n^\star = L_\Phi \left((\mu_n^{k_n})^{k_n} + (\mu_n - \alpha_n)^{k_n}\right) = L_\Phi \left((\alpha_n + \mathscr{O}(H))^{k_n} + \mathscr{O}(H)^{k_n}\right) < 1,$$

and therefore the maximum is bounded as well

$$\alpha^\star := \max_{0\leq m\leq n} \alpha_m^\star < 1.$$

Now combining the results from Propositions 3.2 and 3.3 yields

$$\begin{pmatrix} ||\boldsymbol{\epsilon}_{\mathbf{y},n}|| \\ ||\boldsymbol{\epsilon}_{\mathbf{z},n}|| \end{pmatrix} \leq \begin{pmatrix} 1 + C_{\mathbf{e}^\star} & C_{\mathbf{e}^\star} H \\ C_{\mathbf{e}^\star} & \alpha^\star \end{pmatrix} \cdot \begin{pmatrix} ||\boldsymbol{\epsilon}_{\mathbf{y},n-1}|| \\ ||\boldsymbol{\epsilon}_{\mathbf{z},n-1}|| \end{pmatrix} + \begin{pmatrix} C_{\mathbf{d}^\star} H \delta_n^0 \\ C_{\mathbf{d}^\star} \delta_n^0 \end{pmatrix}$$

and this proves the left half of (3.49), the right bound is enforceable since the extrapolation error $\delta_m^0 = \mathscr{O}(H)$ decreases with the step size. □

One can use Theorem 3.2 to prove by induction that the numerical solution remains close to the exact solution, analogously to the application in [1], then the overall convergence and stability follows by

Corollary 3.2 (Global Convergence and Stability) *Let the assumptions of Theorem 3.2 be fulfilled, then there is a constant* $C^\star$, *such that the estimate holds*

$$||\tilde{y} - y||_{[0,t_e]} + ||\tilde{z} - z||_{[0,t_e]} \leq C^\star \cdot \max_{0\leq n<N} \delta_m^0,$$

where δ_m^0 *is the extrapolation error on the* m*-th window.*

This result shows convergence and stability, since the global error can by controlled in terms of the step size H, which determines the extrapolation error.

3.3 Applications in Electrical Engineering

In this section we show how the dynamic iteration theory can be used to study the convergence of iteration schemes for the main coupled models introduced in the previous chapter. These problems basically stem from chip design.

3.3.1 Refined Network Models

We consider an electric network with semiconductor devices, modeled by drift-diffusion equations. The electric network is described by the MNA equations, which can be written in the form:

$$\begin{aligned} \mathbf{A}_C \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}_C(\mathbf{A}_C^T\mathbf{u}) + \mathbf{A}_R\mathbf{r}(\mathbf{A}_R^T\mathbf{u}) + \mathbf{A}_L\mathbf{i}_L + \mathbf{A}_V\mathbf{i}_V + \mathbf{A}_I\mathbf{i}_I + \boldsymbol{\lambda} &= 0, \\ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\phi}_L(\mathbf{i}_L) - \mathbf{A}_L^T\mathbf{u} &= 0, \\ \mathbf{A}_V^T\mathbf{u} - \mathbf{v}_V &= 0. \end{aligned} \tag{3.50}$$

This system is supplemented with initial data for the differential part,

$$\mathbf{P}_C\mathbf{u}(t_0) = \mathbf{P}_C\mathbf{u}_0, \quad \mathbf{i}_L(t_0) = \mathbf{i}_{L,0}. \tag{3.51}$$

Here, we have $\mathbf{P}_C = \mathbf{I} - \mathbf{Q}_C$, where $\mathbf{Q}_C$ is a projector onto the null-space of $\mathbf{A}_C^T$, and we are assuming index-1 conditions for the uncoupled MNA system.

The above equations are coupled, through the current term $\boldsymbol{\lambda}$, to the drift-diffusion equations which describe the devices contained in the circuit. Here, as an exemplification, we use the space-discretization derived in the previous Chapter, by means of the Box Integration method. Then, assuming for simplicity that the circuit contains a single device, this device will be described by the time-dependent vectors $\boldsymbol{\phi}$, $\boldsymbol{n}$, $\boldsymbol{p}$, comprising the values of the electric potential ϕ, the electron concentration n and the hole concentration p, evaluated on the inner grid points, and by the time-dependent vectors $\boldsymbol{\phi}^\partial$, $\boldsymbol{n}^\partial$, $\boldsymbol{p}^\partial$, comprising the values of ϕ, n and p on the boundary grid points. As we have seen in the previous Chapter, these vector functions satisfy the following equations:

$$\begin{aligned} \mathbf{A}_\phi\boldsymbol{\phi} + \mathbf{A}_\phi^\partial\boldsymbol{\phi}^\partial &= \mathbf{b}_\phi(\boldsymbol{n},\boldsymbol{p}), \\ \mathbf{A}^\partial\boldsymbol{\phi} + \boldsymbol{\phi}^\partial &= \mathbf{b}_\phi^\partial(\mathbf{u}_D), \\ \mathbf{A}_0\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}t} + \mathbf{A}_n(\boldsymbol{\phi})\boldsymbol{n} + \mathbf{A}_n^\partial(\boldsymbol{\phi})\boldsymbol{n}^\partial &= \mathbf{b}_n(\boldsymbol{n},\boldsymbol{p}), \\ \mathbf{A}^\partial\boldsymbol{n} + \boldsymbol{n}^\partial &= \mathbf{b}_n^\partial, \\ \mathbf{A}_0\frac{\mathrm{d}\boldsymbol{p}}{\mathrm{d}t} + \mathbf{A}_p(\boldsymbol{\phi})\boldsymbol{p} + \mathbf{A}_p^\partial(\boldsymbol{\phi})\boldsymbol{p}^\partial &= \mathbf{b}_p(\boldsymbol{n},\boldsymbol{p}), \\ \mathbf{A}^\partial\boldsymbol{p} + \boldsymbol{p}^\partial &= \mathbf{b}_p^\partial. \end{aligned} \tag{3.52}$$

These equations must be supplemented with initial data for the differential variables,

$$\boldsymbol{n}(t_0) = \boldsymbol{n}_0, \quad \boldsymbol{p}(t_0) = \boldsymbol{p}_0. \tag{3.53}$$

The MNA equations (3.50) and the device equations (3.52) are coupled by means of appropriate relations which express, on the one hand, the boundary electric potential $\mathbf{u}_D$ in (3.52) in terms of the network variables (network-to-device coupling), and on the other hand, the device current source term $\boldsymbol{\lambda}$ in (3.50) in terms of the device variables (device-to-network coupling). The network-to-device coupling is given by:

$$\mathbf{u}_D = \mathbf{S}_D^T \mathbf{u}. \tag{3.54}$$

The device-to-network coupling is more involved. In Chap. 1 we have introduced two alternative formulations. In the first formulation, the device-to-network coupling relation is given by:

$$\boldsymbol{\lambda} = \mathbf{A}_D \mathbf{i}_D, \quad \mathbf{i}_D = \mathbf{A}^c \frac{\mathrm{d}\boldsymbol{\phi}}{\mathrm{d}t} + \mathbf{A}_n^c(\boldsymbol{\phi})\mathbf{n} + \mathbf{A}_p^c(\boldsymbol{\phi})\mathbf{p}, \tag{3.55}$$

with $\mathbf{A}_D = \mathbf{S}_D \hat{\mathbf{A}}_D$. This formulation is problematic, because of the appearance of the time derivative of $\boldsymbol{\phi}$, which is an algebraic variable for the uncoupled device system. Thus, after the coupling, the set of differential variables generally differs from the union of the differential variables for the network and the device system, considered as uncoupled.

For this reason, we consider the alternative formulation,

$$\boldsymbol{\lambda} = \mathbf{A}_D \frac{\mathrm{d}}{\mathrm{d}t}(\tilde{\mathbf{C}}_D \mathbf{A}_D^\top \mathbf{u}) + \mathbf{A}_D \tilde{\mathscr{I}}_D, \quad \tilde{\mathscr{I}}_D = \mathbf{A}_n^c(\boldsymbol{\phi})\mathbf{n} + \mathbf{A}_p^c(\boldsymbol{\phi})\mathbf{p}. \tag{3.56}$$

In this formulation, it is simpler to see that the differential variables for the coupled system are $\mathbf{P}_C\mathbf{u}$, $\mathbf{i}_L$, $\boldsymbol{n}$, $\boldsymbol{p}$, provided the additional condition

$$\mathbf{A}_D^T \mathbf{Q}_C = 0. \tag{3.57}$$

Under this condition, we can identify the differential and algebraic components, and we set

$$\mathbf{y}_c = \begin{pmatrix} \mathbf{P}_C\mathbf{u} \\ \mathbf{i}_L \end{pmatrix}, \qquad \mathbf{z}_c = \begin{pmatrix} \mathbf{Q}_C\mathbf{u} \\ \mathbf{i}_V \end{pmatrix},$$

and

$$\mathbf{y}_d = \begin{pmatrix} \boldsymbol{n} \\ \boldsymbol{p} \end{pmatrix}, \qquad \mathbf{z}_d = \begin{pmatrix} \boldsymbol{\phi} \\ \boldsymbol{n}^\partial \\ \boldsymbol{p}^\partial \\ \boldsymbol{\phi}^\partial \end{pmatrix}.$$

Then, using the standard reduction to differential and algebraic equations, by means of appropriate projectors, the two systems of equations can be written in the following form:

$$\begin{aligned}
\dot{\mathbf{y}}_c &= \mathbf{f}_c(\mathbf{y}_c, \mathbf{z}_c, \tilde{\boldsymbol{\lambda}}),\\
0 &= \mathbf{g}_c(\mathbf{y}_c, \mathbf{z}_c),\\
\dot{\mathbf{y}}_d &= \mathbf{f}_d(\mathbf{y}_d, \mathbf{z}_d),\\
0 &= \mathbf{g}_d(\mathbf{y}_d, \mathbf{z}_d, \mathbf{u}_D),
\end{aligned} \tag{3.58}$$

with

$$\tilde{\boldsymbol{\lambda}} = \tilde{\boldsymbol{\lambda}}(\mathbf{y}_d, \mathbf{z}_d) := \mathbf{A}_D[\mathbf{A}_n^c(\boldsymbol{\phi})\boldsymbol{n} + \mathbf{A}_p^c(\boldsymbol{\phi})\boldsymbol{p}].$$

Also, we have

$$\mathbf{u}_D = \mathbf{S}_D^T\,(\mathbf{P}_C\mathbf{u} + \mathbf{Q}_C\mathbf{u}) = \mathbf{S}_D^T(\mathbf{y}_c + \mathbf{z}_c),$$

so the above system becomes

$$\begin{aligned}
\dot{\mathbf{y}}_c &= \mathbf{f}_c^*(\mathbf{y}_c, \mathbf{z}_c, \mathbf{y}_d, \mathbf{z}_d),\\
0 &= \mathbf{g}_c(\mathbf{y}_c, \mathbf{z}_c),\\
\dot{\mathbf{y}}_d &= \mathbf{f}_d(\mathbf{y}_d, \mathbf{z}_d),\\
0 &= \mathbf{g}_d^*(\mathbf{y}_d, \mathbf{z}_d, \mathbf{y}_c, \mathbf{z}_c).
\end{aligned} \tag{3.59}$$

Next, we apply the dynamic iteration theory, expounded in this Chapter, to the coupled system (3.59), by using the Gauss-Seidel method. We can use to different strategies: circuit-device iteration, and device-circuit iteration. For the circuit-device coupling, we have[1]:

$$\begin{aligned}
\dot{\tilde{\mathbf{y}}}_c^{(k)} &= \mathbf{f}_c^*(\tilde{\mathbf{y}}_c^{(k)}, \tilde{\mathbf{z}}_c^{(k)}, \tilde{\mathbf{y}}_d^{(k-1)}, \tilde{\mathbf{z}}_d^{(k-1)}),\\
0 &= \mathbf{g}_c(\tilde{\mathbf{y}}_c^{(k)}, \tilde{\mathbf{z}}_c^{(k)}),\\
\dot{\tilde{\mathbf{y}}}_d^{(k)} &= \mathbf{f}_d(\tilde{\mathbf{y}}_d^{(k)}, \tilde{\mathbf{z}}_d^{(k)}),\\
0 &= \mathbf{g}_d^*(\tilde{\mathbf{y}}_c^{(k)}, \tilde{\mathbf{z}}_c^{(k)}, \tilde{\mathbf{y}}_d^{(k)}, \tilde{\mathbf{z}}_d^{(k)}).
\end{aligned} \tag{3.60}$$

We can observe that in this case the matrix $\mathbf{G}_{\mathbf{z}^{(k-1)}}$ is identically zero. Thus, by Corollary 3.1, this scheme leads to an unconditionally, strongly contractive map.

[1] For simplicity we omit the subscript n.

By contrast, if we consider the device-circuit iteration scheme, we have

$$\begin{aligned}
\dot{\tilde{\mathbf{y}}}_d^{(k)} &= \mathbf{f}_d(\tilde{\mathbf{y}}_d^{(k)}, \tilde{\mathbf{z}}_d^{(k)}),\\
0 &= \mathbf{g}_d^*(\tilde{\mathbf{y}}_c^{(k-1)}, \tilde{\mathbf{z}}_c^{(k-1)}, \tilde{\mathbf{y}}_d^{(k)}, \tilde{\mathbf{z}}_d^{(k)}),\\
\dot{\tilde{\mathbf{y}}}_c^{(k)} &= \mathbf{f}_c^*(\tilde{\mathbf{y}}_c^{(k)}\tilde{\mathbf{z}}_c^{(k)}, \tilde{\mathbf{y}}_d^{(k)}, \tilde{\mathbf{z}}_d^{(k)}),\\
0 &= \mathbf{g}_c(\tilde{\mathbf{y}}_c^{(k)}, \tilde{\mathbf{z}}_c^{(k)}).
\end{aligned}$$

and the condition (3.43), in Theorem 3.1, which ensure the contractivity of the dynamic iteration map, is verified if and only if

$$\left\| \left(\frac{\partial \mathbf{g}_d^*}{\partial \mathbf{z}_d^{(k)}} \right)^{-1} \frac{\partial \mathbf{g}_d^*}{\partial \mathbf{z}_c^{(k-1)}} \right\| < 1.$$

Explicitly, this condition is equivalent to

$$\left\| (\mathbf{A}_\phi - \mathbf{A}_\phi^\partial \mathbf{A}^\partial)^{-1} \frac{\partial \mathbf{b}_\phi^\partial}{\partial \mathbf{u}_D} \mathbf{S}_D^T \mathbf{Q}_C \right\| < 1, \tag{3.61}$$

where, by definition, we have

$$\frac{\partial b_{\phi,i}^\partial}{\partial u_{D,j}} = \begin{cases} 1, \text{ if } \mathbf{x}_i \in \Gamma_{D,j}, \\ 0, \text{ otherwise.} \end{cases}$$

The matrix $\mathbf{A}_\phi - \mathbf{A}_\phi^\partial \mathbf{A}^\partial$ depends on the space-discretization, so the above condition implies a smallness assumption on the spacing of the grid, unless $\mathbf{S}_D^T \mathbf{Q}_C = 0$. This condition is stronger than the additional topological condition (3.57), and in general is not satisfied.

In conclusion, the circuit-device iteration scheme is preferable to the device-circuit scheme.

3.3.2 Electro-Thermal Coupling

Similarly, the coupling of heat effects with electric systems plays an important role in electric circuit simulation, see Sect. 2.2.2 and, e.g., [3, 14]. Spatial discretization of certain thermal models (e.g. for heat conduction) can yield a DAE-ODE coupling. This type of coupling is less problematic, since no coupling via old algebraic variables will occur. Therefore no contraction is needed in this case, see, e.g., [3]. In other models, e.g. with patches, the situation is a bit more complicated—for details we refer to [14].

3.3.3 Coupled System of Electric Networks and Maxwell's Magnetoquasistatic Equations and Their Properties

3.3.3.1 Introduction

Let us apply the dynamic iteration theory to the field/circuit coupling as introduced in Sect. 2.2.3.

There are two subproblems, on one hand the electric circuit and on the other hand the magnetoquasistatic field problem ("eddy current problem"). The circuit equations can abstractly by described by the semi-explicit initial value problem

$$\begin{aligned} \dot{\mathbf{y}}_c &= \mathbf{f}_c(\mathbf{y}_c, \mathbf{z}_c, \mathbf{i}_\mathrm{m}), \quad \text{with} \quad \mathbf{y}_c(0) = \mathbf{y}_{c,0} \\ \mathbf{0} &= \mathbf{g}_c(\mathbf{y}_c, \mathbf{z}_c, \mathbf{i}_\mathrm{m}), \end{aligned} \tag{3.62}$$

similar to the derivation in Sect. 3.3.1. We assume an index-1 circuit, i.e., the topological conditions as given in [17] to be fulfilled, such that

$$\frac{\partial \mathbf{g}_c}{\partial \mathbf{z}_c} \text{ is nonsingular.} \tag{3.63}$$

The unknowns are given by

$$\mathbf{y}_c := (\mathbf{q}, \boldsymbol{\phi})^\top, \quad \mathbf{z}_c := (\mathbf{u}\,\mathbf{i}_\mathrm{L}, \mathbf{i}_\mathrm{V})^\top, \quad \mathbf{i}_\mathrm{m} := (\mathbf{i}_\mathrm{str}, \mathbf{i}_\mathrm{sol})^\top$$

where $\mathbf{u}$ denotes node potentials, $\mathbf{q}$ charges, $\boldsymbol{\phi}$ fluxes and $\mathbf{i}_\mathrm{L}$, $\mathbf{i}_\mathrm{V}$ currents through inductances and voltage sources. The additional variables $\mathbf{i}_\mathrm{str}$ and $\mathbf{i}_\mathrm{sol}$ define currents through stranded and solid conductors and are treated separately since they are determined by the field model. This field model describes a relation between those currents and the voltage drops

$$\mathbf{v}_\mathrm{str} := \mathbf{A}_\mathrm{str}^\top \mathbf{u}, \qquad\qquad \mathbf{v}_\mathrm{sol} := \mathbf{A}_\mathrm{sol}^\top \mathbf{u}$$

by one common PDE for the whole domain Ω and an additional differential equation for the coupling of each stranded ($k = 1, \ldots, N_\mathrm{sol}$) and solid conductor ($l = 1, \ldots, N_\mathrm{sol}$) in the corresponding subdomains $\Omega_{\mathrm{str},k}$ and $\Omega_{\mathrm{sol},l}$ to the circuit

$$\sigma \frac{\partial \boldsymbol{A}}{\partial t} + \nabla \times (\nu \nabla \times \boldsymbol{A}) = \sum_k \boldsymbol{\chi}_{\mathrm{str},k} \, (\mathbf{i}_\mathrm{str})_k + \sum_l \sigma \boldsymbol{\chi}_{\mathrm{sol},l} \, (\mathbf{v}_\mathrm{sol})_l \tag{3.64a}$$

$$\int_\Omega \boldsymbol{\chi}_{\mathrm{str},k} \cdot \frac{\partial \boldsymbol{A}}{\partial t} \, \mathrm{d}\Omega = (\mathbf{v}_\mathrm{str})_k - (\mathbf{R}_\mathrm{str})_{k,k} \cdot (\mathbf{i}_\mathrm{str})_k \,, \tag{3.64b}$$

$$\int_\Omega \sigma \boldsymbol{\chi}_{\mathrm{sol},l} \cdot \frac{\partial \boldsymbol{A}}{\partial t} \, \mathrm{d}\Omega = (\mathbf{G}_\mathrm{sol})_{l,l} \cdot (\mathbf{v}_\mathrm{sol})_l - (\mathbf{i}_\mathrm{sol})_l \,, \tag{3.64c}$$

with Coulomb gauging, flux wall boundary and initial conditions

$$\nabla \cdot \boldsymbol{A} = 0, \qquad \boldsymbol{A} \times \mathbf{n}_{\perp} = 0 \text{ on } \partial\Omega, \qquad \boldsymbol{A} = \boldsymbol{A}_0 \text{ at } t = t_0, \tag{3.64d}$$

where $\boldsymbol{A}$ denotes the magnetic vector potential, $\mathbf{n}_{\perp}$ is the vector normal to the boundary, $\nu = \nu(\boldsymbol{A})$ the reluctivity tensor and σ the conductivity tensor vanishing on stranded conductor domains, i.e.

$$\sigma \frac{\partial \boldsymbol{A}}{\partial t}\bigg|_{\Omega_{\mathrm{str},k}} = \mathbf{0} \tag{3.65}$$

since it is assumed that the diameter of the individual strands in the those conductors is thinner that the skin depth. Each *distribution function* $\boldsymbol{\chi}_{\mathrm{str},k}$ and $\boldsymbol{\chi}_{\mathrm{sol},k}$ distributes the current in the corresponding conductor domains $\Omega_{\mathrm{str},k}$ and $\Omega_{\mathrm{sol},k}$. The diagonal matrices

$$(\mathbf{R}_{\mathrm{str}})_{k,k} = \int_{\Omega} \frac{1}{f_{\mathrm{str}}} \sigma^{-1} \boldsymbol{\chi}_{\mathrm{str},k} \cdot \boldsymbol{\chi}_{\mathrm{str},k} \mathrm{d}\Omega \quad \text{and} \quad (\mathbf{G}_{\mathrm{sol}})_{l,l} = \int_{\Omega} \sigma \boldsymbol{\chi}_{\mathrm{sol},l} \cdot \boldsymbol{\chi}_{\mathrm{sol},l} \mathrm{d}\Omega$$

describe lumped DC resistances $\mathbf{R}_{\mathrm{str}}$ for stranded conductors using the fill factor $f_{\mathrm{str}} \in (0, 1]$ and DC conductivities $\mathbf{G}_{\mathrm{sol}}$ for the solid conductors.

According to Sect. 2.2.3.3, the spatial discretization of the field PDE yields a DAE, describing a unique vector potential in time. The discrete field problem reads in the FIT notation, [12]

$$\mathbf{M}_{\sigma} \frac{\mathrm{d}}{\mathrm{d}t} \widehat{\mathbf{a}} + \mathbf{K}_{\nu}(\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} = \mathbf{Q}_{\mathrm{str}} \mathbf{i}_{\mathrm{str}} + \mathbf{M}_{\sigma} \mathbf{Q}_{\mathrm{sol}} \mathbf{v}_{\mathrm{sol}} \tag{3.66a}$$

$$\mathbf{Q}_{\mathrm{str}}^{\top} \frac{\mathrm{d}}{\mathrm{d}t} \widehat{\mathbf{a}} = \mathbf{v}_{\mathrm{str}} - \mathbf{R}_{\mathrm{str}} \mathbf{i}_{\mathrm{str}} \tag{3.66b}$$

$$\mathbf{Q}_{\mathrm{sol}}^{\top} \mathbf{M}_{\sigma} \frac{\mathrm{d}}{\mathrm{d}t} \widehat{\mathbf{a}} = \mathbf{G}_{\mathrm{sol}} \mathbf{v}_{\mathrm{sol}} - \mathbf{i}_{\mathrm{sol}}, \tag{3.66c}$$

where $\widehat{\mathbf{a}}$ denotes the discrete magnetic vector potential with consistent initial value $\widehat{\mathbf{a}}(0) = \widehat{\mathbf{a}}_0$, the mass matrix $\mathbf{M}_{\sigma}$ is symmetric positive semi-definite describing the conductivities, $\mathbf{K}_{\nu}$ is a symmetric curl-curl matrix composed of the discrete curl-operators and the reluctivities. We assume a regularization on $\mathbf{K}_{\nu}$, e.g. by the Coulomb gauging, such that

$$\widehat{\mathbf{e}}^{\top} \Big(\alpha \mathbf{M}_{\sigma} + \frac{\partial}{\partial \widehat{\mathbf{a}}} \big(\mathbf{K}_{\nu}(\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} \big) \Big) \widehat{\mathbf{e}} > 0 \quad \text{for all } \widehat{\mathbf{e}} \neq \mathbf{0} \text{ and } \alpha \neq 0. \tag{3.67}$$

which ensures a (symmetric) positive definite matrix pencil and hence allows for the application of iterative solvers, e.g. Krylov subspace methods, [13]. The matrix $\mathbf{Q} = [\mathbf{Q}_{\mathrm{sol}}, \mathbf{Q}_{\mathrm{str}}]$ is the discrete counterpart to the characteristic functions $\boldsymbol{\chi}$ in the continuous model, it imposes currents and voltages onto edges in the computational grid.

The matrices of lumped resistances and conductivities can be extracted from the discrete field model by

$$\mathbf{R}_{\mathrm{str}} := \mathbf{Q}_{\mathrm{str}}^{\top}\mathbf{M}_{\sigma,\mathrm{str}}^{+}\mathbf{Q}_{\mathrm{str}} \qquad \text{and} \qquad \mathbf{G}_{\mathrm{sol}} := \mathbf{Q}_{\mathrm{sol}}^{\top}\mathbf{M}_{\sigma}\mathbf{Q}_{\mathrm{sol}}, \tag{3.68}$$

where $\mathbf{M}_{\sigma,\mathrm{str}}^{+}$ is the pseudo inverse of a conductivity matrix with positive conductivities in the stranded conductor domains.

3.3.3.2 Coupling Analysis

To apply the schemes of Sect. 3.2 to the field/circuit coupled problem we need to verify, that the DAE index of the field problem is one, see Eq. (3.4), and that the contractivity condition (3.30) is fulfilled. Here comes the decomposition of the field system into differential and algebraic parts into play: according to the Lemma the field system (3.66) can be interpreted as the semi-explicit initial value problem

$$\begin{aligned} \dot{\mathbf{y}}_m &= \mathbf{f}_m(\mathbf{y}_m, \mathbf{z}_{ma}, \mathbf{v}_{\mathrm{c}}), \quad \text{with} \quad \mathbf{y}_m(0) = \mathbf{y}_{m,0}, \\ \mathbf{0} &= \mathbf{g}_{ma}(\mathbf{y}_m, \mathbf{z}_{ma}), \\ \mathbf{0} &= \mathbf{g}_{mb}(\mathbf{y}_m, \mathbf{z}_{ma}, \mathbf{z}_{mb}), \end{aligned} \tag{3.69}$$

where $\mathbf{y}_m := \mathscr{P}_\sigma \widehat{\mathbf{a}}$, $\mathbf{z}_{ma} := \mathscr{Q}_\sigma \widehat{\mathbf{a}}$, $\mathbf{z}_{mb} := (\mathbf{i}_{\mathrm{str}}, \mathbf{i}_{\mathrm{sol}})^{\top}$and $\mathbf{v}_{\mathrm{c}} := (\mathbf{v}_{\mathrm{str}}, \mathbf{v}_{\mathrm{sol}})^{\top}$. Now using this semi-explicit problem formulation we obtain the following result

Lemma 3.2 *The field System (3.66) is an index-1 DAEs, i.e.,*

$$\frac{\partial \mathbf{g}_m}{\partial \mathbf{z}_m} \text{ is nonsingular,}$$

for given voltages $\mathbf{v}_{\mathrm{str}}$ and $\mathbf{v}_{\mathrm{sol}}$ and the matrix pencil of the curl-curl equation (3.67) is positive definite.

Proof The DAE-indices of Systems (3.66) and (3.69) are equal, since the second system was obtained only by merely algebraic operations, proof of Lemma 2.1, hence it is sufficient to consider the more abstract system only; with the definitions

$$\mathbf{g}_m := (\mathbf{g}_{ma}, \mathbf{g}_{mb})^{\top} \qquad \text{and} \qquad \mathbf{z}_m := (\mathbf{z}_{ma}, \mathbf{z}_{mb})^{\top}$$

the index-1 requirement corresponds to the non-singularity of the Jacobian

$$\frac{\partial \mathbf{g}_m}{\partial \mathbf{z}_m} = \begin{pmatrix} \dfrac{\partial \mathbf{g}_{ma}}{\partial \mathbf{z}_{ma}} & \dfrac{\partial \mathbf{g}_{ma}}{\partial \mathbf{z}_{mb}} \\ \dfrac{\partial \mathbf{g}_{mb}}{\partial \mathbf{z}_{ma}} & \dfrac{\partial \mathbf{g}_{mb}}{\partial \mathbf{z}_{mb}} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial \mathbf{g}_{ma}}{\partial \mathbf{z}_{ma}} & \mathbf{0} \\ \dfrac{\partial \mathbf{g}_{mb}}{\partial \mathbf{z}_{ma}} & \mathbf{I} \end{pmatrix},$$

where the upper left vanishes, since there is no coupling in the algebraic part of the curl-curl equation. The lower right block $\partial \mathbf{g}_{mb}/\partial \mathbf{z}_{mb} = \mathbf{I}$ is the identity, since the function $\mathbf{g}_{mb}$ is just an assignment of the currents through solid and stranded conductors and hence trivially regular. On the other hand the upper left block coming from Eq. (2.163) reads

$$\begin{aligned} \frac{\partial \mathbf{g}_{ma}}{\partial \mathbf{z}_{ma}} &= \frac{\partial}{\partial \mathbf{z}_{ma}} \Big(\mathscr{Q}_\sigma \mathbf{K}_\nu \mathscr{P}_\sigma^\top \mathbf{y}_2 + \mathscr{Q}_\sigma \mathbf{K}_\nu \mathscr{Q}_\sigma^\top \mathbf{z}_{ma} \Big) \\ &= \frac{\partial}{\partial \widehat{\mathbf{a}}} \Big(\mathscr{Q}_\sigma \mathbf{K}_\nu (\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} \Big) \frac{\partial \widehat{\mathbf{a}}}{\partial \mathbf{z}_{ma}} = \mathscr{Q}_\sigma \frac{\partial}{\partial \widehat{\mathbf{a}}} \Big(\mathbf{K}_\nu (\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} \Big) \mathscr{Q}_\sigma^\top \end{aligned}$$

which is surely regular since the matrix pencil was assumed to be positive definite and thus the transformation

$$\mathscr{Q}_\sigma \left(\lambda \big(\mathbf{M}_\sigma + \mathbf{Q}_{\text{str}} \mathbf{R}_{\text{str}}^{-1} \mathbf{Q}_{\text{str}}^\top \big) + \frac{\partial}{\partial \widehat{\mathbf{a}}} \Big(\mathbf{K}_\nu (\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} \Big) \right) \mathscr{Q}_\sigma^\top$$

is still positive definite because $\mathscr{Q}_\sigma^\top$ has full rank and the mass matrix does not contribute by construction to this submatrix

$$\mathscr{Q}_\sigma \left(\mathbf{M}_\sigma + \mathbf{Q}_{\text{str}} \mathbf{R}_{\text{str}}^{-1} \mathbf{Q}_{\text{str}}^\top \right) \mathscr{Q}_\sigma^\top = \mathbf{0} \, .$$

Hence we obtain the positive definiteness of $\frac{\partial}{\partial \widehat{\mathbf{a}}} \Big(\mathbf{K}_\nu (\widehat{\widehat{\mathbf{b}}}) \widehat{\mathbf{a}} \Big)$, furthermore this shows the regularity of the minor $\partial \mathbf{g}_{ma}/\partial \mathbf{z}_{ma}$ and thus we have finally proven System (3.66) being an index-1 DAE. □

Theorem 3.3 *The field/circuit coupled system* (3.62)+(3.69)*, i.e.,*

$$\begin{aligned} \dot{\mathbf{y}}_c &= \mathbf{f}_c(\mathbf{y}_c, \mathbf{z}_c, \boxed{\mathbf{z}_m}), & &\textit{and} & \dot{\mathbf{y}}_m &= \mathbf{f}_m(\mathbf{y}_m, \mathbf{z}_m, \boxed{\mathbf{z}_c}), \\ \mathbf{0} &= \mathbf{g}_c(\mathbf{y}_c, \mathbf{z}_c, \boxed{\mathbf{z}_m}), & & & \mathbf{0} &= \mathbf{g}_m(\mathbf{y}_m, \mathbf{z}_m), \end{aligned}$$

is index-1, if the circuit fulfills the index-1 assumption (3.63)*, and the matrix pencil of the underlying curl-curl equation* (3.67) *is positive definite.*

Proof We proceed similarly to the proof of Lemma 3.2, where we inspected the algebraic constraints for the field DAE. For the algebraic constraints and variables of the whole coupled system

$$\mathbf{g} := (\mathbf{g}_c, \mathbf{g}_m)^\top = (\mathbf{g}_c, \mathbf{g}_{ma}, \mathbf{g}_{mb})^\top \quad \text{and} \quad \mathbf{z} := (\mathbf{z}_c, \mathbf{z}_m)^\top = (\mathbf{z}_c, \mathbf{z}_{ma}, \mathbf{z}_{mb})^\top,$$

follows analogously the Jacobian

$$\frac{\partial \mathbf{g}}{\partial \mathbf{z}} = \begin{pmatrix} \frac{\partial \mathbf{g}_c}{\partial \mathbf{z}_c} & \frac{\partial \mathbf{g}_c}{\partial \mathbf{z}_m} \\ \frac{\partial \mathbf{g}_m}{\partial \mathbf{z}_c} & \frac{\partial \mathbf{g}_m}{\partial \mathbf{z}_m} \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{g}_c}{\partial \mathbf{z}_c} & \frac{\partial \mathbf{g}_c}{\partial \mathbf{z}_m} \\ \mathbf{0} & \frac{\partial \mathbf{g}_m}{\partial \mathbf{z}_m} \end{pmatrix},$$

which is nonsingular, because the index-1 assumption for the circuit guarantees the regularity of $\partial \mathbf{g}_c / \partial \mathbf{z}_c$ and finally Lemma 3.2 gives the regularity of $\partial \mathbf{g}_m / \partial \mathbf{z}_m$. □

To allow the coupling of already existing simulator packages, the coupled system (3.62)+(3.64) is split such both that sub-problems can be computed independently. The dynamic iteration method will call each simulator to integrate the sub-problem on a time window and then exchange the obtained voltages and currents at the synchronization points. During the computation of a sub-problem on a window the data of the other system is frozen and represented by a source. Since each system describes for current/voltage relations, we have to decide which quantities are considered as known for each branch and conductor. This question is crucial for the field/circuit coupling since the DAE-index of the field system and hence the applicability of the dynamic iteration method depends on this decision:

Corollary 3.3 *The field system* (3.66) *is index-1, if all voltages* ($\mathbf{v}_{\text{sol}}$, $\mathbf{v}_{\text{str}}$) *are given, and in all other cases, i.e., given* ($\mathbf{i}_{\text{sol}}$, $\mathbf{v}_{\text{str}}$), ($\mathbf{v}_{\text{sol}}$, $\mathbf{i}_{\text{str}}$) *or* ($\mathbf{i}_{\text{sol}}$, $\mathbf{i}_{\text{str}}$)*, it is at least index-2.*

Proof The first part for given voltages is proven by Lemma 3.2, since the currents $\mathbf{z}_{mb}$ in (3.69) can be obtained by just evaluating the algebraic equation

$$\mathbf{0} = \mathbf{g}_{mb}(\mathbf{z}_m, \mathbf{z}_{ma}, \mathbf{z}_{mb})$$

but if instead a current is prescribed, then the function $\mathbf{f}_2$ depends on an unknown voltage ($\mathbf{v}_{\text{str}}$ or $\mathbf{v}_{\text{sol}}$) and hence the coupling equation $\mathbf{g}_{mb}$ must be differentiated once with respect to time to obtain a hidden algebraic constraint for the missing voltage, such that the overall system is at least index-2. □

That it is just index-2 has been shown for the case of given currents in solid conductors in [37] and more generally in [34] was proven, that (3.66) is in fact an index-2 Hessenberg system, [8], with some additional algebraic (index-1) equations due to the singularity of the mass matrix.

3.3.3.3 Field-Circuit Scheme

Now having obtained semi-explicit index-1 formulations of both sub-systems, (3.62) and (3.69), we give a more abstract description of the coupling that fits into the framework of dynamic iteration methods in Sect. 3.2.2, i.e., System (3.4).

On hand we have the circuit DAE-IVP

$$\begin{aligned}
\dot{\mathbf{y}}_c &= \mathbf{f}_c(\mathbf{y}_c, \mathbf{z}_c, \boxed{\mathbf{z}_{mb}}), & \mathbf{y}_c(0) &= \mathbf{y}_{c,0}, & \mathbf{y}_c &:= \left(\mathbf{q}^\top, \boldsymbol{\phi}^\top\right)^\top,\\
\mathbf{0} &= \mathbf{g}_c(\mathbf{y}_c, \mathbf{z}_c, \boxed{\mathbf{z}_{mb}}), & & & \mathbf{z}_c &:= \left(\mathbf{u}^\top, \mathbf{i}_{\mathrm{L}}^\top, \mathbf{i}_{\mathrm{V}}^\top\right)^\top,
\end{aligned}$$

and on the other hand the field DAE-IVP

$$\begin{aligned}
\dot{\mathbf{y}}_m &= \mathbf{f}_m(\mathbf{y}_m, \mathbf{z}_{ma}, \boxed{\mathbf{z}_c}), & \mathbf{y}_m(0) &= \mathbf{y}_{m,0}, & \mathbf{y}_m &= \mathscr{P}_\sigma \widehat{\mathbf{a}},\\
\mathbf{0} &= \mathbf{g}_{ma}(\mathbf{y}_m, \mathbf{z}_{ma}), & & & \mathbf{z}_{ma} &= \mathscr{Q}_\sigma \widehat{\mathbf{a}},\\
\mathbf{0} &= \mathbf{g}_{mb}(\mathbf{y}_m, \mathbf{z}_{ma}, \mathbf{z}_{mb}), & & & \mathbf{z}_{mb} &= \left(\mathbf{i}_{\mathrm{str}}^\top, \mathbf{i}_{\mathrm{sol}}^\top\right)^\top
\end{aligned}$$

where a slight abuse of notation is introduced when inserting all algebraic circuit unknowns $\mathbf{z}_c$ into $\mathbf{f}_{ma}$ instead of only the actually needed voltage drops $\mathbf{v}$.

Let us discuss a dynamic iteration of Gauss-Seidel type on the time interval $[0, t_e]$, with $1 \leq n \leq N$ windows $[t_n, t_{n+1}] \subset [0, t_e]$ and adequate initial values for each window

$$\begin{pmatrix} \mathbf{y}_{c,n} \\ \mathbf{y}_{m,n} \end{pmatrix} := \begin{pmatrix} \mathbf{y}_c(t_n) \\ \mathbf{y}_m(t_n) \end{pmatrix} =: \mathbf{y}(t_n).$$

We start each iteration with the integration of the field DAE-IVP. It depends on data from the circuit DAE-IVP (denoted by $\mathbf{y}_c$, $\mathbf{z}_c$). These missing data, i.e., the voltage drops $\mathbf{v}$ at the conductors, are unknown at start time. Hence we extrapolate the initial value to the current time. We choose the following constant extrapolation of the differential variables

$$\begin{pmatrix} \mathbf{y}_{c,n}^{(0)} \\ \mathbf{y}_{m,n}^{(0)} \end{pmatrix} = \Phi_{\mathbf{y},n}(\mathbf{y}|_{(t_{n-1}, t_n)}) := \mathbf{y}(t_n), \tag{3.70}$$

from which a consistent supplement $\mathbf{z}_{1,n}^{(0)}$ and $\mathbf{z}_{2,n}^{(0)}$ for the algebraic variables is obtained. Providing this data the field system can be solved for the first time ($k = 1$)

on the time window $[t_n, t_{n+1}]$

$$\begin{aligned}\dot{\mathbf{y}}_m^{(k)} &= \mathbf{f}_m(\mathbf{y}_m^{(k)}, \mathbf{z}_{ma}^{(k)}, \boxed{\mathbf{z}_c^{(k-1)}}), \quad \mathbf{y}_m^{(k)}(0) = \mathbf{y}_{m,n}, \\ \mathbf{0} &= \mathbf{g}_{ma}(\mathbf{y}_m^{(k)}, \mathbf{z}_{ma}^{(k)}), \\ \mathbf{0} &= \mathbf{g}_{mb}(\mathbf{y}_m^{(k)}, \mathbf{z}_{ma}^{(k)}, \mathbf{z}_{mb}^{(k)}).\end{aligned} \tag{3.71a}$$

Having obtained a first algebraic iterate $\mathbf{z}_{mb}^{(k)}$ (at $k = 1$) for the currents $\mathbf{i}_{\mathrm{str}}$ and $\mathbf{i}_{\mathrm{sol}}$ we may continue to solve the circuit subsystem

$$\begin{aligned}\dot{\mathbf{y}}_c^{(k)} &= \mathbf{f}_c(\mathbf{y}_c^{(k)}, \mathbf{z}_c^{(k)}, \boxed{\mathbf{z}_{mb}^{(k)}}), \quad \mathbf{y}_c^{(k)}(0) = \mathbf{y}_{c,n}, \\ \mathbf{0} &= \mathbf{g}_c(\mathbf{y}_c^{(k)}, \mathbf{z}_c^{(k)}, \boxed{\mathbf{z}_{mb}^{(k)}}).\end{aligned} \tag{3.71b}$$

After the first iteration the functions $\mathbf{y}^{(k)}$ and $\mathbf{z}^{(k)}$ ($k = 1$) are obtained and we may restart the scheme for $k + 1$ until k_n sweeps of the n-th time window are completed. After that we proceed to the next time window ($n + 1$) and start again with the constant extrapolation (3.11) and the following k_{n+1} Gauss-Seidel iterations, until the end of the integration interval $[0, t_e]$ is reached.

In this application of the Gauss-Seidel-Scheme the splitting functions, as introduced in Eqs. (3.13), are defined as the mappings

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{pmatrix} \mathbf{f}_c(\mathbf{y}_c^{(k)}, \mathbf{z}_c^{(k)}, \mathbf{z}_{mb}^{(k)}) \\ \mathbf{f}_m(\mathbf{y}_m^{(k)}, \mathbf{z}_{ma}^{(k)}, \mathbf{z}_c^{(k-1)}) \end{pmatrix}$$

and

$$\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}) := \begin{pmatrix} \mathbf{g}_c(\mathbf{y}_c^{(k)}, \mathbf{z}_c^{(k)}, \mathbf{z}_{mb}^{(k)}) \\ \mathbf{g}_{ma}(\mathbf{y}_m^{(k)}, \mathbf{z}_{ma}^{(k)}) \\ \mathbf{g}_{mb}(\mathbf{y}_m^{(k)}, \mathbf{z}_{ma}^{(k)}, \mathbf{z}_{mb}^{(k)}) \end{pmatrix}$$

where $\mathbf{G}$ does not depend on an old algebraic variable $\mathbf{z}^{(k-1)}$. Therefore Corollary 3.1 applies here

Corollary 3.4 *The dynamic iteration scheme* (3.71) *is is unconditionally stable on the time interval* $[0, t_e]$.

By contrast, if we consider the circuit-field Gauss-Seidel scheme or a Jacobi-Scheme, we have to deal with the partial derivatives as in the case of the device-circuit scheme in Sect. 3.3.

3.3.3.4 Multimethod and Multirate Benefits

Besides advantages in software engineering, there are other benefits for the coupling of simulation packages: the most important are benefits due the use of problem-specific methods for time integration (multimethod) and the possibility of different step sizes (multirate) for each subproblem. Thus adaptive time stepping schemes will apply automatically the time step sizes, that are inherently given by the subproblem and not the minimum of all those step size as in the monolithic approach. This will yield a computational more efficient integration.

The benefit of the multimethod approach is obviously present since the packages for field simulation are commonly applying the implicit Euler scheme or implicit Runge-Kutta schemes for time integration, [28], while circuit simulators are typically based on schemes from the BDF family, [21].

The advantage due multirate behavior depend highly on the specific configuration of the problem considered, since different time scales do not occur in the field/circuit coupling as natural as in the thermal coupling (Sect. 3.3.2), where the effects are clearly from multiphysics. In contrast to this, the described phenomena of the field/circuit coupling originate all from Maxwell's equations and hence there is no guarantee of multirate effects. Even if present, e.g. due to switches or filters, the partition of the subsystems according to the network DAE and field PDE model does not necessarily correspond to time constants of different magnitude. Moreover a partition into fast and slow switching components would require to split the circuit at arbitrary nodes and could hence destroy the advantages of the simulator coupling approach.

Anyhow if the circuit contains only a small number of devices that are active at a time, while others remain latent and the field model belongs to such a latent part, then the computational expensive solution of the PDE can be obtained using less time steps than the circuit solution requires. This *weak* coupling will be naturally exploited by the dynamic iteration method, if the step sizes for the time integration of the sub-problems are chosen accordingly (or are automatically determined by an adaptive time integrator) and increase its efficiency when compared to a single-rate integration method.

For such configurations an efficient special case of the dynamic iteration method is the *multirate co-simulation*, where only one sweep ($k_n = 1$) is made, but obviously smaller synchronization steps have to be chosen.

3.3.3.5 Numerical Example

Let us discuss a classical example from engineering: a transformer is excited at its primary coil by an alternating voltage source with $\mathbf{v}_{\text{eff}} = 250\,\text{V}$ at $f = 50\,\text{Hz}$ and is connected to a rectifier circuit at its secondary coil with a load resistance of $R_{\text{load}} = 100\,\Omega$, Fig. 3.1a. The diodes are described by Shockley's model with $I_s = 10\mu\text{A}$. The transformer is represented by a PDE in 3D, discretized by `EM`

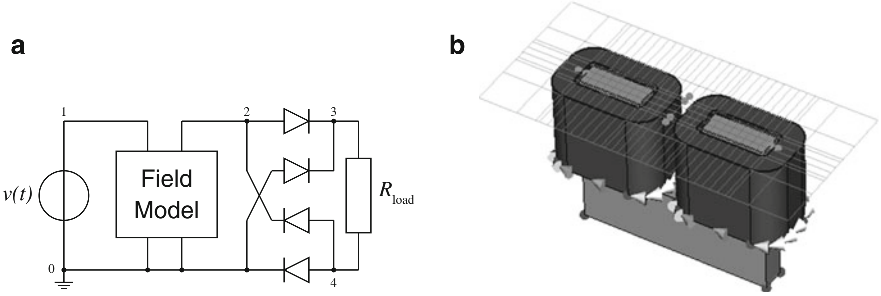


Fig. 3.1 (**a**) Example circuit: rectifier. (**b**) Field model: transformer

`Studio` from CST Software,[2] where each coil is connected to the circuit using the a stranded conductor model.

The simulation software was implemented within the `COMSON DP` and applies either the classic monolithic strategy or the dynamic iteration method by using Gauss-Seidel's scheme. Simulation results are presented in Fig. 3.2.

3.3.3.6 Summary

In this section we shown how nonlinear index-analysis of DAEs can be used to prove the convergence of dynamic iteration methods applied coupled problems. In the case of Maxwell's magnetoquasistatic equations coupled to electric circuits we find that there is no dependence of the algebraic equations on previous algebraic iterates. This guarantees an index-1 problem in the case of monolithic coupling and furthermore proofs convergence and stability of the proposed dynamic iteration scheme. To obtain this result it is not even necessary to validating the contractivity condition given in Sect. 3.2.3.

3.4 Coupled Numerical Simulations of the Thermal Effects in Silicon Devices

In this section we analyze the discretization of the model presented in Sect. 2.2.4, describing the coupling between the transport of electrons and the heating of the crystal lattice. Results of the simulation of a MOSFET with a nanoscale channel are presented and the influence of the thermal effects on the electrical performance is analyzed. This section is based on reference [30, 31] where the interested reader can find more details.

[2] see http://www.cst.com/

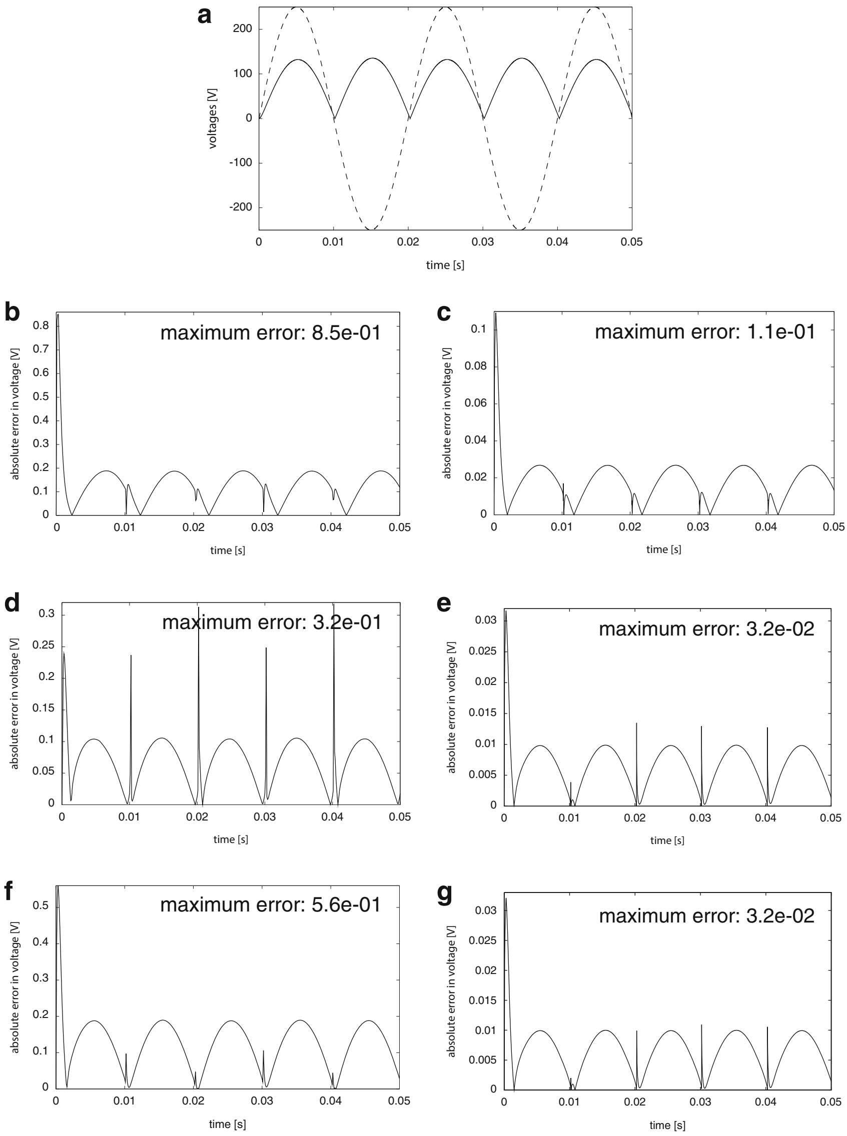


Fig. 3.2 *Numerical Example.* Error plots with respect to the reference solution. (**a**) Input (*dashed*) and output (*solid*) voltages of the reference solution, obtained by monolithic simulation with step size $H = 1e - 5$. (**b**) Error in output voltage in multirate co-simulation with window size $H = 1e - 4$. (**c**) Error in output voltage in multirate co-simulation with window size $H = 1e - 5$. (**d**) Error in output voltage in dynamic iteration with 3 sweeps and window size $H = 1e - 4$. (**e**) Error in output voltage in dynamic iteration with 3 sweeps and window size $H = 1e - 5$. (**f**) Error in output voltage in monolithic simulation with step size $H = 1e - 4$. (**g**) Error in output voltage in monolithic simulation with step size $H = 1e - 5$

In addition to the model presented in Chap. 2, also the holes will be included with a simple drift-diffusion equation.

The complete mathematical model is given by the equations

$$\frac{\partial n}{\partial t} + \operatorname{div}(n\,\mathbf{V}) = -R, \tag{3.72}$$

$$\frac{\partial p}{\partial t} + \operatorname{div}\left(p\,\mathbf{V_p}\right) = -R, \tag{3.73}$$

$$\frac{\partial (nW)}{\partial t} + \operatorname{div}(n\,\mathbf{S}) + nq\mathbf{V}\cdot\nabla\phi = nC_W, \tag{3.74}$$

$$\rho c_V \frac{\partial T_L}{\partial t} - \operatorname{div}\left[K(T_L)\nabla T_L\right] = H, \tag{3.75}$$

$$\mathbf{E} = -\nabla\phi, \quad \epsilon\Delta\phi = -q(N_D - N_A - n + p), \tag{3.76}$$

with n and p the electron and holes density respectively, W the electron energy, T_L the lattice temperature, ϕ the electrostatic potential and $\mathbf{E} = -\nabla\phi$ the electric field. N_D and N_A are the density of donors and acceptors respectively (assumed as known function of the position). q is the elementary charge, ρ the silicon density, c_V the specific heat, C_W the energy production term, which can be written in a relaxation form as

$$C_W = -\frac{W - W_0}{\tau_W}, \tag{3.77}$$

with $W_0 = 3/2k_B T_L$ and $\tau_W(W)$ the energy relaxation time. k_B is the Boltzmann constant and ϵ is the dielectric constant.

The closure relations for the electron velocity $\mathbf{V}$, the energy flux $\mathbf{S}$, the thermal conductivity $K(T_L)$ and the crystal energy production term H have been obtained in [32, 33] by employing MEP and are reported in Chap. 2. The holes are described by a standard drift-diffusion model with constant mobility. $\mathbf{V_p}$ is the velocity of holes.

Since the electron production terms are slowly changing with respect to $k_B T_L$, we adopt the simplification that they are evaluated at $T_L = 300\,\mathrm{K}$.

The phonon energy production is given by

$$H = -(1 + P_S)\,n\,C_W + P_S\;\mathbf{J}\cdot\mathbf{E}, \tag{3.78}$$

where $P_S = -c^2\,\tau_R\,c_{12}^{(p)}$ plays the role of a thermopower coefficient and τ_R is the phonon relaxation time in resistive processes.

R is the generation-recombination term (see [35] for a complete review) which splits into the Shockley-Read-Hall (SRH) and the Auger contribution (AU) $R = R^{SRH} + R^{AU}$ where

$$R^{SRH} = \frac{np - n_i^2}{\tau_p(n + n_1) + \tau_n(p + p_1)}, \quad R^{AU} = \left(C_{cn}n + C_{cp}p\right)(np - n_i^2), \tag{3.79}$$

We will take the values $C_{cn} = 2.8 \times 10^{-31}$ cm ^{6} s^{-1} and $C_{cp} = 9.9 \times 10^{-32}$ cm ^{6} s^{-1}. In our numerical experiments we set $n_1 = p_1 = n_i$, n_i being the intrinsic concentration. The expressions of τ_p and τ_n we will use are [35]

$$\tau_n = \frac{\tau_{n0}}{1 + \frac{N_D(x)+N_A(x)}{N_n^{ref}}}, \quad \tau_p = \frac{\tau_{p0}}{1 + \frac{N_D(x)+N_A(x)}{N_p^{ref}}}, \tag{3.80}$$

where $\tau_{n0} = 3.95 \times 10^{-4}$, $\tau_{p0} = 3.25 \times 10^{-5} s$, $N_n^{ref} = N_p^{ref} = 7.1 \times 10^{15}\text{cm}^{-3}$.

At the source and drain contacts the Robin boundary condition

$$-k_L \frac{\partial T_L}{\partial n} = R_{th}^{-1}(T_L - T_{env}), \tag{3.81}$$

is assumed, R_{th} being the thermal resistivity of the contact and T_{env} the environment temperature. We use no-flux condition for the temperature on the lateral boundary and oxide silicon interface and Dirichlet condition at the bulk contact. The electron energy at the source, drain and bulk contacts is set equal to the lattice energy. The other boundary conditions needed for integrating the Mosfet model are described in [29].

3.4.1 The Numerical Method

The crystal lattice temperature T_L changes much slower than other variables. For instance the typical relaxation time for the temperature in our simulations is in the order of thousand picoseconds, while relaxation time of the other fields is in the order of picoseconds. We exploit this double-scale behavior by applying a variant of the multirate integration scheme [18, 20] which is a popular choice in coupled electro-thermal circuit simulation [5]. For the simulation of the transient response of the model we solve the balance equations by adopting the following multirate integration scheme:

- Step 1. We first integrate the balance equations for electrons and holes with the crystal lattice energy and the electric field frozen at the time step $k-1$. This gives the density of the electrons and holes and the electron energy at the time step k and schematically can be written as

$$\frac{\partial \mathcal{U}^k}{\partial t} + F(\mathcal{U}^k, \phi^{k-1}, T_L^{k-1}) = 0, \tag{3.82}$$

  with $\mathcal{U} = (n, p, W)$. Here $k = 1, \ldots, N$ is the index of the integration interval $[t_{k-1}, t_k]$, with $t_k = t_{k-1} + \Delta t$, Δt being the time size of the synchronization window.

- Step 2. We integrate the lattice energy balance equation with n and W given by the step 1

$$\rho c_V \frac{\partial T_L^k}{\partial t} - \operatorname{div}\left[K(T_L^k)\nabla T_L^k\right] = H(\mathscr{U}^k, T_L^k) \tag{3.83}$$

along with the Poisson equation with $n = n^k$ and $p = p^k$.

For steps 1 and 2 different time steps for the numerical integration over the interval $[t_{k-1}, t_k]$ are used. Typically the time step for integration of (3.83) we can use is 100 times larger than the time step for (3.82).

This sequence can be considered as steps of a splitting technique [26] and we expect that such a numerical scheme is a stable first-order approximation with respect to time, as confirmed by the numerical experiments presented in the next section.

3.4.1.1 Step 1

The numerical scheme is based on an exponential fitting like that employed in the Scharfetter-Gummel scheme for the drift-diffusion model of semiconductors. The basic idea is to split the particle and energy density currents as the difference of two terms. Each of them is written by introducing suitable mean mobilities in order to get expressions of the currents similar to those arising in other energy-transport models known in literature [6, 7, 11, 25, 36]. A simple explicit discretization in time with constant time step proves satisfactorily efficient and avoids the problem related to the high nonlinear coupling of the discretized equations of [27]. The equations are spatially discretized on a regular grid. The details of the numerical scheme can be found in [29]. Here a brief account is given.

For the sake of simplicity, the numerical method is presented only for the electron part, putting equal to zero the generation-recombination term. The inclusion of holes and the coupling with electrons is performed straightforwardly in an explicit way.

First the current density $\mathbf{J} = n\mathbf{V}$ and the energy-flux density $\mathbf{Z} = n\,\mathbf{S}$ are rewritten as

$$\mathbf{J} = \mathbf{J}^{(1)} - \mathbf{J}^{(2)}, \quad \mathbf{Z} = \mathbf{Z}^{(1)} - \mathbf{Z}^{(2)} \tag{3.84}$$

and then each term is put into a drift-diffusion form

$$\mathbf{J}^{(1)} = \frac{c_{22}}{D}\left[\nabla(nU) - qn\nabla\phi\right], \quad \mathbf{J}^{(2)} = \frac{c_{12}}{D}\left[\nabla(nF) - qn\frac{F}{U}\nabla\phi\right], \tag{3.85}$$

$$\mathbf{Z}^{(1)} = \frac{c_{11}}{D}\left[\nabla(nF) - qn\frac{F}{U}\nabla\phi\right], \quad \mathbf{Z}^{(2)} = \frac{c_{12}}{D}\left[\nabla(nU) - qn\nabla\phi\right], \tag{3.86}$$

with $D = c_{11}c_{22} - c_{12}c_{21}$.

We introduce the grid points (x_i, y_j) with $x_{i+1} - x_i = h$ = constant and $y_{j+1} - y_j = k$ = constant, and the middle points $(x_i, y_{j\pm1/2}) = (x_i, y_j \pm k/2)$ and $(x_{i\pm1/2}, y_j) = (x_i \pm h/2, y_j)$. A uniform time step Δt is used and we set $u^l_{i,j} = u(x_i, y_j, l\,\Delta t)$.

By indicating with J_x and J_y the x and y component of the current density $\mathbf{J}$ and by Z_x and Z_y the x and y component of $\mathbf{Z}$, we discretize the balance equations (3.72) and (3.74) up to terms of order $O(h^2, k^2, \Delta t)$ in the bidimensional case as

$$\frac{n_i^{l+1} - n_i^l}{\Delta t} + \frac{(J_x)_{i+1/2,j} - (J_x)_{i-1/2,j}}{h} + \frac{(J_y)_{i,j+1/2} - (J_y)_{i,j-1/2}}{k} = 0, \tag{3.87}$$

$$\begin{aligned} &\frac{(n\,W)_i^{l+1} - (n\,W)_i^l}{\Delta t} + \frac{(Z_x)_{i+1/2,j} - (Z_x)_{i-1/2,j}}{h} + \frac{(Z_y)_{i,j+1/2} - (Z_y)_{i,j-1/2}}{k} + \\ &- q\,\frac{(J_x)_{i+1/2,j} + (J_x)_{i-1/2,j}}{2}\,\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h} \\ &- q\,\frac{(J_y)_{i,j+1/2} + (J_y)_{i,j-1/2}}{2}\,\frac{\phi_{i,j+1} - \phi_{i,j-1}}{2k} + n_{i,j}\,\frac{W_{i,j} - W_0}{(\tau_W)_{i,j}} = 0. \end{aligned} \tag{3.88}$$

The variables without temporal index must be considered evaluated at time level l.

In order to evaluate the components of the currents in the middle points, let us consider the sets

$$I_{i+1/2,j} = [x_i, x_{i+1}] \times [y_{j-1/2}, y_{j+1/2}], \quad I_{i,j+1/2} = [x_{i-1/2}, x_{i+1/2}] \times [y_j, y_{j+1}]$$

and expand $J_x^{(r)}$, r= 1, 2, in Taylor's series in $I_{i+1/2,j}$

$$J_x^{(r)}(x,y) \approx (J_x^{(r)})_{i+1/2,j} + (x - x_{i+1/2})\left(\frac{\partial J_x^{(r)}}{\partial x}\right)_{i+1/2,j} + (y - y_j)\left(\frac{\partial J_x^{(r)}}{\partial y}\right)_{i+1/2,j}$$

and $J_y^{(r)}$, r =1, 2, in Taylor's series in $I_{i,j+1/2}$

$$J_y^{(r)}(x,y) \approx (J_y^{(r)})_{i,j+1/2} + (x - x_i)\left(\frac{\partial J_y^{(r)}}{\partial x}\right)_{i,j+1/2} + (y - y_{j+1/2})\left(\frac{\partial J_y^{(r)}}{\partial y}\right)_{i,j+1/2}.$$

First, we introduce $U_T = U(W)/q$, which plays the role of a thermal potential (see [29] for more details) and indicate by $\overline{U}_T$ its piecewise constant approximation, which is given by $\overline{U}_T = \frac{U(W_{i,j}) + U(W_{i+1,j})}{2q}$ in the cell $I_{i+1/2,j}$ and by

$\overline{U}_T = \frac{U(W_{i,j+1})+U(W_{i,j})}{2q}$ in the cell $I_{i,j+1/2}$. Then we introduce the *local* mobilities

$$g_{11} = -\frac{\overline{c}_{22}}{\overline{D}} nU, \quad g_{12} = -\frac{\overline{c}_{12}}{\overline{D}} nF, \quad g_{21} = -\frac{\overline{c}_{11}}{\overline{D}} nF, \quad g_{22} = -\frac{\overline{c}_{12}}{\overline{D}} nU, \tag{3.89}$$

where $\overline{c}_{pq}$ is a piecewise constant approximation of $c_{pq} p, q = 1, 2$, given by $\overline{c}_{pq} = c_{pq}\left(\frac{W_{i,j}+W_{i+1,j}}{2}\right)$ in the cell $I_{i+1/2,j}$ and by $\overline{c}_{pq} = c_{pq}\left(\frac{W_{i,j}+W_{i,j+1}}{2}\right)$ in the cell $I_{i,j+1/2}$, and, as in [15], the *local* Slotboom variables

$$s_{kr} = \exp\left(-\phi/\overline{U}_T\right) g_{kr} \quad k, r = 1, 2$$

that satisfy

$$\nabla s_{1r} \simeq -\exp\left(-\phi/\overline{U}_T\right) \mathbf{J}^{(r)}, \quad \nabla s_{2r} \simeq -\exp\left(-\phi/\overline{U}_T\right) \mathbf{H}^{(r)} \quad r = 1, 2. \tag{3.90}$$

From the x component of $(3.90)_1$, one has

$$\frac{\partial s_{1r}(x, y_j)}{\partial x} \simeq -\exp\left(-\phi/\overline{U}_T\right) J_x^{(r)}(x, y_j) =$$
$$-\exp\left(-\phi/\overline{U}_T\right)\left\{(J_x^{(r)})_{i+1/2,j} + (x - x_{i+1/2})\left(\frac{\partial J_x^{(r)}}{\partial x}\right)_{i+1/2,j} + o(\Delta x, \Delta y)\right\},$$

which, after integration over $[x_i, x_{i+1}]$ and some algebra, gives

$$\begin{aligned}(J_x^{(r)})_{i+1/2,j} &= -z_{i+1/2,j} \coth z_{i+1/2,j} \frac{(g_{1r})_{i+1,j} - (g_{1r})_{i,j}}{h} \\ &\quad + z_{i+1/2,j} \frac{(g_{1r})_{i+1,j} + (g_{1r})_{i,j}}{h}, \quad r = 1, 2\end{aligned} \tag{3.91}$$

where $z_{i+1/2,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{2\overline{U}_T}$.

Likewise by evaluating the y component of $(3.90)_2$ and integrating over $[y_j, y_{j+1}]$ we find

$$\begin{aligned}(J_y^{(r)})_{i,j+1/2} &= -z_{i,j+1/2} \coth z_{i,j+1/2} \frac{(g_{1r})_{i,j+1} - (g_{1r})_{i,j}}{k} \\ &\quad + z_{i,j+1/2} \frac{(g_{1r})_{i,j+1} + (g_{1r})_{i,j}}{k}, \quad r = 1, 2\end{aligned} \tag{3.92}$$

where $z_{i,j+1/2} = \frac{\phi_{i,j+1}-\phi_{i,j}}{2\overline{U}_T}$. With the same procedure the following discrete expression for the components of the energy flux are obtained

$$\begin{aligned}(H_x^{(r)})_{i+1/2,j} &= -z_{i+1/2,j} \coth z_{i+1/2,j} \, \frac{(g_{2r})_{i+1,j} - (g_{2r})_{i,j}}{h} \\ &\quad + z_{i+1/2,j} \frac{(g_{2r})_{i+1,j} + (g_{2r})_{i,j}}{h}, \end{aligned} \tag{3.93}$$

$$\begin{aligned}(H_y^{(r)})_{i,j+1/2} &= -z_{i,j+1/2} \coth z_{i,j+1/2} \, \frac{(g_{2r})_{i,j+1} - (g_{2r})_{i,j}}{k} \\ &\quad + z_{i,j+1/2} \frac{(g_{2r})_{i,j+1} + (g_{2r})_{i,j}}{k}, \quad r = 1,2. \end{aligned} \tag{3.94}$$

The error in formulas (3.91)–(3.94) is $O(h,k)$.

The Poisson equation is solved by replacing it with

$$\phi_t - \mathrm{div}\,(\epsilon \nabla \phi) = q(N_D - N_A - n). \tag{3.95}$$

The solution of (3.95) as $t \mapsto +\infty$ is the same as that of the original Poisson equation, at least in the smooth case.

If we introduce a time step $\Delta\hat{t}$ and set $\phi_{ij}^r = \phi(x_i, y_j, r\Delta\hat{t})$, (3.95) can be discretized in an explicit way as

$$\begin{aligned}\phi_{ij}^{r+1} &= \phi_{ij}^r + \epsilon\Delta\hat{t}\left[\frac{1}{h^2}\left(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}\right) + \frac{1}{k^2}\left(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}\right)\right. \\ &\quad \left. + q(C_{i,j} - n_{i,j})\right] \end{aligned} \tag{3.96}$$

with the notable advantage to take easily into account the different types of boundary conditions, that will be considered in more detail in the next sections. The price to pay is that at each time step, we need to reach the stationary state of (3.95) by using a time step satisfying the CFL condition, usual for parabolic equations,

$$\Delta\hat{t} \leq \frac{1}{2}\frac{1}{\frac{1}{h^2} + \frac{1}{k^2}}.$$

However the computational effort is comparable with that required by direct methods.

3.4.1.2 Step 2

A coordinate splitting technique [26] is used for the solution of the lattice energy equation for the variable $u = k_B T$ with time step Δt_T. The splitting technique allows an efficient usage of stable implicit time schemes. The procedure contains two steps with the two sub operators

$$\rho c_V \frac{u^{n+1/2} - u^n}{\Delta t_T} = \frac{\partial}{\partial x}\left[K(T_L^n)\frac{\partial u^{n+1/2}}{\partial x}\right] + \frac{k_B}{2} H^{n+1/2}, \tag{3.97}$$

$$\rho c_V \frac{u^{n+1} - u^{n+1/2}}{\Delta t_T} = \frac{\partial}{\partial y}\left[K(T_L^n)\frac{\partial u^{n+1}}{\partial y}\right] + \frac{k_B}{2} H^{n+1}. \tag{3.98}$$

This scheme is absolutely stable and approximates the equation of the lattice energy with first order accuracy in time. For the approximation of the spatial derivatives, the standard stencil with three points has been chosen. For instance, the approximation of (3.98) is the following

$$\begin{aligned}\rho c_V u_{i,j}^{n+1} = \rho c_V u_{i,j}^{n+1/2} + \frac{\Delta t_T}{k^2}\Bigg[&\frac{\tilde{K}_{i,j} + \tilde{K}_{i,j+1}}{2}(u_{i,j+1}^{n+1} - u_{i,j}^{n+1}) - \\ &\frac{\tilde{K}_{i,j} + \tilde{K}_{i,j-1}}{2}(u_{i,j}^{n+1} - u_{i,j-1}^{n+1})\Bigg] \\ &+ \frac{k_B}{2}\frac{\Delta t_T}{\tau_W}(1 + P_S)n_{i,j}^{n+1}\left(W_{ij}^{n+1} - \frac{3}{2}u_{i,j}^{n+1}\right) + \frac{k_B}{2}\Delta t_T\, J_{i,j}^{n+1}\, E_{i,j}^{n+1},\end{aligned}$$

where $\tilde{K}_{i,j} = K(T_{Li,j})$. Of course such a discretization is valid in the interior points of the mesh.

The Robin boundary condition (3.81) is approximated as

$$-k_L \frac{u_{i,1}^{n+1} - u_{i,0}^{n+1}}{k} = R_{th}^{-1}(u_{i,0}^{n+1} - k_B T_{env}). \tag{3.99}$$

Here we have assumed that at the portion of boundary where the Robin condition holds, one has $j = 0$ and the closest interior points have $j = 1$.

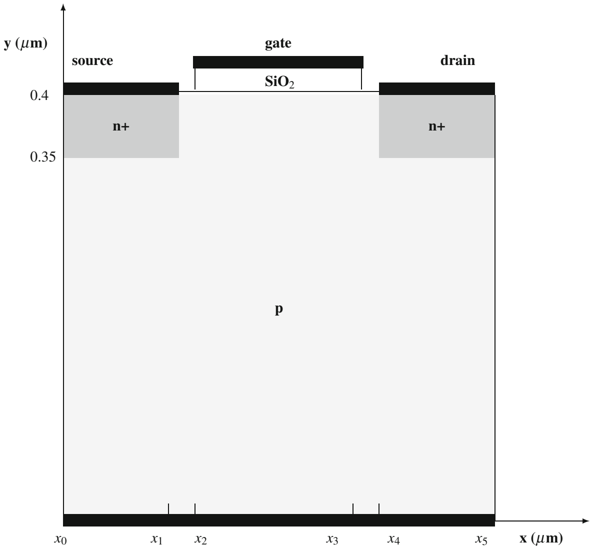
The obtained linear system can be solved efficiently with the tridiagonal matrix factorization procedure.

3.4.2 Numerical Simulation of the Crystal Lattice Heating in MOSFETs

We apply the above numerical method for the simulation of the heating of the crystal lattice in a MOSFET described by the MEP model.

We have modeled the thermal conductivity with the fitting formula $K(T_L) = 1.5486\,(T_L/300\,\mathrm{K})^{-4/3}$ V A/cm K and have set $c_V = 703\,\mathrm{m}^2/\mathrm{s}^2\,\mathrm{K}$ (see [35]). The

Fig. 3.3 Schematic representation of a bidimensional MOSFET



mobility of holes has been considered as constant and equal to 500 cm^2/V s. More details about the values of the physical parameters can be found in [30].

The shape of the device is shown in Fig. 3.3. The length of the channel ($x_4 - x_1$ in the figure) is L_c, the length of source and drain ($x_1 - x_0$) is $L = L_c/2$. We will consider $L_c = 50$ nm and $L_c = 200$ nm. The source and drain depths are 0.1 μm. The gate oxide is 20 nm thick. The substrate thickness is 0.4 μm. An environment temperature $T_{env} = 300$ K has been considered. In most of our numerical experiments we will take $R_{th} = 10^{-8}$ K m^2/W as in [9].

The doping concentration is

$$N_D(x) - N_A(x) = \begin{cases} n_+ \text{ in the } n^+ \text{ regions} \\ -p_- = -10^{14}\text{cm}^{-3}\text{in the } p \text{ region} \end{cases} \tag{3.100}$$

with abrupt junctions. We will consider different values of n_+ in the simulations.

First a MOSFET with 200 nm channel length has been simulated. The stationary solution is shown in the Figs. 3.4–3.8. The distance between gate and source ($x_2 - x_1$) and between drain and gate ($x_4 - x_3$) is 25 nm. The thermal resistivity of the contact is set equal to $R_{th} = 10^{-8}$ K m^2/W. The donor concentration is $n_+ = 10^{17}$cm^{-3}. In Fig. 3.4 one can see a relatively small heating of the crystal, just a maximum of about 7° above the environment temperature. The maximum of the crystal temperature is attained near the drain contact where also the maximum of the electron energy is observed (see Fig. 3.5). It is worth remarking that there is almost no influence of the device self-heating on the current through the device as shown in Fig. 3.8, where the characteristic curves with the lattice temperature fixed at 300 K are compared with those obtained with varying T_L.

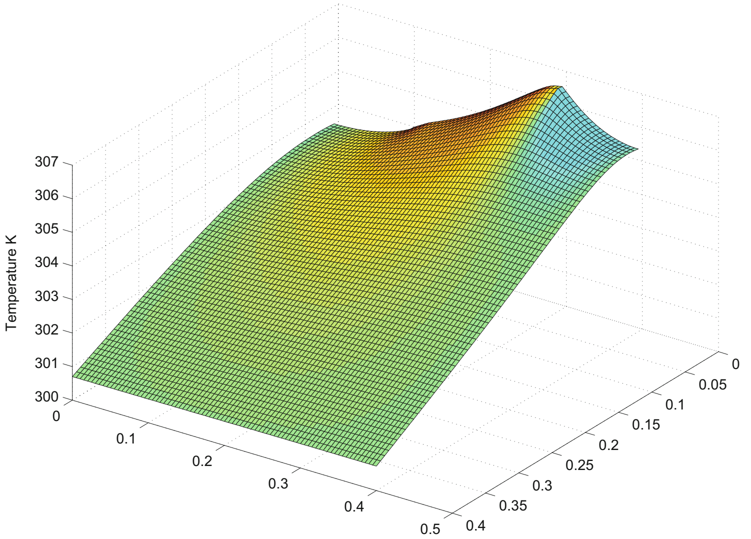


Fig. 3.4 Stationary solution of the lattice temperature in the MOSFET with channel of 200 nm by setting $R_{th} = 10^{-8}$ Km^2/W

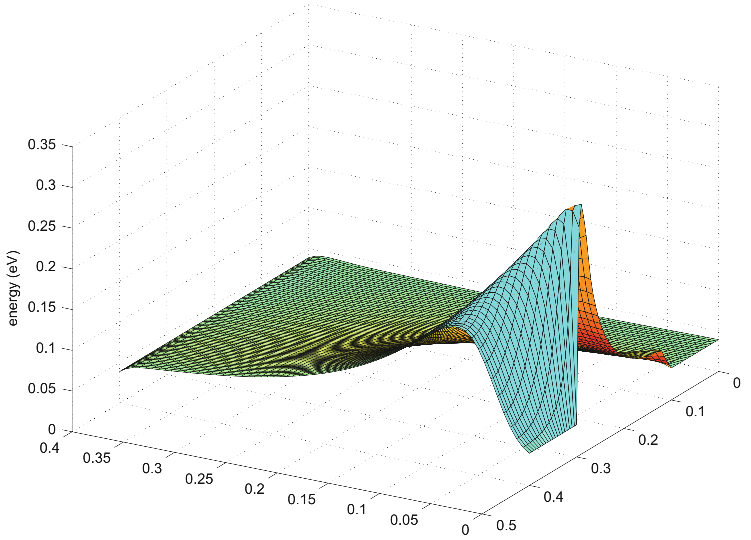


Fig. 3.5 Stationary solution of the electron energy in the MOSFET with channel of 200 nm by setting $R_{th} = 10^{-8}$ Km^2/W

As second example we have simulated a nanoscale MOSFET device with a channel of length 50 nm . The gate length is 45 nm and the gate voltage $V_{\rm DG} = 0.8$ V. The donor concentration is $n_+ = 10^{17}$cm^{-3}. In the Figs. 3.9 and 3.10 is plotted the stationary solution of the lattice temperature and the electron energy. In contrast to

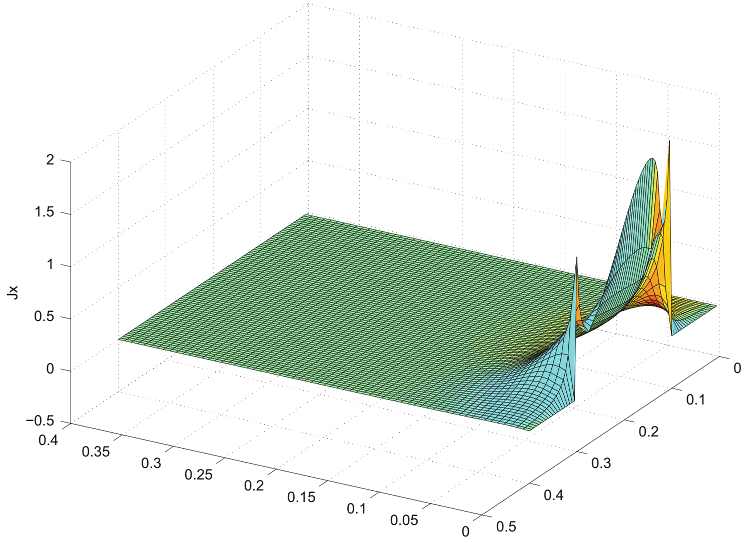


Fig. 3.6 Stationary solution of the x component of current in the MOSFET with channel of 200 nm by setting $R_{th} = 10^{-8}$ Km^2/W

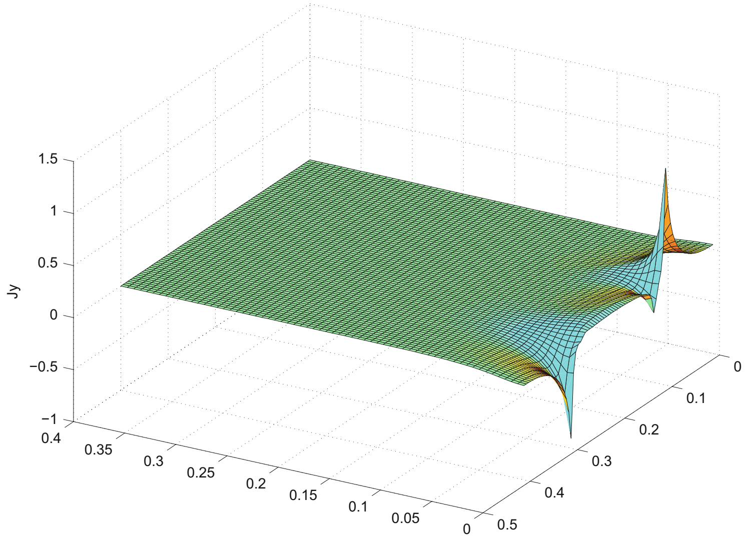


Fig. 3.7 Stationary solution of the y component of current in the MOSFET with channel of 200 nm by setting $R_{th} = 10^{-8}$ Km^2/W

the previous case, now the lattice temperature raises up to 380 K in the area near the gate. We argue that this temperature raise should depend, beside the strength of the electric field, on the density of the hot electrons and might be higher for higher

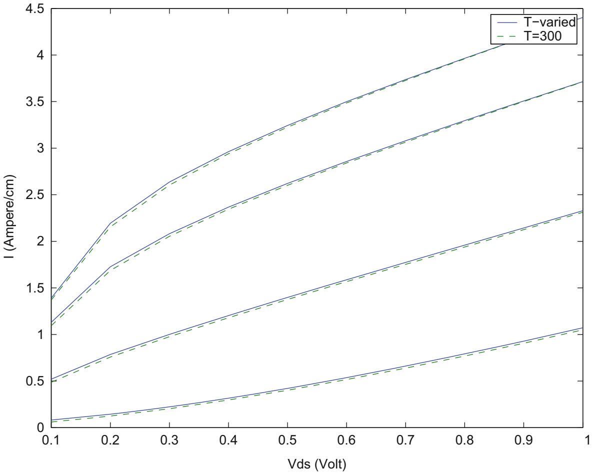


Fig. 3.8 Drain current for $V_{DG} = 0.4, 0.6, 0.8, 0.9\,\text{V}$. The current increases by increasing V_{DG}

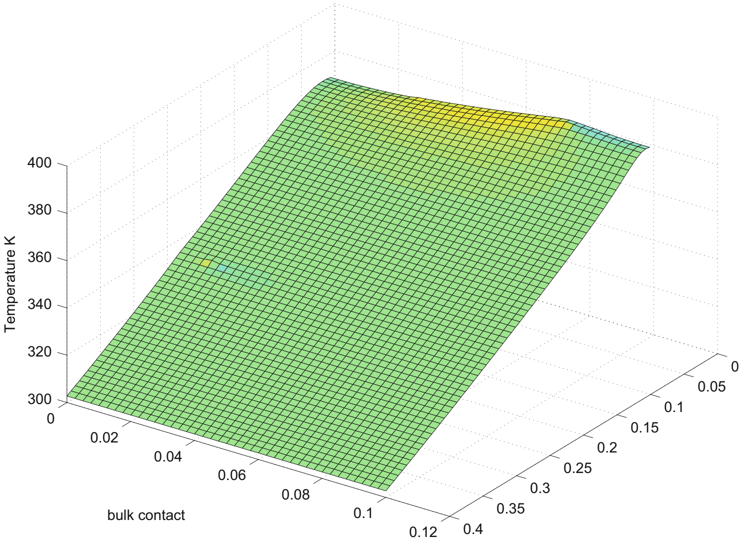


Fig. 3.9 Stationary solution of the lattice temperature in the MOSFET with channel of 50 nm by setting $R_{th} = 10^{-8}\,\text{Km}^2/\text{W}$

doping concentration. In order to investigate this assumption, a simulation with $n_+ = 5 \times 10^{17}\text{cm}^{-3}$ has been performed too. As expected one can see in Fig. 3.11 that the maximum lattice temperature attains about 550 K. In Fig. 3.12 the result of the lattice temperature for the even higher donor concentration $n_+ = 10^{18}\,\text{cm}^{-3}$ is reported. The maximum lattice temperature achieves about 700 K, confirming the dependence of it on the density of the electron current.

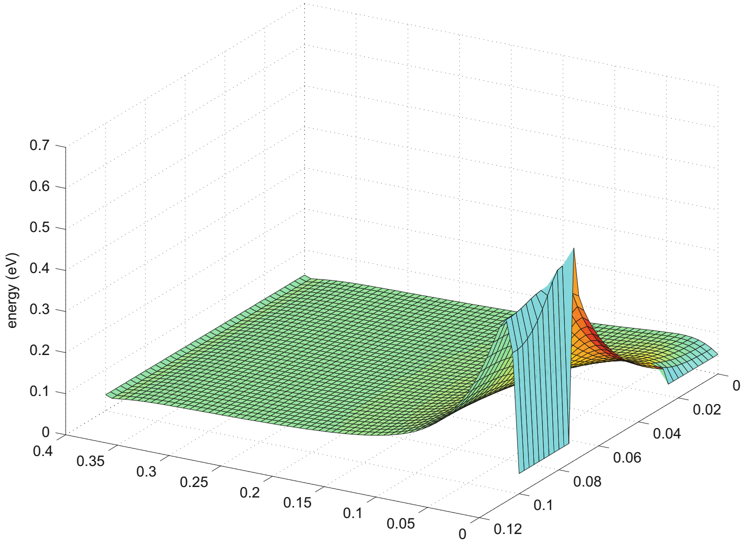


Fig. 3.10 Stationary solution of the electron energy in the MOSFET with channel of 50 nm by setting $R_{th} = 10^{-8}$ Km^2/W

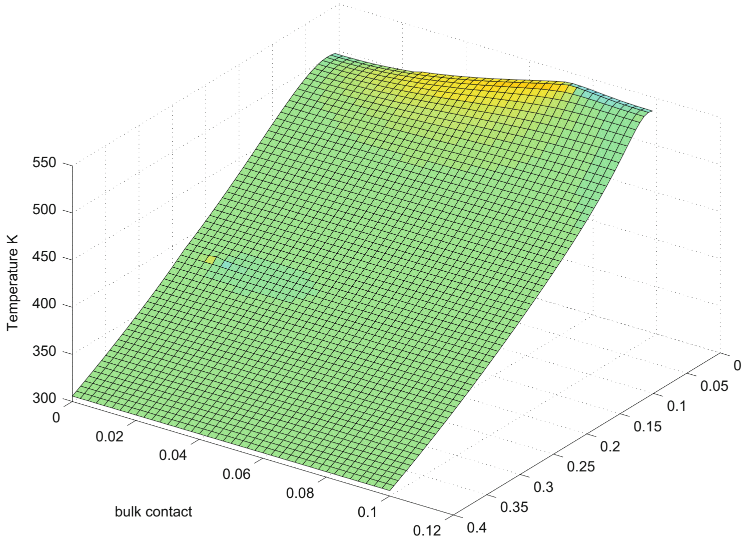


Fig. 3.11 Stationary solution of the lattice temperature in the MOSFET with channel of 50 nm by setting $n_+ = 5 \times 10^{17}$ cm^{-3} and $R_{th} = 10^{-8}$ Km^2/W

By shrinking the dimension of the device the thermal effects have also a non negligible influence on the current through the device. In Fig. 3.13 current Ãś voltage characteristics for the device with $n_+ = 10^{17}$ cm^{-3} are shown. With

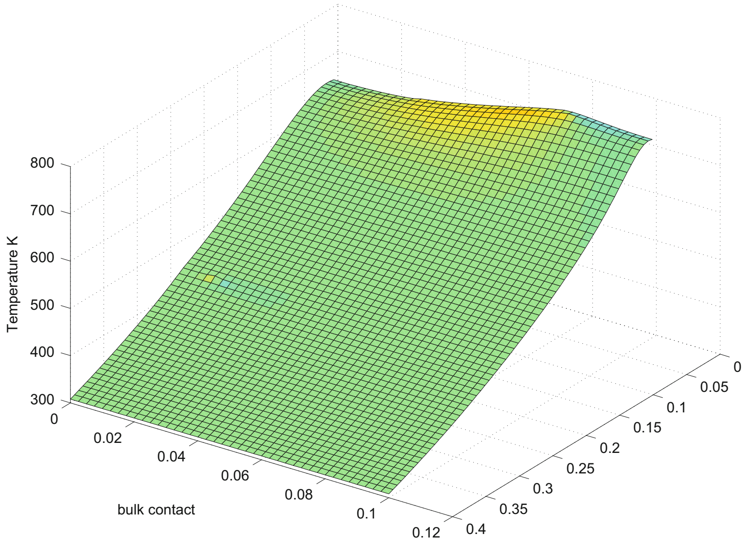


Fig. 3.12 Stationary solution of the lattice temperature in the MOSFET with channel of 50 nm by setting $n_+ = 10^{18}\,\text{cm}^{-3}$ and $R_{th} = 10^{-8}\,\text{Km}^2/\text{W}$

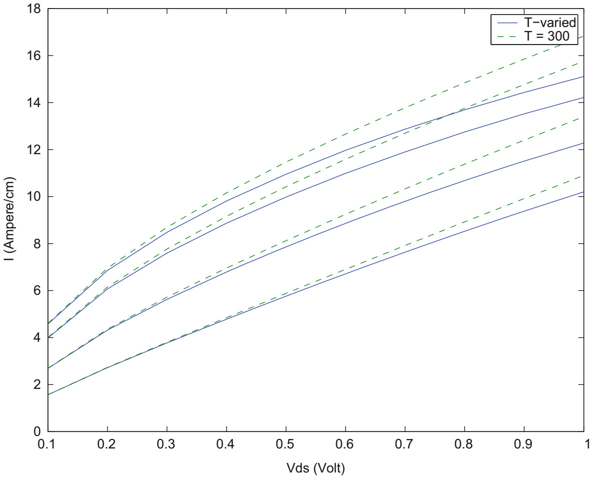


Fig. 3.13 Drain current with constant and varying lattice temperature in the MOSFET with channel of 50 nm by setting $n_+ = 10^{18}\text{cm}^{-3}$ and $R_{th} = 10^{-8}$ for $V_{DG} = 0.4, 0.6, 0.8, 0.9\,\text{V}$. The current increases by increasing V_{DG}

increasing electric field strength, we observe a rising deviation of the characteristic curves corresponding to a constant lattice temperature from those with varying T_L.

The lattice temperature in the device is also strongly influenced by the thermal resistivity of the contact R_{th}. This value depends on the manufacturing process. In Figs. 3.14 and 3.15 the lattice temperature is shown for $R_{th} = 10^{-10}\,\text{Km}^2/\text{W}$ and $R_{th} = 10^{-9}\,\text{Km}^2/\text{W}$ with $n_+ = 10^{17}\,\text{cm}^{-3}$.

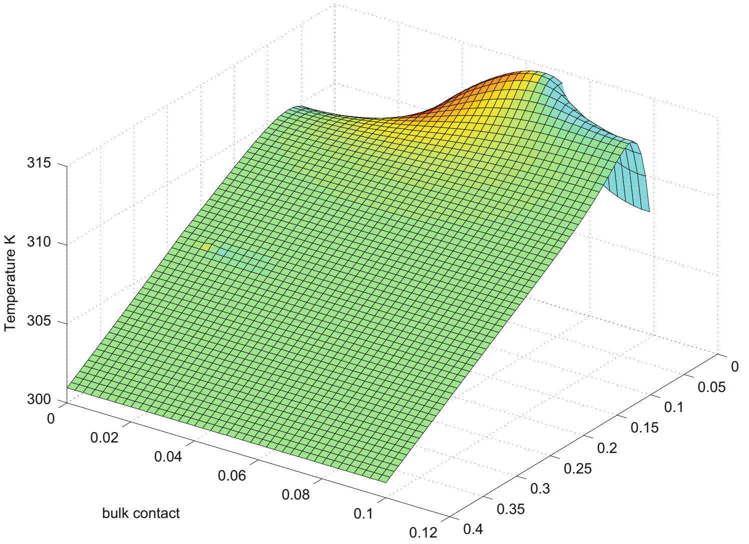


Fig. 3.14 Stationary solution of the lattice temperature in the MOSFET with channel of 50 nm by setting $R_{th} = 10^{-10}\,\mathrm{Km^2/W}$ and $n_+ = 10^{17}\,\mathrm{cm^{-3}}$

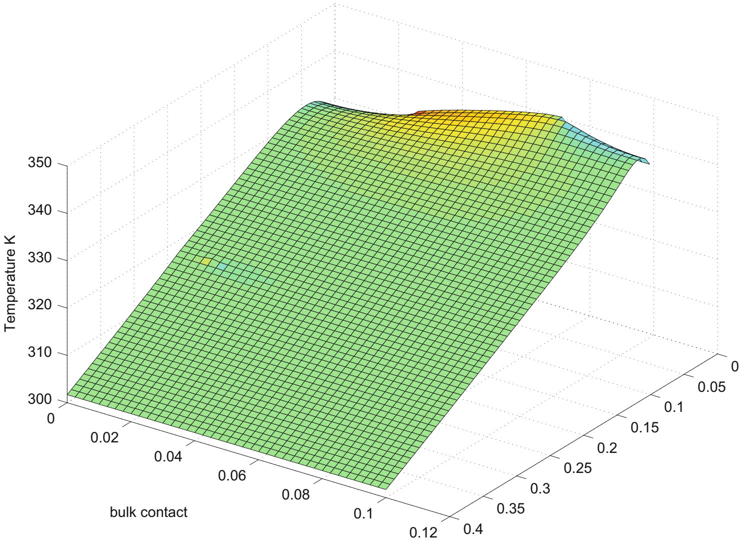


Fig. 3.15 Stationary solution of the lattice temperature in the MOSFET with channel of 50 nm by setting $R_{th} = 10^{-9}\,\mathrm{Km^2/W}$ and $n_+ = 10^{17}\,\mathrm{cm^{-3}}$

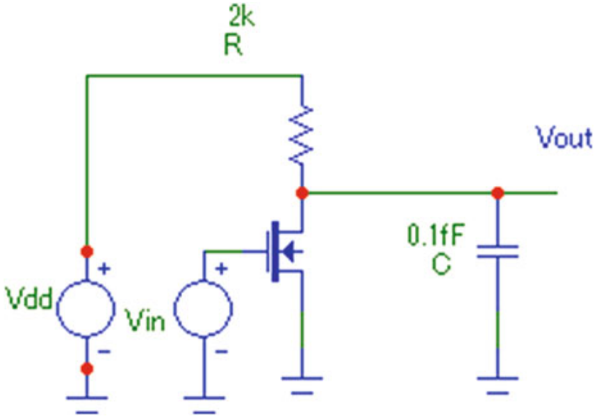


Fig. 3.16 Simulated inverter circuit

3.4.3 Coupled Circuit-Device Simulation

At last a case of coupling between a Mosfet and a circuit is present. We simulate the heating of a transistor in the electrical circuit representing an inverter. The inverter circuit is plot in Fig. 3.16. Input voltage on the gate contact is (in Volt) $V_{in} = 0.3\cos(\omega t) + 0.5$, with frequency $\omega = 2\pi\ 10^9$ rad/s and power voltage $V_{dd} = 1V$. The width of the transistor (length in the orthogonal direction with respect to the considered 2D cross section) is set equal to 200 nm. Modified nodal analysis gives us for the output voltage V_{out}:

$$C\frac{dV_{out}}{dt} + \frac{V_{out} - V_{dd}}{R} + j(V_{in}, V_{out}, t) = 0, \tag{3.101}$$

where current through the transistor $j(V_{in}, V_{out}, t)$ is computed by the energy-transport model. We refer for instance to [9] for details of device-circuits coupled modeling algorithm.

The output voltage simulated with and without transistor self heating and maximum temperature in the transistor are plot in the Fig. 3.17. One can see that lattice temperature in the transistor does not achieve 400 K as we have observed in the single transistor simulation. It can be explained with smaller average voltage at the gate and consequently smaller average electrical field. However there is still a shift in the minimum values of the output voltage and a clear indication of the crystal heating.

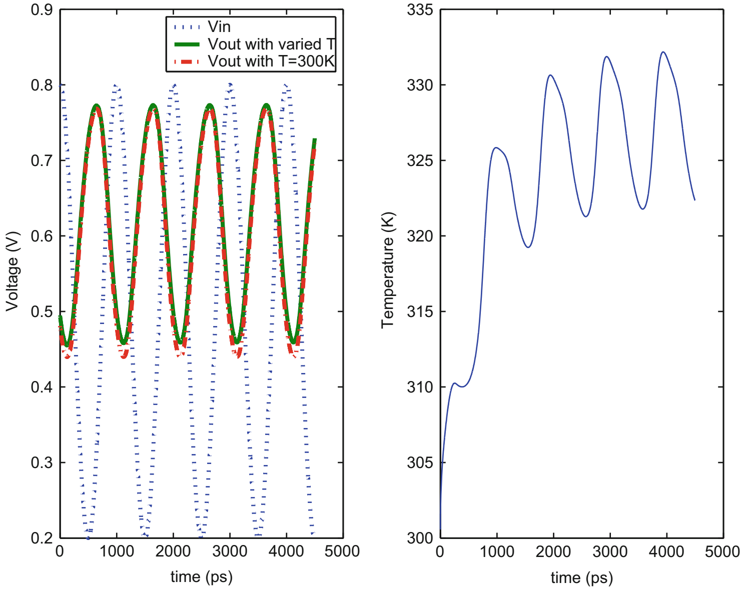


Fig. 3.17 On the *left* input and output voltages versus time. On the *right* maximum value of the lattice temperature in the MOSFET versus time

References

1. Arnold, M., Günther, M.: Preconditioned dynamic iteration for coupled differential-algebraic systems. BIT **41**(1), 1–25 (2001)
2. Arnold, M., Heckmann, A.: From multibody dynamics to multidisciplinary applications. In: García Orden, J., Goicolea, J., Cuadrado, J. (eds.) Multibody Dynamics. Computational Methods and Applications, pp. 273–294. Springer, Dordrecht (2007)
3. Bartel, A.: Partial Differential-Algebraic Models in Chip Design – Thermal and Semiconductor Problems. Forschrittsberichte. VDI-Verlag, Düsseldorf (2004)
4. Bartel, A., Brunk, M., Günther, M., Schöps, S.: Dynamic iteration for coupled problems of electric circuits and distributed devices. SIAM J. Sci. Comput. **35**(2), B315–B335 (2012)
5. Bartel, A., Günther, M.: Multirate co-simulation of first order thermal models in electric circuit design. In: Schilders, W., ter Maten, E., Houben, S. (eds.) Scientific Computing in Electrical Engineering SCEE 2002, Mathematics in Industry, pp. 23–28. Springer, Berlin (2004)
6. Ben Abdallah, N., Degond, P.: On a hierarchy of macroscopic models for semiconductors. J. Math. Phys. **37**, 3306–3333 (1996)
7. Ben Abdallah, N., Degond, P., Genieys, S.: An energy-transport model for semiconductors derived from the boltzmann equation. J. Stat. Phys. **84**, 205–231 (1996)
8. Brenan, K.E., Campbell, S.L.V., Petzold, L.R.: Numerical solution of initial-value problems in differential-algebraic equations. SIAM, Philadelphia (1995)
9. Brunk, M., Jüngel, A.: Numerical coupling of electric circuit equations and energy-transport models for semiconducotrs. SIAM J. Sci. Comput. **30**, 873–894 (2008)
10. Burrage, K.: Parallel and Sequential Methods for Ordinary Differential Equations. Clarendon, Oxford (1995)

11. Chen, D., Kan, E., Ravaioli, U., Shu, C.W., Dutton, R.: An improved energy-transport model including nonparabolicity and non-maxwellian distribution effects. IEEE Electron Device Lett. **13**, 26–28 (1992)
12. Clemens, M.: Large systems of equations in a discrete electromagnetism: formulations and numerical algorithms. IEE Proceedings - Science, Measurement and Technology **152**(2), 50–72 (2005). doi:10.1049/ip-smt:20050849
13. Clemens, M., Schuhmann, R., van Rienen, U., Weiland, T.: Modern Krylov subspace methods in electromagnetic field computation using the finite integration theory. Appl. Comput. Electromagn. Soc. J. **11**(1), 70–84 (1996)
14. Culpo, M.: Numerical Algorithms for System Level Electro-Thermal Simulation. Ph.D. thesis, Bergische Universität Wuppertal (2009)
15. Degond, P., Jüngel, A., Pietra, P.: Numerical discretization of energy-transport models for semiconductors with nonparabolic band structure. SIAM J. Sci. Comput. **22**, 986–1007 (2000)
16. Deuflhard, P., Hairer, E., Zugck, J.: One-step and extrapolation methods for differential-algebraic systems. Numer. Math. **51**, 501–516 (1987)
17. Estévez Schwarz, D., Tischendorf, C.: Structural analysis of electric circuits and consequences for mna. Int. J. Circuit Theory Appl. **28**(2), 131–162 (2000)
18. Gear, C., Wells, R.: Multirate linear multistep methods. BIT **24**, 484–502 (1984)
19. Günther, M.: Preconditioned splitting in dynamic iteration schemes for coupled dae systems in rc network design. In: Buikis, A., Ciegis, R., Fitt, A. (eds.) Progress in Industrial Mathematics at ECMI 2002, Mathematics in Industry, pp. 173–177. Springer, Berlin (2004)
20. Günther, M., Rentrop, P.: Multirate row methods and latency of electric circuits. Appl. Numer. Math. **13**, 83–102 (1993)
21. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer Series in Computational Mathematics. Springer, Berlin (1996)
22. Jackiewicz, Z., Kwapisz, M.: Convergence of waveform relaxation methods for differential-algebraic systems. SIAM J. Numer. Anal. **33**, 2303–2317 (1996)
23. Kübler, R., Schielen, W.: Two methods for simulator coupling. Math. Comput. Model. Dyn. Syst. **6**, 93–113 (2000)
24. Lelarasmee, E., Ruehli, A., Sangiovanni-Vincentelli, A.: The waveform relaxation method for time domain analysis of large scale integrated circuits. IEEE Trans. on CAD of IC and Syst. **1**, 131–145 (1982)
25. Lyumkis, E., Polsky, B., Shir, A., Visocky, P.: Transient semiconductor device simulation including energy balance equation. Compel **11**, 311–325 (1992)
26. Marchuk, G.: Splitting and alternating direction method. In: Ciarlet, P., Lions, J. (eds.) Handbook of Numerical Analysis. Vol. 1: Finite Difference Methods (Part 1) and Solution of Equations in $\mathbb{R}^n$ (Part 1), pp. 197–462. North-Holland, Amsterdam (1990)
27. Marrocco, A., Anile, A., Romano, V., Sellier, J.M.: 2d numerical simulation of the mep energy-transport model with a mixed finite elements scheme. J. Comput. Electron. **4**, 231–259 (2005)
28. Nicolet, A., Delincé, F.: Implicit Runge-Kutta methods for transient magnetic field computation. IEEE Trans. Magn. **32**(3), 1405–1408 (1996)
29. Romano, V.: 2d numerical simulation of the mep energy-transport model with a finite difference scheme. J. Comput. Phys. **221**, 439–468 (2007)
30. Romano, V., Rusakov, A.: 2d numerical simulations of an electron-phonon hydrodynamical model based on the maximum entropy principle. Comput. Meth. Appl. Mech. Eng **199**(41–44), 2741–2751 (2010)
31. Romano, V., Rusakov, A.: Numerical simulation of coupled electron devices and circuits by the mep hydrodynamical model for semiconductors with crystal heating. Il Nuovo Cimento C (2010). doi:10.1393/ncc/i2010-10573-5
32. Romano, V., Scordia, C.: Simulations of an electron-phonon hydrodynamical model based on the maximum entropy principle. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008, Mathematics in Industry. Springer, Berlin/Heidelberg (2010)

33. Romano, V., Zwierz, M.: Electron-phonon hydrodynamical model for semiconductors. 2741–2751 (2008) doi:10.1016/j.cma.2010.06.005
34. Schöps, S., Bartel, A., De Gersem, H., Günther, M.: DAE-index and convergence analysis of lumped electric circuits refined by 3-d magnetoquasistatic conductor models. Preprint 08/06, Bergische Universität Wuppertal, Wuppertal (2008)
35. Selberherr, S.: Analysis and simulation of semiconductor devices. Springer, Wien/New York (1984)
36. Stratton, R.: Diffusion of hot and cold electrons in semiconductor barriers. Phys. Rev. **126**, 2002–2014 (1962)
37. Tsukerman, I.: Finite element differential-algebraic systems for eddy current problems. Numer. Algorithms **31**(1), 319–335 (2002)

Part III
Model Order Reduction

MOR techniques are very well established in the electronics design community since nearly two decades. The main drivers have been the coupling of circuit simulation with electromagnetic problems by using the PEEC concept, and capturing parasitic effects due to interconnect on ICs. The subsystems generated this way are linear but very large. Furthermore they exhibit properties like passivity, controllability and observability, which must be maintained in the reduced order model. Based on sound mathematics, remarkable progress has been achieved in the development of methods and codes.

Chapter 4 gives an overview of current methods, concepts and properties. Chapter 5 is devoted to Parameterized Model Order Reduction: aiming at design optimisation including yield improvement in the COMSON project, it is natural to ask for MOR techniques which propagate relevant design or technological parameters from the original system to the reduced order model. Chapter 6 deals with more advanced topics: nonlinear networks and multi-terminal circuits.

Chapter 4
Model Order Reduction: Methods, Concepts and Properties

Athanasios C. Antoulas, Roxana Ionutiu, Nelson Martins, E. Jan W. ter Maten, Kasra Mohaghegh, Roland Pulch, Joost Rommes, Maryam Saadvandi, and Michael Striebel

Abstract This chapter offers an introduction to Model Order Reduction (MOR). It gives an overview on the methods that are mostly used. It also describes the main

A.C. Antoulas (✉)
Department of Electrical and Computer Engineering, William Marsh Rice University, MS-380, PO Box 1892, Houston, TX 77251-1892, USA

School of Engineering and Science, Jacobs University Bremen, 28725 Bremen, Germany
e-mail: A.C.Antoulas@jacobs-university.de; ACA@rice.edu

R. Ionutiu
ATTE, ABB Switzerland Ltd, Austraße, 5300 Turgi, Switzerland
email: Roxana.Ionutiu@ch.abb.com

N. Martins
CEPEL, 21941-911, Rio de Janeiro, RJ, Brazil
e-mail: Nelson@cepel.br

E.J.W. ter Maten
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Gaußstraße 20, D-42119 Wuppertal, Germany

Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O.Box 513, 5600 Eindhoven, The Netherlands
e-mail: Jan.ter.Maten@math.uni-wuppertal.de; E.J.W.ter.Maten@tue.nl

K. Mohaghegh
Multiscale in Mechanical and Biological Engineering (M2BE), Aragón Institute of Engineering Research (I3A), University of Zaragoza, María de Luna, 3, E-50018 Zaragoza, Spain
Kasra@unizar.es

R. Pulch
Institut für Mathematik und Informatik, Ernst Moritz Arndt Universität Greifswald, Walther-Rathenau-Straße 47, D-17487 Greifswald, Germany
e-mail: PulchR@uni-greifswald.de

J. Rommes
Mentor Graphics, DSM/AMS, Le Viseo – Bâtiment B, 110 rue Blaise Pacal, Inovalee, 38330 Montbonnot, France
e-mail: Joost_Rommes@mentor.com

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_4

concepts behind the methods and the properties that are aimed to be preserved. The sections are in a prefered order for reading, but can be read independentlty. Section 4.1, written by Michael Striebel, E. Jan W. ter Maten, Kasra Mohaghegh and Roland Pulch, overviews the basic material for MOR and its use in circuit simulation. Issues like Stability, Passivity, Structure preservation, Realizability are discussed. Projection based MOR methods include Krylov-space methods (like PRIMA and SPRIM) and POD-methods. Truncation based MOR includes Balanced Truncation, Poor Man's TBR and Modal Truncation.

Section 4.2, written by Joost Rommes and Nelson Martins, focuses on Modal Truncation. Here eigenvalues are the starting point. The eigenvalue problems related to large-scale dynamical systems are usually too large to be solved completely. The algorithms described in this section are efficient and effective methods for the computation of a few specific dominant eigenvalues of these large-scale systems. It is shown how these algorithms can be used for computing reduced-order models with modal approximation and Krylov-based methods.

Section 4.3, written by Maryam Saadvandi and Joost Rommes, concerns passivity preserving model order reduction using the spectral zero method. It detailedly discusses two algorithms, one by Antoulas and one by Sorenson. These two approaches are based on a projection method by selecting spectral zeros of the original transfer function to produce a reduced transfer function that has the specified roots as its spectral zeros. The reduced model preserves passivity.

Section 4.4, written by Roxana Ionutiu, Joost Rommes and Athanasios C. Antoulas, refines the spectral zero MOR method to dominant spectral zeros. The new model reduction method for circuit simulation preserves passivity by interpolating dominant spectral zeros. These are computed as poles of an associated Hamiltonian system, using an iterative solver: the subspace accelerated dominant pole algorithm (SADPA). Based on a dominance criterion, SADPA finds relevant spectral zeros and the associated invariant subspaces, which are used to construct the passivity preserving projection. RLC netlist equivalents for the reduced models are provided.

Section 4.5, written by Roxana Ionutiu and Joost Rommes, deals with synthesis of a reduced model: reformulate it as a netlist for a circuit. A framework for model reduction and synthesis is presented, which greatly enlarges the options for the re-use of reduced order models in circuit simulation by simulators of choice. Especially when model reduction exploits structure preservation, we show that using the model as a current-driven element is possible, and allows for synthesis without controlled

M. Saadvandi
Numerical Approximation and Linear Algebra Group, Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium Maryam.Saadvandi@cs.kuleuven.be

M. Striebel
ZF Lenksysteme GmbH, Richard-Bullinger-Straße 77, D-73527 Schwäbisch Gmünd, Germany
e-mail: Michael.Striebel@zf-lenksysteme.com

sources. Two synthesis techniques are considered: (1) by means of realizing the reduced transfer function into a netlist and (2) by unstamping the reduced system matrices into a circuit representation. The presented framework serves as a basis for reduction of large parasitic R/RC/RCL networks.

Co-operations Between the Various Co-authors

The subactivity on Model Order Reduction (MOR) of the COMSON project[1] was greatly influenced by interaction with additional research on MOR, first at Philips Research Laboratories and (from october 2006 on) at NXP Semiconductors (both in Eindhoven). There was direct project work with the TU Eindhoven, with the Bergische Universität Wuppertal and with the Royal Institute of Technology (KTH) in Stockholm:

- R. IONUTIU: *Model order reduction for multi-terminal Systems – with applications to circuit simulation*. Ph.D.-Thesis, TU Eindhoven, 2011, http://alexandria.tue.nl/extra2/716352.pdf.
- M. SAADVANDI: *Passivity preserving model reduction and selection of spectral zeros*. MSc. Thesis, Royal Institute of Technology (KTH), Stockholm. Also published as Technical Note NXP-TN-2008/00276, Unclassified Report, NXP Semiconductors, Eindhoven, 2008. [In September 2012, Maryam Saadvandi did complete a Ph.D.-Thesis at KU Leuven, Belgium, on *Nonlinear and parametric model order reduction for second order dynamical systems by the dominant pole algorithm*.]
- M.V. UGRYUMOVA: *Applications of Model Order Reduction for IC Modeling*. Ph.D.-Thesis, TU Eindhoven, 2011, http://alexandria.tue.nl/extra2/711015.pdf.
- A. VERHOEVEN: *Redundancy reduction of IC models by multirate time-integration and model order reduction*. Ph.D.-Thesis, TU Eindhoven, 2008, http://alexandria.tue.nl/extra2/200712281.pdf.
- T. VOSS: *Model reduction for nonlinear differential algebraic equations*, MSc. Thesis, University of Wuppertal, 2005. Unclassified Report PR-TN-2005/00919, Philips Research Laboratories, September 2005.
  [Afterwards, Thomas Voß did complete a Ph.D.-Thesis at the Rijksuniversiteit Groningen, the Netherlands, on *Port-Hamiltonian modeling and control of piezoelectric beams and plates: application to inflatable space structures*, 2010, http://catalogus.rug.nl/DB=1/SET=1/TTL=4/REL?PPN=326-918639.]

[1] *Coupled Multiscale Simulation and Optimization in Nano-electronics, COMSON* – EU-FP6 MCA-RTN Research and Training Network Project, 2006–2010, http://www.comson.eu/.

Here Roxana Ionutiu was partially funded by the COMSON project. Apart from TU Eindhoven she also worked with Thanos Antoulas at the Jacobs University in Bremen. Roxana Ionutiu appears several times as co-author in this chapter and in the following ones. Also Maryam Saadvandi appears as co-author of a section. Work by the others is found in the reference lists at each section.

Parallel to the COMSON Project research on MOR was done within the O-MOORE-NICE! project.[2] The Marie Curie Fellows, Luciano De Tommasi (University of Antwerp), Davit Harutyunyan (TU Eindhoven), Joost Rommes (NXP Semiconductors), and Michael Striebel (Chemnitz University of Technology), interacted actively with the COMSON PhD-students. They contribute to several sections as co-authors, together with researchers from the staff from NXP Semiconductors (Eindhoven), TU Eindhoven, Bergische Universität Wuppertal and the Politehnica Univ. of Bucharest.

The Politehnica Univ. of Bucharest greatly acknowledges co-operation with Jorge Fernandez Villena and Luis Miguel Silveira of INESC-ID in Lisbon. They appear as co-author in the next chapter. Jorge Fernandez Villena was partially funded by the COMSON project. Work in Bucharest and in Lisbon also did benefit from financial support during earlier years from the following complementary projects: FP6/Chameleon, FP5/Codestar, CEEX/nEDA, UEFISCSU/IDEI 609/16.01.2009 and POSDRU/89/1.5/S/62557.

The fourth co-author acknowledges the ENIAC JU Project /2010/SP2(Wireless communication)/270683-2 Artemos, *Agile Rf Transceivers and front-Ends for future smart Multi-standard cOmmunications applicationS*, http://.artemos.eu.

The COMSON project did directly lead to four Ph.D.-Theses on MOR-related topics:

- Z. ILIEVSKI: *Model order reduction and sensitivity analysis*. Ph.D.-Thesis, TU Eindhoven, 2010, http://alexandria.tue.nl/extra2/201010770.pdf.
- S. KULA: *Reduced order models of interconnects in high frequency integrated circuits*. Ph.D.-Thesis, Politehnica Univ. of Bucharest, 2009.
- K. MOHAGHEGH: *Linear and nonlinear model order reduction for numerical simulation of electric circuits*. Ph.D.-Thesis, Bergische Universität Wuppertal, Germany. Available at Logos Verlag, Berlin, Germany, 2010.
- A. ŞTEFĂNESCU: *Parametric models for interconnections from analogue high frequency integrated circuits*. Ph.D.-Thesis, Politehnica Univ. of Bucharest, 2009.

[2] *Operational MOdel Order REduction for Nanoscale IC Electronics (O-MOORE-NICE!)* – EU-FP6 MCA-ToK Transfer of Knowledge Project, 2007–2010, http://www.tu-chemnitz.de/mathematik/industrie_technik/projekte/omoorenice/index.php?lang=en

4.1 Circuit Simulation and Model Order Reduction

Speaking of “circuit models”, we refer to models of electrical circuits derived by a network approach.[3] In circuit simulation the charge-oriented modified nodal analysis (MNA) is a prominent representative of network approaches used to automatically create mathematical models for a physical electrical circuit. In the following we give a short introduction to circuit modeling with MNA. For a detailed discussion we refer to [22].

In charge-oriented MNA, voltages, currents, electrical charges and magnetic fluxes are the quantities that describe the activity of a circuit. The electrical circuit to be modelled is considered to be an aggregation of basic network elements: the ohmic resistor, capacitor, inductor, voltage source and current source. Real physical circuit elements, especially semiconductor devices, are replaced by idealised network elements or so-called “companion models”. The basic network elements correlate the network quantities. Each basic element is associated to a *characteristic equations*:

- Resistor: $I = r(U)$ (linear case: $I = \frac{1}{R} \cdot U$)
- Capacitor: $I = \dot{q}$ with $q = q_C(U)$ (linear case: $I = C \cdot \dot{U}$)
- Inductor: $U = \dot{\phi}$ with $\phi = \phi_L(I)$ (linear case: $U = L \cdot \dot{I}$)
- Voltage source: $U = v(t)$ (controlled source: $U = v(U_{\text{ctrl}}, I_{\text{ctrl}}, t)$)
- Current source: $I = \imath(t)$ (controlled source: $I = \imath(U_{\text{ctrl}}, I_{\text{ctrl}}, t)$)

where U is the voltage drop across the element’s terminal, I is the current flowing through the element, q is the electric charge stored in a capacitor and ϕ is the magnetic flux of an inductor. The dot $\dot{}$ on top of a quantity indicates the usual time derivative d/dt on that quantity.

All wires, connecting the circuit elements are considered to be electrically ideal, i.e., no wire possesses any resistance, capacitance or inductance. Thereby, also the volume expansion of the circuit becomes irrelevant, the electrical system is considered being a lumped circuit.The circuit’s layout, defined by the interconnects between elements, is thus reduced to its conceptional structure, which is called *network topology*.

The network’s topology consists of *branches* and *nodes*. Each network element is regarded as a branch of the circuit and its terminals are the nodes by which it is connected to other elements. Assigning a direction to each branch – the direction of the current traversing the corresponding element – and a serial number to each node, we end up with a *directed graph* representing the network. As any directed graph, the network can be described by an *incidence matrix* A. This matrix has as

[3] Section 4.1 has been written by: Michael Striebel, E. Jan W. ter Maten, Kasra Mohaghegh and Roland Pulch. For additional details we refer to the Ph.D.-Thesis [33] of the third author.

many columns as there are branches, i.e., elements and as many rows as there are nodes in the circuit. Each column of the matrix has one entry $+1$ and one entry -1, displaying the start and end point of the branch. As all other entries are 0, the matrix A is sparse.

Usually, one circuit node is tagged as *ground node*. As a consequence, each branch voltage U between two nodes l an m can be expressed by the two *node voltages* e_l and e_m, which are the voltage differences between each node and the ground node. From this agreement, the node voltage of the ground node is constantly 0 and therefore the information stored in the corresponding row of the incidence matrix becomes redundant and this very row can be removed. Hence, frequently by the term *incidence matrix*, one refers to the reduced matrix $\mathbf{A}$, given by removing the row corresponding to the ground node.

As each branch of the network represents one of the five basic network element resistor (R), capacitor (C), inductor (L), voltage and current source (V and I, respectively), the indicence matrix can be described as an assembly of element related incidence matrices:

$$\mathbf{A} = (\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_V, \mathbf{A}_I)\,,$$

with $A_\Omega \in \{0, +1, -1\}^{n_e \times n_\Omega}$ for $\Omega \in \{C, R, L, V, I\}$. Here, n_e is the number of nodes (without the ground node) and $n_C, \dots, n_I$ are the cardinalities of the sets of the different basic elements' branches.

The *Kirchhoff's laws*, which relate the branch voltages in a loop and the currents accumulating in a node, namely *Kirchhoff's voltage law* and *Kirchhoff's current law*, respectively, are the final component for setting up the *MNA network equations*:

$$\mathbf{A}_C \frac{d}{dt}\mathbf{q} + \mathbf{A}_R \mathbf{r}(\mathbf{A}_R^T \mathbf{e}) + \mathbf{A}_L \boldsymbol{\imath}_L + \mathbf{A}_V \boldsymbol{\imath}_V + \mathbf{A}_I \imath(t) = \mathbf{0}, \tag{4.1a}$$

$$\frac{d}{dt}\boldsymbol{\phi} - \mathbf{A}_L^T \mathbf{e} = \mathbf{0}, \tag{4.1b}$$

$$\mathbf{v}(t) - \mathbf{A}_V^T \mathbf{e} = \mathbf{0}, \tag{4.1c}$$

$$\mathbf{q} - \mathbf{q}_C(\mathbf{A}_C^T \mathbf{e}) = \mathbf{0}, \tag{4.1d}$$

$$\boldsymbol{\phi} - \boldsymbol{\phi}_L(\boldsymbol{\imath}_L) = \mathbf{0}. \tag{4.1e}$$

It is worthwile to highlight the subequations (4.1a) and (4.1c). The former is the personification of Kirchhoff's current law, stating that for each network node the sum of branch currents meeting is identically zero. The latter reflects the functionality of voltage sources: dictating branch voltages.

The unknowns $\mathbf{q}, \boldsymbol{\phi}, \mathbf{e}, \boldsymbol{\imath}_L, \boldsymbol{\imath}_V$, i.e., the charges, fluxes, node voltages and currents traversing inductors and voltage sources, respectively – each of them functions of time t – are combined to the *state vector* $\mathbf{x}(t) \in \mathbb{R}^n$ of unknowns, of dimension $n = n_C + n_L + n_e + n_L + n_V$. Then, the network equations (4.1) can be stated in a compact form:

$$\frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{4.2}$$

where $\mathbf{q}, \mathbf{j} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ describe the contribution of reactive and nonreactive elements, respectively.[4] The excitations defined by the voltage- and current-sources are combined to the vector $\mathbf{u}(t) \in \mathbb{R}^m$ with $m = n_V + n_I$. The excitations are assigned to the corresponding nodes and branches by the matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$.

If the circuit under considerations contains only elements with a linear characteristic equation, the network equations can be written as[5]

$$\mathbf{E}\dot{\mathbf{x}}(t) + \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{4.3a}$$

with

$$\mathbf{E} = \begin{pmatrix} \mathbf{A}_C \mathscr{C} \mathbf{A}_C^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathscr{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_R \mathscr{G} \mathbf{A}_R^T & \mathbf{A}_L & \mathbf{A}_V \\ -\mathbf{A}_L^T & \mathbf{0} & \mathbf{0} \\ -\mathbf{A}_V^T & \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \mathbf{A}_I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_V} \end{pmatrix}, \tag{4.3b}$$

where $\mathscr{C}, \mathscr{L}, \mathscr{G}$ are basically diagonal matrices containing the individual capacitors, inductances and conductances (inverse resistances) of the basic network elements. $\mathbf{I}_{n_V}$ is the identity matrix in $\mathbb{R}^{n_V \times n_V}$.

We arrive at this formulation by eliminating the charges and fluxes. Hence the unknown state vector here is $\mathbf{x} = (\mathbf{e}^T, \boldsymbol{\imath}_L^T, \boldsymbol{\imath}_V^T)^T$ and the excitation vector is $\mathbf{u} = (\boldsymbol{\imath}_I^T, \mathbf{v}^T)^T$.

It is straightforward to see that the structure of the matrices $\mathbf{E}, \mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times m}$ is determined by the element related incidence matrices $\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_V, \mathbf{A}_I$. As there is usually only a week linkage amongst the network node, i.e., nodes are connected directly to only a few other nodes, these incidence matrices are sparse and so are the system matrices in (4.3a) and the Jacobian matrices $d\mathbf{q}/d\mathbf{x}, d\mathbf{j}/d\mathbf{x} \in \mathbb{R}^{n \times n}$ of the element functions in (4.2), respectively.

[4] Note that the meaning $\mathbf{q}$ in (4.1) and (4.2) is different: in the prior it is an unknown, in the latter it is a mapping.

[5] Note that $\mathbf{A}$ in (4.3a) does not refer to the incidence matrix $\mathbf{A}$. Furthermore the composition of the unknown $\mathbf{x}$ in (4.2) and (4.3a) can be different. In the latter, taking into account the linear characteristics for capacitors and inductors, the time derivatives of the charges and fluxes can be expressed by the time derivative of the node volages $\mathbf{e}$ and the inductor current $\boldsymbol{\imath}_L$ directly. In this case the unknown state vector amounts to $\mathbf{x} = (\mathbf{e}^T, \boldsymbol{\imath}_L^T, \boldsymbol{\imath}_V^T)^T \in \mathbb{R}^n$ with $n = n_e + n_V + n_L$.

In general, real circuit designs contain a large number of transistors. In the course of setting up the network equations such semiconductor devices are replaced by companion models that consist of a larger number of the basic network elements. Here especially resistors with nonlinear characteristics emerge. Hence, the "mathematical image" of an integrated circuit is usually a nonlinear network equation of the form (4.2).

However, also linear network equations of the form (4.3a) are fundamental problems in the design process. As mentioned above, one disregards the volume extension of a circuit and considers wires as electrically ideal. At the end of the design process, however, there will be a physical integrated circuit. Even on the smallest dies there are kilometers of wiring. These wires do have an electric resistance. As the actual devices are getting small and smaller, capacitive effects introduced by neighbouring wires can not be neglected just as little as inductive effects arising from increasing clock rates.

In fact these issues are not neglected. At least at the end of the design process, when the layout of the chip has to be determined these effects are taken into account. In the *parasitic extraction* from the routing on the chip an artificial linear network is extracted which again is assumed to be a lumped and comprise of ideal wires. However, the resistances, capacitances and inductances that are present there describe the effects caused by the wiring on the actual circuit. A characteristic of these artificial networks is their large dimension: here n can easily be in the range of 10^6.

The impact of the effects on the behaviour of the actual circuit are accounted for by coupling the linear parasitic model to the underlying circuit design.

If the electrical circuit comprises reactive elements, i.e., capacitors and inductors, the network equation (4.2) or (4.3a), respectively, forms a dynamical problem for the unknown state vector $\mathbf{x}$. Usually, however, the system matrix $\mathbf{E}$, or the Jacobian $d\mathbf{q}/d\mathbf{x}$, respectively, does not have full rank.[6] Dynamical systems with this property, i.e., systems consisting of differential and algebraic equations are called *differential algebraic equations (DAE)*, or *descriptor systems*. DAEs differ in several senses from purely differential equations, causing problems in various aspects. A requirement for the solvability of the network equation is the regularity of the matrix pencil $\{\mathbf{E}, \mathbf{A}\}$. The matrix pencil is called regular, if the polynomial $\det(\lambda\mathbf{E} + \mathbf{A})$ does not vanish identically. Otherwise $\{\mathbf{E}, \mathbf{A}\}$ is called *singular matrix pencil*. Then a normal initial-value problem for the linear DAE (4.3a) has none or infinitely many solutions. The regularity of the matrix pencil can be checked by examining the element related incidence matrices [15].

[6] This is easy to see from inspecting the first subequation – the node-current relation – of the MNA equation (4.1): a network node for instance, that is not the starting or end point of a capacitor branch causes a row equal to zero in the incidence matrix $\mathbf{A}_C$ and therefore the node-current relation for that node is an algebraic equation only.

In the context of numerical time integration, needed to solve the network problem in time domain, worthwhile stressing that the initial value has to be chosen properly – $\mathbf{x}(0)$ has to satisfy the algebraic constraints – and that numerical perturbations can be amplified dramatically. Hence, numerical methods have to match the requirements posed by the differential-algebraic structure.

For a detailed analysis of DAEs we refer to the textbook [29]. A detailed discussion of solving DAEs can be found in the textbook [24].

4.1.1 Input-Output Systems in Circuit Simulation

We recall that the origin of the network equations in nonlinear or linear form is a real circuit design, ment to be simulated, i.e., tested with respect to its performance under different circumstances. Nowadays, complex integrated circuits are usually not designed from scratch by a single engineer. In fact, large electrical circuits are usually developed in a modular way. In radio frequency applications, for instance, analogue and digital subcircuits are connected to each other. In general several subunits of different functionality, e.g., one providing stable oscillations another one amplifying a signal, are developed separately and glued together. Hence, subunits possess a way of communication with other subunits, the environment they are embedded in.

To allow for a communication with an environment, the network model (4.2) (or (4.3a)) has to be augmented and transfered to a system that can receive and transmit information. Abstractly, the output of a system can be defined as a function of the state and the input:

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)) \in \mathbb{R}^p .$$

In circuit simulation, however, usually the output is a linear relation of the form:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t),$$

with the *output matrix* $\mathbf{C} \in \mathbb{R}^{p\times n}$ and the *feedthrough matrix* $\mathbf{D} \in \mathbb{R}^{p\times m}$.

The excitation, we mentioned above, i.e., the last term $\mathbf{B}\mathbf{u}(t)$ in the network model (4.2) (or (4.3a)) can be understood as information imposed on the system, in the form of branch currents and node voltages. Therefore we call $\mathbf{u}(t)$ the *input* and $\mathbf{B}$ the *input matrix* to the system.

Hence, an *input-output system* in electrical circuit simulation is given in the form

$$\mathbf{0} = \mathbf{E}\dot{\mathbf{x}}(t) + \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4.4a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{4.4b}$$

if only linear elements form the system. If also nonlinear elements are present, we arrive at systems of the form:

$$\mathbf{0} = \frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t), \tag{4.5a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t). \tag{4.5b}$$

The input to state mapping (4.4a) and (4.5a), respectively, is a relation defined by a dynamical system. Therefore, the representation of the input-output system (4.4) and (4.5), respectively, is said to be given in *state space formulation*. The dimension n of the state space is referred to as the *order* of the system.

Frequently the state space formulation in circuit design exhibits a special structure.

- Often there is no direct feedthrough of the input to the output, i.e.

$$\mathbf{D} = \mathbf{0} \in \mathbb{R}^{p \times m}\,. \tag{4.6a}$$

- We often observe

$$p \equiv m \quad \text{and} \quad \mathbf{C} = \mathbf{B}^T \in \mathbb{R}^{m \times n}\,. \tag{4.6b}$$

In full system simulation, individual subcircuit models are connected to each other. To allow for an information exchange, done in terms of currents and voltages, each subcircuit possesses a set of terminals – a subset of the unit's pins.

From a subcircuit's point of view incoming information is either a current being added to or a voltage drop being imposed to the terminal nodes. The former corresponds to adding a current source term to (4.1a), the latter corresponds to adding a voltage source to (4.1c). Information returned by the subsystem is the voltage at the terminal node in the former case or the current traversing that artificial voltage source in the latter case. Having a detailed look at the MNA network equations (4.1) and the composition of the state vector $\mathbf{x}(t)$, it is easy to understand that in this setup, assuming that there are no additional sources in the subcircuits, the output matrix is the transpose of the input matrix.

4.1.2 The Need for Model Order Reduction

Clearly, mathematical models for a physical circuit are extracted for a purpose. In short, the manufacturing process of an electrical circuit starts with an idea of what the physical system should do and ends with the physical product. In-between there is a, usually iterative, process of conceptual designing the circuit in the form of a circuit schematic, that comprises parameters defining the layout and nominal values of circuit elements and, choosing the parameters, testing the design, adapting the parameter, ..., etc.

Testing the design means to analyse its behaviour. There are several types of analysis we briefly want to mention in the following. For a more detailed discussion we refer to [22].

- *Static (DC) analysis* searches for the point to which the system settles in an equilibrium or rest condition. This is characterised by $d/dt\, \mathbf{x}(t) = 0$.
- *Transient analysis* computes the response $\mathbf{y}(t)$ to the time varying excitation $\mathbf{u}(t)$ as a function of time.
- *(Periodic) steady-state analysis*, also called *frequency response analysis*, determines the response of the system in the frequency domain to an oscillating, i.e., sinusoidal input signal.
- *Modal analysis* finds the system's natural vibrating frequency modes and their corresponding modal shapes;
- *Sensitivity analysis* determines the changes of the time-domain response and/or the frequency-domain response to variations in the design parameters.

Transient analysis is run in the time domain. Here the challenge is to numerically integrate a very high-dimensional DAE problem.

Both the frequency response and the modal analysis are run in the frequency domain. Hence, a network description in the frequency domain is needed. As this is basically defined only for linear systems[7] we concentrate on linear network problems of the form (4.4). The *Laplace transform* is the tool to get from the time to the frequency domain.

Recall that for a function $f : [0, \infty) \to \mathbb{C}$ with $f(0) = 0$, the Laplace transform $F : \mathbb{C} \to \mathbb{C}$ is defined by

$$F(s) := \mathscr{L}\{f\}(s) = \int_0^\infty f(t)e^{-st}\,dt.$$

For a vector-valued function $\mathbf{f} = (f_1, \ldots, f_q)^T$, the Laplace transform is defined component-wise: $\mathbf{F}(s) = (\mathscr{L}\{f_1\}(s), \ldots, \mathscr{L}\{f_q\}(s))^T$.

The physically meaningful values of the complex variable s are $s = i\omega$ where $\omega \geq 0$ is referred to as the *(angular) frequency*. Taking the Laplace transform of the time domain representation of the linear network problem (4.4) we obtain the following frequency domain representation:

$$\begin{aligned} \mathbf{0} &= s\mathbf{EX}(s) + \mathbf{AX}(s) + \mathbf{BU}(s), \\ \mathbf{Y}(s) &= \mathbf{CX}(s) + \mathbf{DU}(s), \end{aligned} \tag{4.7}$$

[7] Applying those types of analysis to nonlinear problems involves a linearisation about some point of interest $\mathbf{x}$ in the state space.

where $\mathbf{X}(s), \mathbf{U}(s), \mathbf{Y}(s)$ are the Laplace transforms of the states, the input and the output, respectively. Note that we assumed zero initial conditions, i.e., $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{u}(0) = \mathbf{0}$ and $\mathbf{y}(0) = \mathbf{0}$.

Eliminating the variable $\mathbf{X}(s)$ in the frequency domain representation (4.7) we see that the system's response to the input $\mathbf{U}(s)$ in the frequency domain is given by

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s)$$

with the matrix-valued *transfer function*

$$\mathbf{H}(s) = -\mathbf{C}\left(s\mathbf{E} + \mathbf{A}\right)^{-1}\mathbf{B} + \mathbf{D} \quad \in \mathbb{C}^{p \times m}. \tag{4.8}$$

The evaluation of the transfer function is the key to the frequency domain based analyses, i.e., the steady-state analysis and the modal frequency analysis. The key to the evaluation of the transfer function, in turn, is the solution of a linear system with the system Matrix $(s\mathbf{E} + \mathbf{A}) \in \mathbb{C}^{n \times n}$.[8]

Note that at the very core of any numerical time integration scheme applied in transient simulation we have to solve as well linear equations with system matrices of the form $\alpha\mathbf{E} + \mathbf{A}$ were $\alpha \in \mathbb{R}$ depends on some coefficient characteristic to the method and the stepsize used.

It is the order n of the problem, i.e., the dimension of the state space that determines how much computational work has to be spend to compute the p output quantities. Usually, the order n in circuit simulation is very large, whereas the dimension of the output is rather small.

The idea of *model order reduction (MOR)* is to replace the high dimensional problem by one of reduced order such that the reduced order model produces an output similar to the output of the original problem when excited with the same input.

Before we give an overview of some of the most common MOR techniques we specify the requirement a reduced order model should satisfy. Again, we just briefly describe some concepts. For a more detailed discussion we refer to the textbook [1].

4.1.2.1 Approximation

The output of the ersatz model should approximate the output of the original model for the same input signal. There are various measures for "being an approximation". In fact these different viewpoints form the basis for different reduction strategies.

[8] Note that here we see the necessity of $\{\mathbf{E}, \mathbf{A}\}$ being a regular matrix pencil.

We give first a Theorem (for an ODE) that confirms how an approximation in the frequency domain leads to an accurate result in the time domain. Let $I_\omega \subset \mathbb{R}$ be a closed interval (but may be $I_\omega = \{\omega_0\}$ or $I_\omega = \mathbb{R}$). For convenience we assume single input $u(t)$ and single output $y(t)$, with transfer function $H(s)$ in the frequency domain between the Laplace transforms $U(s)$ and $Y(s)$. Let $\tilde{H}(s)$ be the approximation to $H(s)$ which gives $\tilde{Y}(s) = \tilde{H}(s)\ U(s)$ and $\tilde{y}(t)$ as the output approximation in the time domain.

Theorem 4.1 *Let $||u(t)||_{L^2([0,\infty))} < \infty$ and $U(i\omega) = 0$ for $\omega \notin I_\omega$. If the system (4.4a) consists of ODEs, then we have the estimate*

$$\max_{t>0} |y(t) - \tilde{y}(t)| \leq (\frac{1}{2\pi} \int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega)^{\frac{1}{2}} (\int_0^\infty |u(t)|^2 dt)^{\frac{1}{2}}. \tag{4.9}$$

Proof We obtain by using the Cauchy-Schwarz inequality in $L^2(I_\omega)$

$$\begin{aligned}
\max_{t>0} |y(t) - \tilde{y}(t)| &\leq \max_{t>0} |\frac{1}{2\pi} \int_{\mathbb{R}} (Y(i\omega) - \tilde{Y}(i\omega)) e^{i\omega t} d\omega| \\
&\leq \max_{t>0} \frac{1}{2\pi} \int_{\mathbb{R}} |Y(i\omega) - \tilde{Y}(i\omega)| \cdot |e^{i\omega t}|\ d\omega \\
&= \frac{1}{2\pi} \int_{\mathbb{R}} |H(i\omega) - \tilde{H}(i\omega)| \cdot |U(i\omega)|\ d\omega \\
&= \frac{1}{2\pi} \int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)| \cdot |U(i\omega)|\ d\omega \\
&\leq \frac{1}{2\pi} (\int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega)^{\frac{1}{2}} (\int_{I_\omega} |U(i\omega)|^2\ d\omega)^{\frac{1}{2}} \\
&\leq (\frac{1}{2\pi} \int_{I_\omega} |H(i\omega) - \tilde{H}(i\omega)|^2 d\omega)^{\frac{1}{2}} (\int_0^\infty |u(t)|^2 dt)^{\frac{1}{2}}.
\end{aligned}$$

This completes the proof. □

We note that for $I_\omega = \mathbb{R}$ the above error estimate is already found in [23], also for parameterized problems. In [42] the more general case I_ω is considered and applied to Uncertainty Quantification for parameterized problems. In MOR the error estimate becomes often small in an interval I_ω sufficiently close to the used expansion point.

Besides producing similar outputs, the reduced order model should behave similar to the original model in various aspects, which we discuss next.

4.1.2.2 Stability

One of the principal concepts of analyzing dynamical systems is its stability.

An autonomous dynamical system, i.e., a system without input is called *stable* if the solution trajectories are bounded in the time domain. For a linear autonomous system the system matrices determine whether it is stable or unstable. Considering for instance the network equation (4.3a) with $\mathbf{B} \equiv \mathbf{0}$ we have to calculate the generalized eigenvalues[9] $\{\lambda_i(\mathbf{A}, -\mathbf{E}), i = 1, \ldots, n\}$ of the matrix pair $(\mathbf{A}, -\mathbf{E})$ to decide whether or not the system is stable. The system is stable if, and only if, all generalized eigenvalues have non-positive real parts and all generalized eigenvalues with $\mathrm{Re}(\lambda_i(\mathbf{A}, -\mathbf{E})) = 0$ are simple.

4.1.2.3 Passivity

For input-output systems of the form (4.4), stability is not strong enough. If nonlinear components are connected to a stable system it can become unstable.

For square systems, i.e., system where the number of inputs is equal to the number of outputs, $p = m$, a property called *passivity* can be defined. This property is much stronger than stability: it means that a system is unable to generate energy.

Here, an inspection of the system's transfer function yields evidence if the system is passive or not. A necessary and sufficient conditions for a square system to be passive is that the transfer function is *positive real*. This means that

- $\mathbf{H}(s)$ is analytic for $\mathrm{Re}(s) > 0$;
- $\mathbf{H}(\bar{s}) = \overline{\mathbf{H}(s)}$, for all $s \in \mathbb{C}$;
- The Hermitian part of $\mathbf{H}(s)$ is symmetric positive, i.e.: $\mathbf{H}^H(s) + \mathbf{H}(s) \geq 0$, for all s with $\mathrm{Re}(s) > 0$ [50]. Here ^{H} means the transposed conjugate complex: $\mathbf{A}^H = \bar{\mathbf{A}}^T$.

The second condition is satisfied for real systems and the third condition implies the existence of a rational function with a stable inverse. Any congruence transformation applied to the system matrices satisfies the previous conditions if the original system satisfies them, and so preserves passivity of the system if the following conditions are true:

- The system matrices are positive definite, $\mathbf{E}$, $\mathbf{A} \geq 0$.
- $\mathbf{B} = \mathbf{C}^T, \mathbf{D} = 0$.

These conditions are sufficient, but not necessary. They are usually satisfied in the case of electrical circuits, which makes congruence-based projection methods very popular in circuit simulation.

[9] For a matrix pair $(\mathbf{A}, \mathbf{B})$ λ is a generalized eigenvalue with a generalized eigenvector $\mathbf{v}$, if $\mathbf{A}\mathbf{v} = \lambda\mathbf{B}\mathbf{v}$.

4.1.2.4 Structure Preservation

For the case of having a circuit made up of linear elements only we have seen in (4.3b) that the system matrices exhibit a block structured form. Furthermore we recognized that the system matrices are sparse. In fact, the same properties hold for the linear case (4.3a) also.

As a consequence, the matrices of the form $(\xi \mathbf{E} + \mathbf{A})$ that have to be decomposed during the different modes of analysis exhibit already a form that can be exploited when solving the corresponding linear systems.

If the full system (4.4) is replaced by a small dimensional system, it would be most desirable if that ersatz system again has a structure similar to the structure of the full problem. Namely, a block structure should be preserved and the system matrix arising from the reduced order model should be sparse as well, as it can be more expensive to decompose a small dense matrix then a larger sparse one.

4.1.2.5 Realizability

Preserving the block structure, as just mentioned, is crucial for *realizing* a reduced order model again as an RLC-circuit again. Another prerequisit for a reduced order model to be synthesizable is *reciprocity*.[10] This is a special form of symmetry of the transfer function **H**. We will not give details here but refer to [44] for a precise definition and MOR techniques and to [6] for other reciprocity preserving MOR techniques.

There is an ongoing discussion if it is necessary to execute this realization (also referred to as *un-stamping*). It is worthwhile mentioning two benefits of that

- An industrial circuit simulator does in fact never create the MNA equations. Actually, a circuit is given in the form of a netlist, i.e., a table where each line correspond to one element. Each time a system has to be solved, the simulator runs through that list, evaluates each element and stamps the corresponding value in the correct places of the system matrix and the corresponding right-hand side. If a reduced order model is available in the form of such a table as well, the simulator can treat that ersatz model like any other subcircuit and does not have to change to a different mode of including the contribution of the subsystem to the overall system.
- A synthezised reduced order model can provide more insight to the engineers and designers than the reduced order model in mathematical form [52].

[10] A two-terminal element is said to be reciprocal, if a variation of the values of one terminal immediately has the reverse effect on the other terminal's value. Linear characteristics obviously have this property.

4.1.3 MOR Methods

We recall the idea of model order reduction (MOR):

Replace a high dimensional problem, say of order n by one of reduced order $r \ll n$ such that the two input-output systems produce a similar output when excited with the same input. Furthermore the reduced order problem should conserve the characteristics of the full model it was derived from.

In fact there is a need for MOR techniques in various fields of applications and for different kind of problem structures. Although a lot of effort is being spent on deriving reliable MOR methods for, e.g., nonlinear problems of the form (4.5) and for linear time varying (LTV) problems – these are problems of the form (4.4) where the system matrices $\mathbf{E}, \mathbf{A}, \ldots$ depend on time t – MOR approaches for linear time systems, or, more precisely, for linear time invariant (LTI) systems, are best understood and are technically mature.

The outcome of MOR applied to the linear state space problem (4.4) is an ersatz system of the form

$$\mathbf{0} = \hat{\mathbf{E}}\dot{\mathbf{z}}(t) + \hat{\mathbf{A}}\mathbf{z}(t) + \hat{\mathbf{B}}\mathbf{u}(t), \tag{4.10a}$$

$$\tilde{\mathbf{y}}(t) = \hat{\mathbf{C}}\mathbf{z}(t) + \hat{\mathbf{D}}\mathbf{u}(t), \tag{4.10b}$$

with state variable $\mathbf{z}(t) \in \mathbb{R}^r$, output $\tilde{\mathbf{y}}(t) \in \mathbb{R}^p$ and system matrices $\hat{\mathbf{E}}, \hat{\mathbf{A}} \in \mathbb{R}^{r\times r}$, $\hat{\mathbf{B}} \in \mathbb{R}^{r\times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{p\times r}$ and $\hat{\mathbf{D}} \in \mathbb{R}^{p\times m}$. The order r of this system is much smaller than the order n of the original system (4.4).

There are many ways to derive such a reduced order model and there are several possibilities for classifying these approaches. It is beyond the scope of this introductory chapter to give a detailed description of all the techniques – for this we refer to [3] and to the textbooks [1, 7, 51] and the papers cited therein.

We classify MOR approaches in projection and truncation based techniques. For each of the two classes we reflect two methods that can be seen as the basis for current developments. Note, that actually it is not possible to draw a sharp line. In fact all MOR techniques aim at keeping major information and removing the less important one. It is in how they measurure importance that the methods differ. In fact several current developments can be regarded as a hybridization of different techniques.

4.1.4 Projection Based MOR

The concept of all projection based MOR techniques is to approximate the high dimensional state space vector $\mathbf{x}(t) \in \mathbb{R}^n$ with the help of a vector $\mathbf{z}(t) \in \mathbb{R}^r$ of reduced dimension $r \ll n$, within the meaning of

$$\mathbf{x}(t) \approx \tilde{\mathbf{x}}(t) := \mathbf{V}\mathbf{z}(t) \quad \text{with } \mathbf{V} \in \mathbb{R}^{n\times r}.$$

Note that the first approximation may be interpreted as a wish. We will only aim for $\mathbf{y}(t) \approx \tilde{\mathbf{y}}(t) = \mathbf{C}\mathbf{V}\mathbf{z}(t) + \hat{\mathbf{D}}\mathbf{u}(t)$. The columns of the matrix $\mathbf{V}$ are a basis of a subspace $\tilde{\mathscr{M}} \subseteq \mathbb{R}^n$, i.e., the state space $\mathscr{M}$, the solution $\mathbf{x}(t)$ of the network equation (4.4a) resides in, is projected on $\tilde{\mathscr{M}}$. A reduced order model, representing the full problem (4.4) results from deriving a state space equation that determines the reduced state vector $\mathbf{z}(t)$ such that $\tilde{\mathbf{x}}(t)$ is a reasonable approximation to $\mathbf{x}(t)$.

If we insert $\tilde{\mathbf{x}}(t)$ on the right-hand side of the dynamic part of the input-output problem (4.4a), it will not vanish identically. Instead we get a residual:

$$\mathbf{r}(t) := \mathbf{E}\mathbf{V}\dot{\mathbf{z}}(t) + \mathbf{A}\mathbf{V}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t) \quad \in \mathbb{R}^n .$$

We can not demand $\mathbf{r}(t) \equiv \mathbf{0}$ in general as this would state an overdetermined system for $\mathbf{z}(t)$. Instead we apply the Petrov-Galerkin technique, i.e., we demand the residual to be orthogonal to some testspace $\mathscr{W}$. Assuming that the columns of a matrix $\mathbf{W} \in \mathbb{R}^{n\times r}$ span this testspace, the mathematical formulation of this orthogonality becomes

$$\mathbf{0} = \mathbf{W}^T\mathbf{r}(t) = \mathbf{W}^T\left(\mathbf{E}\mathbf{V}\dot{\mathbf{z}}(t) + \mathbf{A}\mathbf{V}\mathbf{z}(t) + \mathbf{B}\mathbf{u}(t)\right) \quad \in \mathbb{R}^r,$$

which states a differential equation for the reduced state $\mathbf{z}(t)$.

Defining

$$\begin{aligned} &\hat{\mathbf{E}} := \mathbf{W}^T\mathbf{E}\mathbf{V} \in \mathbb{R}^{r\times r}, \quad &&\hat{\mathbf{A}} := \mathbf{W}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{r\times r},\\ &\hat{\mathbf{B}} := \mathbf{W}^T\mathbf{B} \in \mathbb{R}^{r\times m}, &&\hat{\mathbf{C}} := \mathbf{C}\mathbf{V} \in \mathbb{R}^{p\times r},\\ &\hat{\mathbf{D}} := \mathbf{D} \in \mathbb{R}^{p\times m}, \end{aligned} \tag{4.11}$$

we arrive at the reduced order model (4.10).

To relate $\mathbf{V}$ and $\mathbf{W}$ we demand biorthogonality of the spaces $\mathscr{V}$ and $\mathscr{W}$ spanned by the columns of the two matrices, respectively, i.e. $\mathbf{W}^T\mathbf{V} = \mathbf{I}_r$. With this, the reduced problem (4.10) is the projection of the full problem (4.4) onto $\mathscr{V}$ along $\mathscr{W}$. If an orthonormal $\mathbf{V}$ and $\mathbf{W} = \mathbf{V}$ is chosen, we speak of an orthogonal projection on the space $\mathscr{V}$ (and we come down to a Galerkin method).

Now, MOR projection methods are characterised by the way of how to construct the matrices $\mathbf{V}$ and $\mathbf{W}$ that define the projection. In the following we find a short introduction of *Krylov methods* and *POD* approaches. The former starts from the frequency domain representation, the latter from the time domain formulation of the input-output problem.

4.1.4.1 Krylov Method

Krylov-based methods to MOR are based on a series expansion of the transfer function $\mathbf{H}$. The idea is to construct a reduced order model such that the series expansions of the transfer function $\hat{\mathbf{H}}$ of the reduced model and the full problem's transfer function agree up to a certain index of summation.

In the following we will assume that the system under consideration does not have a direct feedthrough, i.e., (4.6a) is satisfied. Furthermore we restrict to SISO systems, i.e., single input single output systems. In this case we have $p = m = 1$, i.e., $\mathbf{B} = \mathbf{b}$ and $\mathbf{C} = \mathbf{c}^H$ where $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$, and the (scalar) transfer function becomes:

$$\mathbf{H}(s) = -\mathbf{c}^H\,(s\mathbf{E} + \mathbf{A})^{-1}\,\mathbf{b} \quad \in \mathbb{C},$$

As $\{\mathbf{E}, \mathbf{A}\}$ is a regular matrix pencil we can find some frequency s_0 such that $s_0\mathbf{E} + \mathbf{A}$ is regular (for a good discussion on how to choose such "expansion points" s_0, see [17]). Then the transfer function can be reformulated as

$$\mathbf{H}(s) = \mathbf{l}\,(\mathbf{I}_n - (s - s_0)\mathbf{F})^{-1}\,\mathbf{r}, \tag{4.12}$$

with $\mathbf{l} := -\mathbf{c}^H$, $\mathbf{r} := -(s_0\mathbf{E} + \mathbf{A})^{-1}\mathbf{b}$ and $\mathbf{F} := (s_0\mathbf{E} + \mathbf{A})^{-1}\mathbf{A}$.

In a neighbourhood of s_0 one can replace the matrix inverse in (4.12) by the corresponding Neumann series. Hence, a series expansion of the transfer function is

$$\mathbf{H}(s) = \sum_{k=0}^{\infty} m_k (s - s_0)^k \quad \text{with} \quad m_k := \mathbf{l}\,\mathbf{F}^k\,\mathbf{r} \quad \in \mathbb{C}. \tag{4.13}$$

The quantities m_k for $k = 0, 1, \ldots$ are called *moments* of the transfer function.

A different model, of lower dimension, can now be considered to be an approximation to the full problem, if the moments $\hat{m}_k$ of the new model's transfer function $\hat{\mathbf{H}}(s)$ agree with the moments m_k defined above, for $k = 1, \ldots, q$ for some $q \in \mathbb{N}$.

AWE [38], the *Asymptotic Waveform Evaluation*, was the first MOR method that was based on this idea. However, the explicit computation of the moments m_k, which is the key to AWE, is numerically unstable. This method can, thus, only be used for small numbers q of moments to be matched.

Expressions like $\mathbf{F}^k\,\mathbf{r}$ or $\mathbf{l}\,\mathbf{F}^k$ arise also in methods, like Krylov-subspace-methods, which are used for the iterative solution of large algebraic equations. Here the Lanczos- and the Arnoldi-method are algorithms that compute biorthogonal bases $\mathbf{W}, \mathbf{V}$ or a orthonormal basis $\mathbf{V}$ of the μth left and/or right Krylov subspaces

$$\mathscr{K}_l(\mathbf{F}^T, \mathbf{l}^T, \mu) := \operatorname{span}\left(\mathbf{l}^T, \mathbf{F}^T\,\mathbf{l}^T, \ldots, \left(\mathbf{F}^T\right)^{\mu-1}\,\mathbf{l}^T\right),$$
$$\mathscr{K}_r(\mathbf{F}, \mathbf{r}, \mu) := \operatorname{span}\left(\mathbf{r}, \mathbf{F}\,\mathbf{r}, \ldots, \mathbf{F}^{\mu-1}\,\mathbf{r}\right),$$

for $\mu \in \mathbb{N}$, respectively in a numerically robust way.

The Krylov subspaces, thus "contain" the moments m_k of the transfer function and it can be shown, e.g., [2, 12], that from applying Krylov-subspace methods, reduced order models can be created. These reduced order models, however, did not arise from a projection approach. In fact, the Lanczos- and the Arnold-algorithm produces besides the matrices $\mathbf{W}$ and/or $\mathbf{V}$ whose columns span the Krylov subspaces $\mathscr{K}_l$ and/or $\mathscr{K}_r$, respectively, a tridiagonal or an upper Hessenbergmatrix $\mathscr{T}$, respectively. This matrix is then used to postulate a dynamical system whose transfer function has the desired matching property.

Concerning the moment matching property there is a difference for reduced order models created from a Lanczos- and those created from an Arnoldi-based process.

For a fixed q, the Lanczos-process constructs the qth left and the qth right Krylov-subspace, hence biorthogonal matrices $\mathbf{W}, \mathbf{V} \in \mathbb{R}^{n \times q}$. A reduced order model of order q, arising from this procedure possesses a transfer function $\hat{\mathbf{H}}(s)$ whose first $2q$ moments coincide with the first $2q$ moments of the original problem's transfer function $\mathbf{H}(s)$, i.e. $\hat{m}_k = m_k$ for $k = 0, \ldots, 2q-1$. Hence, the Lanczos MOR model yields a Padé approximation.

The Arnoldi method on the other hand is a one sided Krylov subspace method. For a fixed q only the qth right Krylov subspace is constructed. As a consequence, here only the first q moments of the original system's and the reduced system's transfer function match.

Owing to their robustness and low computational cost, Krylov subspace algorithms proved suitable for the reduction of large-scale systems, and gained considerable popularity, especially in electrical engineering. A number of Krylov-based MOR algorithms have been developed, including techniques based on the Lanczos method [9, 19] and the Arnoldi algorithm [36, 56]. Note that the moment matching, mentioned above, can only be valid locally, i.e., for a certain frequency range around the expansion point s_0. However, also Krylov MOR schemes based on a multipoint expansion in the frequency range have been constructed [21].

The main drawbacks of these methods are, in general, lack of provable error bounds for the extracted reduced models, and no guarantee for preserving stability and passivity. There are techniques to turn reduced systems to passive reduced systems. However, this introduced some post-processing of the model [18].

4.1.4.2 Passivity Preservation

Odabasioglu et al. [36] turned the Krylov based MOR schemes into a real projection method. In addition, the developed scheme, *PRIMA* (Passive Reduced-Order Interconnect Macromodeling Algorithm), is able to preserve passivity.

This MOR technique can be applied to electrical circuits that contain only passive linear resistors, capacitors and inductors and which accepts only currents as input at the terminals. One says that the RLC-circuit is in impedance form, i.e., the inputs $\mathbf{u}(t)$ are currents and the outputs $\mathbf{y}$ are voltages.

In this case, the system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ have a special structure (cp. (4.3b)), namely:

$$\mathbf{E} = \begin{pmatrix} \mathbf{E}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{E}_2 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ -\mathbf{A}_2^T & \mathbf{0} \end{pmatrix}, \quad \mathbf{B} = \mathbf{C}^T = \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{pmatrix}, \tag{4.14}$$

where $\mathbf{E}_1, \mathbf{A}_1 \in \mathbb{R}^{n_e \times n_e}$ and $\mathbf{E}_2 \in \mathbb{R}^{n_L \times n_L}$ and are symmetric non-negative definit matrices.

In *PRIMA*, first the Arnoldi method is applied to create the projection matrix $\mathbf{V}$. Then, choosing $\mathbf{W} = \mathbf{V}$, the system matrices are reduced according to (4.11). For several implementational details, covering Block-Arnoldi as well as deflation, see [55]. The reduced order model arising in this way can be shown to be passive [36]. The key to these findings is the above special structure of linear RLC-circuits in (4.14).

It is, however, not necessary, to use the Arnoldi method to construct the matrix $\mathbf{V}$. Furthermore, it is also possible to apply the technique to systems in admittance form, i.e., where the inputs are voltages and the outputs are currents. For more details we refer to [27] in this book.

4.1.4.3 Structure Preservation

As we have seen *PRIMA* takes advantage of the special block structure (4.14) of linear RLC circuits to create passive reduced order models. The structure, however, is not preserved during the reduction. This makes it hard to synthesise the model, i.e., realize the reduced model as an RLC circuit again.

Freund [12–16] developed a Krylov-based method where passivity, the structure and reciprocity are preserved. *SPRIM* (Structure-Preserving Reduced-Order Interconnect Macromodell) is similar to *PRIMA* as first the Arnoldi-method is run to create a matrix $\mathbf{V} \in \mathbb{R}^{n \times r}$. This, however, is not taken as the projection matrix directly. Instead, the matrix $\mathbf{V}$ is partitioned to

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{pmatrix} \quad \text{with} \quad \mathbf{V}_1 \in \mathbb{R}^{n_e \times r}, \mathbf{V}_2 \in \mathbb{R}^{n_L \times r},$$

corresponding to the block structure of the system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$.

Finally, after re-orthogonalization, the blocks $\mathbf{V}_1, \mathbf{V}_2$ are rearranged to the matrix

$$\hat{\mathbf{V}} = \begin{pmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{pmatrix} \in \mathbb{R}^{n \times (2r)}, \tag{4.15}$$

which is then used to transform the system to a reduced order model, according to the transformations given in (4.11) (with $\mathbf{V} = \mathbf{W} = \hat{\mathbf{V}}$).

It can be shown, that the *SPRIM*-model preserves twice as many moments as the *PRIMA*, if the same Arnoldi-method is applied. Note, however, that the dimension also increases by a factor 2.

4.1.4.4 Multi-input Multi-output

For the general case, where p and m are larger than one, i.e., when we have multiple inputs and multiple outputs, the procedure carried out by the Krylov MOR methods is in principle the same. In this case however, Krylov subspaces for multiple starting vectors have to be computed and one has to take care, when a "breakdown" or a "near-breakdown" occurs, that is, when the basis vectors constructed for differing starting vectors, $\mathbf{r}_1$ and $\mathbf{r}_2$ become linearly dependent. In this case the progress for the Krylov subspace becoming linear dependent has to be stopped. The Krylov subspace methods arising from that considerations are called *Block Krylov methods*. For a detailed discussion we refer to the literature given above.

4.1.4.5 POD Method

While the Krylov approaches are based on the matrices, i.e., on the system itself, the method of Proper Orthogonal Decomposition (*POD*) is based on the trajectory $\mathbf{x}(t)$, i.e., the outcome of the system (4.4). One could also say that the Krylov methods are based on the frequency domain, whereas POD is based on the time domain formulation of the input output system to be modelled.

POD first collects data $\{\mathbf{x}_1, \dots, \mathbf{x}_K\}$. The datapoints are snapshots of the state space solution $\mathbf{x}(t)$ of the network equation (4.4a) at different timepoints t or for different input signals $\mathbf{u}(t)$. They are usually constructed by a numerical time simulation, but may also arise from measurements of a real physical system.

From analysing this data, a subspace is created such that the data points as a whole are approximated by corresponding points in the subspace in a optimal least-squares sense. The basis of this approach is also known as *Principal Component Analysis* and *Karhunen–Loève Theorem* from picture and data analysis.

The mathematical formulation of POD [39] is as follows: Given a set of K datapoints $X := \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$ a subspace $\mathscr{S}_r \subset \mathbb{R}^n$ of dimension r is searched for that minimizes

$$\|X - \varrho_r X\|_2^2 := \frac{1}{K} \sum_{k=1}^{K} \|\mathbf{x}_k - \varrho_r \mathbf{x}_k\|_2^2, \tag{4.16}$$

where $\varrho_r : \mathbb{R}^n \to \mathscr{S}_r$ is the orthogonal projection onto $\mathscr{S}_r$.

We will not describe POD in full detail here, as in literature, e.g., [1, 39], this is well explained. However, the key to solving this minimization problem is the computation of the eigenvalues λ_i and eigenvectors φ_i (for $i = 1, \dots, n$) of the correlation matrix $\mathbf{X}\mathbf{X}^T$:

$$\mathbf{X}\mathbf{X}^T \boldsymbol{\varphi}_i = \lambda_i \boldsymbol{\varphi}_i,$$

where the eigenvalues and eigenvectors are sorted such that $\lambda_1 \geq \cdots \geq \lambda_n$. The matrix $\mathbf{X}$ is defined as $\mathbf{X} := (\mathbf{x}_1, \ldots, \mathbf{x}_K) \in \mathbb{R}^{n \times K}$ and is called *snapshot matrix*.

Intuitively the correlation matrix detects the principal directions in the data cloud that is made up of the snapshots $\mathbf{x}_1, \ldots, \mathbf{x}_K$. The eigenvectors and eigenvalues can be thought of as directions and radii of axes of an ellipsoid that incloses the cloud of data. Then, the smaller the radii of one axis is, the less information is lost if that direction is neglected.

The question arises, how many directions r should be kept and how many can be neglected. There is no a-priori error bound for the POD reduction (Rathinam and Petzold [43], though, perform a precise analysis of the POD accuracy). However, the eigenvalues are a measure for the relevance of the dimensions of the state space. Hence, it seems reasonable to choose the dimension r of the reduced order model in such a way, that the relative information content of the reduced model with respect to the full system is high. The measure for this content, used in the literature cited above is

$$\mathscr{I}(r) = \frac{\lambda_1 + \cdots \lambda_r}{\lambda_1 + \cdots \lambda_r + \lambda_{r+1} + \cdots \lambda_n}.$$

Clearly, a high relative information content means to have $\mathscr{I}(r) \approx 1$. Typically r is chosen such that this measure is around 0.99 or 0.995.

If the eigenvalues and eigenvectors are available and a dimension r has been chosen, the projection matrices $\mathbf{V}$ and $\mathbf{W}$ in (4.11) are taken as

$$\mathbf{V} := \mathbf{W} := (\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_r) \in \mathbb{R}^{n \times r}\,.$$

leading to an orthogonal projection $\varrho_r = \mathbf{V}\mathbf{V}^T$ on the space $\mathscr{S}_r$ spanned by $\varphi_1, \ldots, \varphi_r$.

The procedure described so far relies on the eigenvalue decomposition of the $n \times n$ matrix $\mathbf{X}\mathbf{X}^T$. This direct approach is feasible only for problems of moderate size. For high dimensional problems, i.e., for dimensions $n \gg 1$, the eigenvalues and eigenvectors are derived form the Singular Value Decomposition (SVD) of the snapshot matrix $\mathbf{X} \in \mathbb{R}^{n \times K}$.

The SVD provides three matrices:

$$\begin{aligned}
\boldsymbol{\Phi} &= (\varphi_1, \cdots, \varphi_n) \in \mathbb{R}^{n \times n} \quad \text{orthogonal},\\
\boldsymbol{\Psi} &= (\psi_1, \cdots, \psi_K) \in \mathbb{R}^{K \times K} \quad \text{orthogonal},\\
\boldsymbol{\Sigma} &= \operatorname{diag}(\sigma_1, \ldots, \sigma_\nu) \in \mathbb{R}^{\nu \times \nu} \quad \text{with } \sigma_1 \geq \cdots \geq \sigma_\nu > \sigma_{\nu+1} = \ldots = \sigma_K = 0,
\end{aligned}$$

such that

$$\mathbf{X} = \boldsymbol{\Phi} \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \boldsymbol{\Psi}^T, \tag{4.17}$$

where the columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are the left and right singular eigenvectors, respectively, and $\sigma_1, \ldots, \sigma_\nu$ are the singular values of $\mathbf{X}$ (σ_ν being the smallest non-zero singular value; this also defines the index ν). It follows that $\varphi_1, \ldots, \varphi_n$ are eigenvectors of the correlation matrix $\mathbf{X}\mathbf{X}^T$ with the n eigenvalues $\sigma_1^2, \ldots, \sigma_\nu^2, 0, \ldots, 0$.

4.1.5 Truncation Based MOR

The MOR approaches we reviewed so far rely on the approximation of the high-dimensional state space, the solution of (4.4) resides in, by an appropriate space of lower dimension. An equation for the correspondent $\mathbf{z}(t)$ of $\mathbf{x}(t)$ is derived by constructing a projection onto that lower-dimensional space.

Although the approaches we are about to describe in the following can also be considered as projection methods in a certain sense, we decided to present them separately. What makes them different, is that these techniques base on preserving key characteristics of the system rather than reproducing the solution. We will get aquainted with an ansatz based upon energy considerations and an approach meant to preserve poles and zeros of the transfer function.

4.1.5.1 Balanced Truncation

The technique of Balanced Truncation, introduced by Moore [35], is based on control theory, where one essentially investigates how a system can be steered and how its reaction can be observed. In this regard, the basic idea of *Balanced Truncation* is to first classify, which states $\mathbf{x}$ are hard to reach and which states $\mathbf{x}$ are hard to deduce from observing the output $\mathbf{y}$, then to reformulate the system such that the two sets of states coincide and finally truncate the system such that the reduced system does not attach importance to these problematic cases.

The system (4.4) can be driven to the state $\bar{\mathbf{x}}$ in time T if an input $\bar{\mathbf{u}}(t)$, with $t \in [0, T]$ can be defined such that the solution at time T, i.e., $\mathbf{x}(T)$ takes the value $\bar{\mathbf{x}}$ where $\mathbf{x}(0) = \mathbf{0}$. We perceive the L_2-norm $\|\cdot\|_2$, with $\|\bar{\mathbf{u}}\|_2^2 = \int_0^T \bar{\mathbf{u}}(t)^T \bar{\mathbf{u}}(t)\, dt$ as energy of the input signal. If the system is in state $\tilde{\mathbf{x}}$ at time $t = 0$ and no input is applied at its ports we can observe the output $\tilde{\mathbf{y}}(t)$ for $t \in [0, T]$ and the energy $\|\tilde{\mathbf{y}}\|_2$ emitted at the system's output ports.

We consider a state as hard to reach if the minimal energy needed to steer the system to that state is large. Similarly, a state whose output energy is small leaves a weak mark and is therefore considered to be hard to be observed.

The minimal input energy needed and the maximal energy emitted can be calculated via the finite and the infinite *controllability Gramian*

$$\mathscr{P}(T) = \int_0^T e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt \quad \text{and} \quad \mathscr{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t} dt \tag{4.18a}$$

and the finite and infinite *observability Gramian*

$$\mathscr{Q}(T) = \int_0^T e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t}dt \quad \text{and} \quad \mathscr{Q} = \int_0^\infty e^{\mathbf{A}^T t}\mathbf{C}^T\mathbf{C}e^{\mathbf{A}t}dt, \tag{4.18b}$$

respectively. Note that the system (4.4) is assumed to be stable. Furthermore, the above definition is valid for the case $\mathbf{E} = \mathbf{I}_{n\times n}$. The latter does not mean a limitation of the method of Balanced Truncation to standard state space systems. In fact, these considerations can be applied to descriptor systems as well, e.g., [54].

With the above definitions one can prove that the minimal energy needed, i.e., the energy connected to the most economical input $\bar{\mathbf{u}}$, to reach the state $\bar{\mathbf{x}}$ holds

$$\|\bar{\mathbf{u}}\|_2^2 = \bar{\mathbf{x}}^T\mathscr{P}^{-1}\bar{\mathbf{x}}.$$

Similarly, the energy emitted due to the state $\tilde{\mathbf{x}}$ holds

$$\|\tilde{\mathbf{y}}\|_2^2 = \tilde{\mathbf{x}}\mathscr{Q}\tilde{\mathbf{x}}.$$

The Gramians are positive definite. Applying a diagonalization of the controllability Gramian, it is easy to see that states that have a large component in the direction of eigenvectors corresponding to small eigenvalues of $\mathscr{P}$ are hard to reach. In the same way it is easy to see that states pointing in the direction of eigenvectors to small eigenvalues of the observability Gramian $\mathscr{Q}$ are hard to observe.

The basic idea of the Balanced Truncation MOR approach is to neglect states that are both hard to reach and hard to observe. This marks the truncation part. However, to reach this synchrony of a state being both hard to reach and hard to observe, the basis of the state space has to be transformed. This marks the balancing part. Generally, a basis transformation introduces new coordinates $\tilde{\mathbf{x}}$ such that $\mathbf{x} = \mathbf{T}^{-1}\tilde{\mathbf{x}}$ where $\mathbf{T}$ is the matrix representation of the basis transformation. Here the Gramians transform equivalently to

$$\tilde{\mathscr{P}} = \mathbf{T}\mathscr{P}\mathbf{T}^T \quad \text{and} \quad \tilde{\mathscr{Q}} = \mathbf{T}^{-1}\mathscr{Q}\mathbf{T}^{-T}.$$

The transformation $\mathbf{T}$ is called *balancing transformation* and the system arising from applying the transformation to the system (4.4) is called *balanced* if the transformed Gramians satisfy

$$\tilde{\mathscr{P}} = \tilde{\mathscr{Q}} = \operatorname{diag}(\sigma_1, \ldots, \sigma_n). \tag{4.19}$$

The values $\sigma_1, \ldots, \sigma_n$ are called *Hankel Singular Values*. They are the positive square roots of the eigenvalues of the product of the Gramians:

$$\sigma_l = \sqrt{\lambda_k(\mathscr{P}\cdot\mathscr{Q})}, \quad l = 1, \ldots, n.$$

Now we assume that the eigenvalues are sorted in descending order, i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. We introduce the cluster

$$\left(\begin{array}{ccc|ccc} \sigma_1 & & & & & \\ & \ddots & & & & \\ & & \sigma_r & & & \\ \hline & & & \sigma_{r+1} & & \\ & & & & \ddots & \\ & & & & & \sigma_n \end{array}\right) = \begin{pmatrix} \varSigma_1 & \\ & \varSigma_2 \end{pmatrix},$$

and adopt this to the tranformed input-output system[11]

$$\begin{aligned} \mathbf{0} &= \begin{pmatrix} \dot{\tilde{\mathbf{x}}}_1(t) \\ \dot{\tilde{\mathbf{x}}}_2(t) \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{x}}_1(t) \\ \tilde{\mathbf{x}}_2(t) \end{pmatrix} + \begin{pmatrix} \tilde{\mathbf{B}}_1 \\ \tilde{\mathbf{B}}_2 \end{pmatrix} \mathbf{u}(t), \\ \mathbf{y}(t) &= (\mathbf{C}_1, \mathbf{C}_2) \begin{pmatrix} \tilde{\mathbf{x}}_1(t) \\ \tilde{\mathbf{x}}_2(t) \end{pmatrix}, \end{aligned} \tag{4.20}$$

such that $\tilde{\mathbf{x}}_1 \in \mathbb{R}^r$ and $\tilde{\mathbf{x}}_2 \in \mathbb{R}^{n-r}$.

Finally we separate the cluster and derive the reduced order model

$$\mathbf{0} = \dot{\hat{\mathbf{x}}}_{11}(t) + \tilde{\mathbf{A}}_1 \hat{\mathbf{x}}_1(t) + \tilde{\mathbf{B}}_1 \mathbf{u}(t), \tag{4.21a}$$

$$\tilde{\mathbf{y}}_1(t) = \tilde{\mathbf{C}}_1 \hat{\mathbf{x}}_1(t) \tag{4.21b}$$

of dimension $r \ll n$, by skipping the part corresponding to the small eigenvalues $\sigma_{r+1}, \ldots, \sigma_n$ of both Gramians.

Important Properties

Balanced Truncation is an appealing MOR technique because it automatically preserves stability.

Furthermore, and even more attractive is that this MOR approach provides a *computable error bound*: Let $\sigma_{r+1}, \ldots, \sigma_k$ be the different eigenvalues that are truncated. Then, for the transfer function $\mathbf{H}_1$ corresponding to (4.21), it holds

$$\|\mathbf{H} - \mathbf{H}_1\|_{\mathbb{H}_\infty} \leq 2\left(\sigma_{r+1} + \cdots + \sigma_k\right), \tag{4.22}$$

where the $\mathbb{H}_\infty$ norm is defined as $\|\mathbf{H}\|_{\mathbb{H}_\infty} := \sup_{\omega \in \mathbb{R}} \|\mathbf{H}(i\omega)\|_2$ where $\|\cdot\|_2$ is the matrix spectral norm.

[11] To simplify matters we have chosen $\mathbf{E} = \mathbf{I}_{n \times n} = \operatorname{diag}(1, \ldots, 1) \in \mathbb{R}^{n \times n}$ and $\mathbf{D} = \mathbf{0}$.

Computation

Applying the method of Balanced Truncation as presented above makes it necessary to compute the Gramians and the simultaneous diagonalization of the Gramians.

The infinite Gramians $\mathscr{P}$ and $\mathscr{Q}$ are defined by infinity integrals. However, it is not hard to show that they arise from solving the *Lyapunov equations*:

$$\begin{aligned} \mathbf{A}\mathscr{P} + \mathscr{P}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T &= \mathbf{0} \\ \mathbf{A}^T\mathscr{Q} + \mathscr{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} &= \mathbf{0} \end{aligned} \tag{4.23}$$

Having solved the Lyapunov equations, one way to determine the balancing transformation is described by the *square root algorithm* (see e.g. [1]). The basic steps in this approach are the computation of the Cholesky factorisations of the Gramians $\mathscr{P} = \mathbf{S}^T\mathbf{S}$ and $\mathscr{Q} = \mathbf{R}^T\mathbf{R}$ and the singular value decomposition of the product $\mathbf{S}\mathbf{R}^T$.

In the past Balanced Bruncation was not favored because the computation of the solution of the high dimensional matrix equations (4.23) and the balancing was very cumbersome and costly. In recent years however, progress was made in the development of techniques to overcome these difficulties. Techniques that can be applied to realize the Balanced Truncation include the ADI method [30], the sign function method [4] or other techniques, e.g. [49]. For a collection of techniques we also refer to [5].

Poor Man's TBR

Another method that should be mentioned is *Poor Man's TBR,*[12] introduced by Phillips and Silveira [37]. Balanced Truncation relies on the Gramians. The methods we mentioned so far compute these Gramians based on the Lyapunov equations (4.23).

The idea of Poor Man's TBR (PMTBR) however, is to compute the Gramians from their definition (4.18). If the system to be reduced is symmetric, i.e. $\mathbf{A} = \mathbf{A}^T$ and $\mathbf{C} = \mathbf{B}^T$, $\mathscr{P}$ and $\mathscr{Q}$ coincide. The (controllability and observability) Gramian is then defined as

$$\mathscr{P} = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{B}^T e^{\mathbf{A}^T t}\,dt.$$

As the Laplace transform of $e^{\mathbf{A}t}$ is $(s\mathbf{I}-\mathbf{A})^{-1}$, we can apply Parseval's lemma, which says that a signal's energy in the time domain is equal to its energy in the frequency

[12] TBR = Truncated Balanced Realization

domain and transfer the time domain integral to the frequency domain:

$$\mathscr{P} = \int_{-\infty}^{\infty} (i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\mathbf{B}^T (i\omega\mathbf{I} - \mathbf{A})^{-H} d\omega.$$

PMTBR now starts with applying a numerical quadrature scheme: With nodes ω_k, weights w_k and defining $z_k = (i\omega_k\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ an approximation $\hat{\mathscr{P}}$ to the gramian $\mathscr{P}$ can be computed as:

$$\mathscr{P} \approx \hat{\mathscr{P}} = \sum_k w_k\, z_k\, z_k^H = ZW \cdot (ZW)^H,$$

where $Z = (z_1, z_2, \ldots)$ and $W = \text{diag}(\sqrt{w_1}, \sqrt{w_2}, \ldots)$.

For further details on the order reduction we refer to the original paper mentioned above.

4.1.5.2 Modal Truncation

Engineers usually investigate the transfer behavior of an input-output system by inspecting its frequency response $\mathbf{H}(i\omega) =: \mathbf{G}(\omega)$ for frequencies $\omega \in \mathbb{R}^+$. The Bode plot, i.e. the combination of the Bode magnitude and phase plot, expressing how much a signal component with a specific frequency is amplified or attenuated and which phase shift can be observed from in- to output, respectively, is a graphical representation of the frequency response.

One is especially interested in the peaks and sinks of the Bode magnitude plot, which are caused by the *poles* and *zeros* of the transfer function $\mathbf{H}$. The *Modal Truncation* [45] is aimed at constructing a reduced order model (4.10) such that peaks and sinks of the reduced order model's frequency response $\hat{\mathbf{G}}(\omega) = \hat{\mathbf{H}}(i\omega)$ match with the one of the full dynamical problem (4.4).

Applying Cramer's rule it is obvious that the transfer function is a rational function:

$$\mathbf{H}(s) = \frac{p_{n-1}(s)}{q_n(s)},$$

with polynomials p_{n-1} and q_n of degree $n-1$ and n respectively. The zeros of the numerator are the zeros of the transfer function and the zeros of the denominator are its poles.

The generalized eigenvalues of the matrix pencil $\{\mathbf{E}, \mathbf{A}\}$, or the eigenvalues of $\mathbf{A}$, if we assume $\mathbf{E} = \mathbf{I}_{n\times n}$, are the key to the poles of the transfer function. For a more detailed discussion we refer to [28]. To illustrate this relation we restrict to the latter case and consider a SISO system without direct feedtrough, i.e., $\mathbf{D} = \mathbf{0}$.

The eigentriples $(\lambda_i, \mathbf{v}_i, \mathbf{w}_i)$ for $i = 1, \ldots, n$ of $\mathbf{A}$ consist of the ith eigenvalue $\lambda_i \in \mathbb{C}$ and the ith right and left eigenvalue $\mathbf{v}_i, \mathbf{w}_i \in \mathbb{C}^n$, respectively, that satisfy

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad \text{and} \quad \mathbf{w}_i^H \mathbf{A} = \lambda_i \mathbf{w}_i^H.$$

From assuming that $\mathbf{A}$ is diagonalizable it can be derived that

$$\mathbf{L}^H \mathbf{A} \mathbf{R} = \boldsymbol{\Lambda},$$

where $\boldsymbol{\Lambda} = (\lambda_1, \ldots, \lambda_n)$, $\mathbf{R} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ and $\mathbf{L} = (\mathbf{w}_1, \ldots, \mathbf{w}_n) \in \mathbb{C}^{n \times n}$, where the left and right eigenvectors are scaled such that $\mathbf{L}^H \mathbf{R} = \mathbf{I}_{n \times n}$.

We apply a change of coordinates $\mathbf{x} = \mathbf{R}\tilde{\mathbf{x}}$ and multiply the input to state mapping (4.4a) with $\mathbf{L}^H$ which is a projection on the space spanned by the columns of $\mathbf{R}$ along the space spanned by the columns of $\mathbf{L}$. This transforms the input-output system (4.4) to

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}} &= \boldsymbol{\Lambda}\tilde{\mathbf{x}} + \mathbf{L}^H \mathbf{b}\mathbf{u}, \\ \mathbf{y} &= \mathbf{c}^H \mathbf{R}\tilde{\mathbf{x}}. \end{aligned} \tag{4.24}$$

The transformed system is equivalent to the original system (4.4), the (scalar) transfer function can be represented as

$$\mathbf{H}(s) = \sum_{i=1}^{n} \frac{r_i}{s - \lambda_i} \quad \text{with} \quad r_i = \left(\mathbf{c}^H \mathbf{v}_i\right)\left(\mathbf{w}_i^H \mathbf{b}\right) \in \mathbb{C} \quad \text{for } i = 1, \ldots, n. \tag{4.25}$$

This form of displaying the transfer function is known as *Pole-Residue Representation*, where the quantities $r_i \in \mathbb{C}$ are called *residues* and where we can see that the eigenvalues of the matrix $\mathbf{A}$ are the poles of $\mathbf{H}(s)$.

The idea of modal truncation is to replace the full order problem with a reduced order model of say order $r < n$ whose transfer function has a pole-residue representation that is a truncation of the corresponding full model's representation (4.25), i.e.

$$\hat{\mathbf{H}}(s) = \sum_{i=1}^{r} \frac{r_i}{s - \lambda_i}, \tag{4.26}$$

where r_i and λ_i for $i = 1, \ldots, r$ are the same as in (4.25). The corresponding state-space representation (4.10) evolves from carrying out the matrix projections defined in (4.11) where $\mathbf{V}, \mathbf{W} \in \mathbb{C}^{n \times r}$ comprises r right and left eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_r$ and $\mathbf{w}_1, \ldots, \mathbf{w}_r$, respectively. As no new poles arise by constructing the reduced order model in this way, the stability property is inherited from the full order problem.

Immediately the question arises, which pairs (λ_i, r_i) of poles and residues and how many should be taken into account.

Rommes [45] and Martins et al. [31] sort these pairs according to decreasing *dominance* of the pole. Their measure for dominance of a pole is the magnitude of the relation

$$\frac{|r_i|}{|\mathrm{re}(\lambda_i)|}.$$

Hence, modal truncation takes into account the first r poles/residues according to this ordering scheme. The answer to the second part of the question, i.e., how many poles/residues to keep, arises from the error bound [20]

$$\|\mathbf{H} - \hat{\mathbf{H}}\|_{\mathbb{H}_\infty} \leq \sum_{j=r+1}^{n} \frac{|r_j|}{|\mathrm{re}(\lambda_j)|}, \tag{4.27}$$

and hence from the deviation one is willing to tolerate.

The computation of the error bound (4.27) necessitates a full eigenvalue decomposition. This is only feasible for moderate orders $n \leq 2{,}000$. For large scale systems methods using only a partial eigenvalue decomposition can be applied. Here the *Subspace Accelerated Dominant Pole Algorithm (SADPA)*, introduced by Rommes and Martins [46] produces a series of dominant poles. The main principle of SADPA is to search for the zeros of $\frac{1}{\mathbf{H}(s)}$ using a Newton-iteration. As the Newton-iteration can only find one zero sufficiently close to a starting value at a time, the iteration procedure has to be applied several times. In order to find less dominant poles at each time, the system the dominant pole algorithm is applied to is adjusted each time one dominant pole has been found. This adjustment is referred to as *subspace acceleration*.

Again, for further details we refer to the papers cited above.

4.1.6 Other Approaches

We shortly address some other approaches. In [26, 40, 41], port-Hamiltonian systems are considered to guarantee structure preserving reduced models. In [8, 10, 11], vector fitting techniques are used to obtain passivity preserving reduced models. In [25, 32, 47], one matches additional moments of Laurent expansions involving terms with $1/s$. These are applied to obtain passive reduced models for RLC circuits.

4.1.7 Examples

In this part we will introduce linear circuits and reduce them with techniques which have already been discussed. We give results for the methods PRIMA [36], SPRIM [12–16], and PMTBR [37].

In simulation a Bode magnitude plot of the transfer function shows the magnitude of $\mathbf{H}(i\omega)$, in decibel, for a number of frequencies ω in the frequency domain of interest. If the transfer function of the original system can be evaluated at enough points $s = i\omega$ to produce an accurate Bode plot, the original frequency response can be compared with the frequency response of the reduced model. In our examples, $\mathbf{H}$ is a scalar.

4.1.7.1 Example 1

We choose an RLC ladder network [33] shown in Fig. 4.1. We set all the capacitances and inductances to the same value 1 while $R_1 = \frac{1}{2}$ and $R_2 = \frac{1}{5}$, see [34, 53]. We arrange 201 nodes which gives us the order 401 for the mathematical model of the circuit.

If we write the standard MNA formulation the linear dynamical system is derived. The system matrices are as follows (for $K = 3$, for example):

$$\mathbf{E} = \mathbf{I}, \quad \mathbf{A} = \begin{bmatrix} -2 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -5 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 0 \; 0 \; -5 \; 0 \; 0 \end{bmatrix}, \quad D = 5. \tag{4.28}$$

In the state variable $\mathbf{x}$, x_k is the voltage across capacitance C_k $(k = 1, \dots, K)$, or the current through inductor L_{k-K} $(k = K+1, \dots, 2K-1)$. In general the number of nodes K is odd. The voltage u and the current y are input and output, respectively. Note that when the number of nodes is K the order of the system becomes $n = 2K - 1$. In this test case we have an ODE instead of a DAE as $\mathbf{E} = \mathbf{I}$, see (4.28). The original transfer function is shown in Fig. 4.2. The plot already illustrates how difficult it is to reduce this transfer function as many oscillations appear.

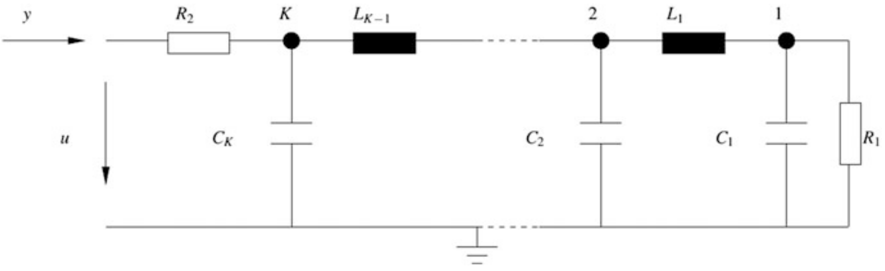


Fig. 4.1 RLC Circuit of order $n = 2K - 1$, Example 1

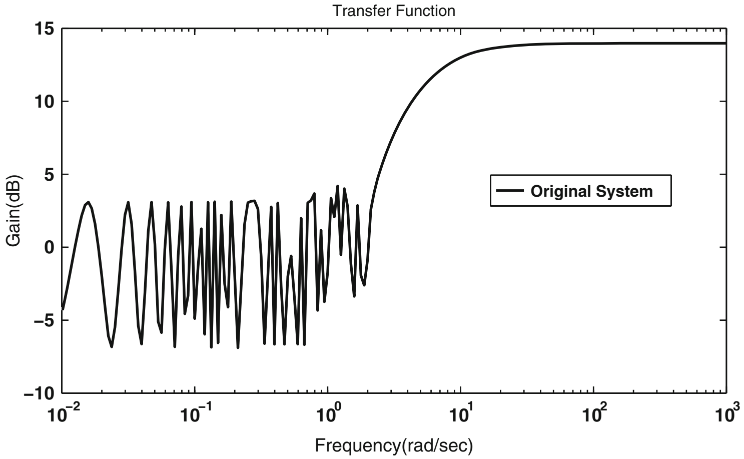


Fig. 4.2 Original transfer function for the first example of Fig. 4.1, order $n = 401$. The frequency domain parameter ω varies between 10^{-2} to 10^{3}

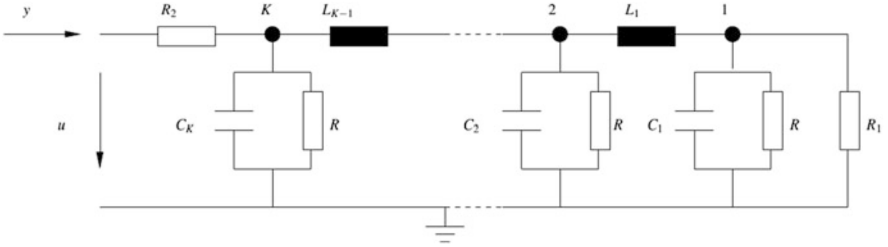


Fig. 4.3 RLC Circuit of order $n = 2K - 1$, Example 2

4.1.7.2 Example 2

Next, we use another RLC ladder network, given in Fig. 4.3 [33, 48], for the second example. The major difference to the previous example is that we introduced a resistor (all of equal value) in parallel to the capacitors at each node connected to the ground. We set all the capacitances and inductances to the same value 1 while $R_1 = \frac{1}{2}$, $R_2 = \frac{1}{5}$ and $R = 1$. We choose 201 nodes which results in a system having order 401 for the mathematical model of the circuit. Like the previous example we again derive a system of ODEs. The original transfer function of the second example is shown in Fig. 4.4.

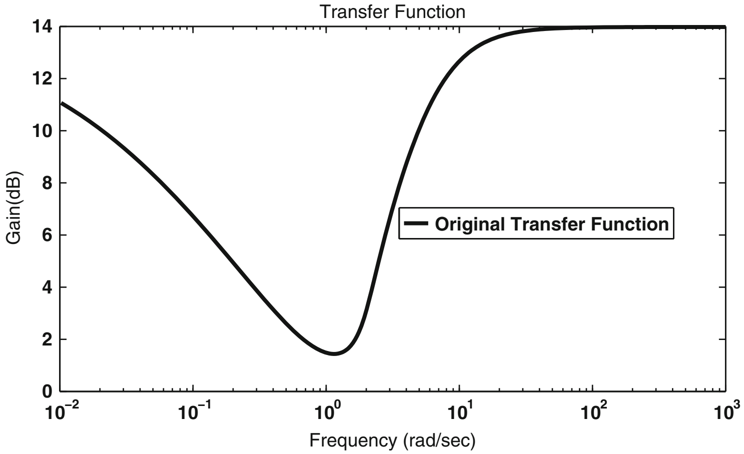


Fig. 4.4 Original transfer function for the second example of Fig. 4.3, order $n = 401$. The frequency domain parameter ω varies between 10^{-2} to 10^{3}

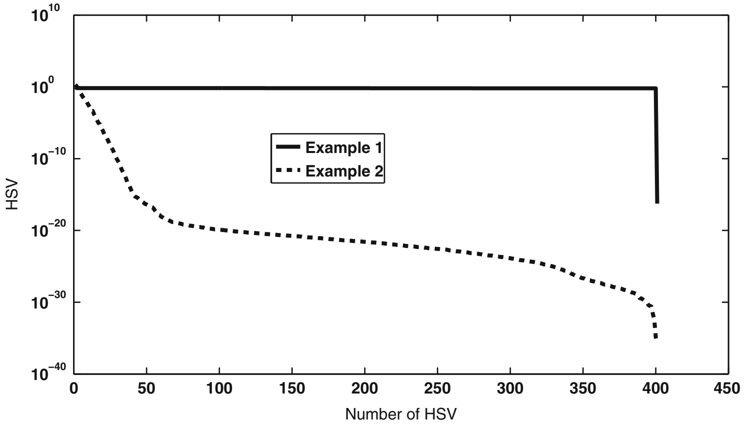


Fig. 4.5 Hankel Singular Values for Example 1 and 2, (semi-logarithmic scale)

4.1.7.3 MOR by PRIMA, SPRIM and PMTBR

The main reason for choosing these two examples is the behavior of the Hankel singular values, see Fig. 4.5. The Hankel singular values for the first example do not show any significant decay, while in the second example we observe a rapid decay in the values. The results are taken from [33].

The Figs. 4.6 and 4.7 show the absolute error between the transfer function of the full system and the transfer function of several reduced systems. The model is reduced by three linear techniques (PRIMA, SPRIM and PMTBR) for both examples.

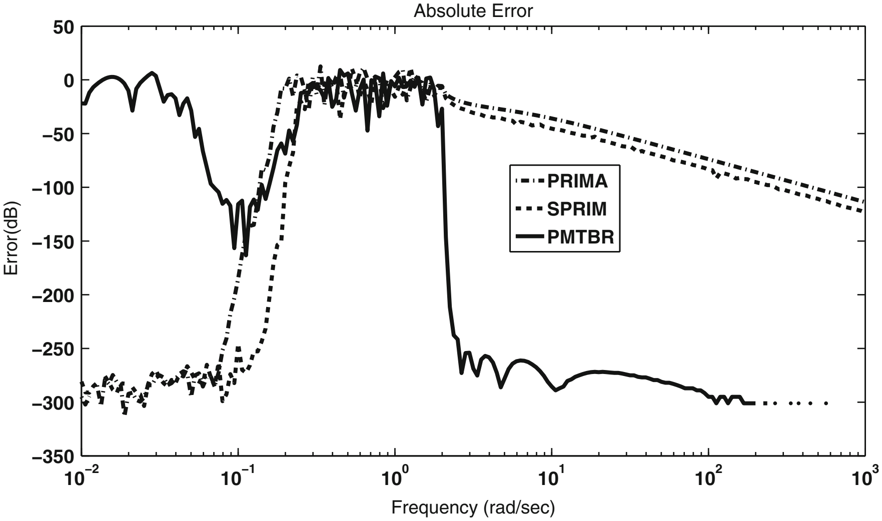


Fig. 4.6 Error plot, the frequency domain parameter ω varies between 10^{-2} to 10^3, Example 1

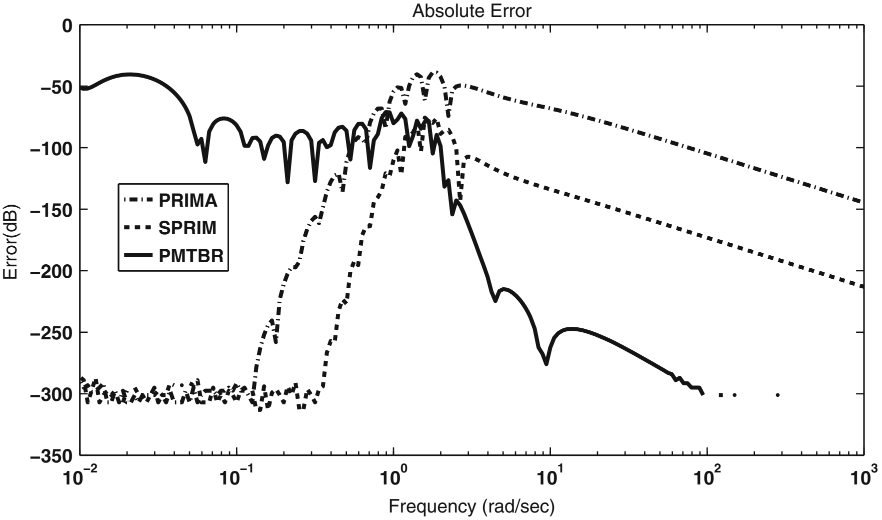


Fig. 4.7 Error plot, the frequency domain parameter ω varies between 10^{-2} to 10^3, Example 2

In the Example 1 we reduced the system from order $n = 401$ (number of nodes is $K = 201$) to order 34, which means that we reduced the system (in all three methods) by a factor of 10. The order of the reduced model is relatively large in this case as the dynamical system is somehow stubborn for any reductions, see Fig. 4.5. The price we will pay for a smaller system is too high as we loose a lot of information during the reduction and the error is becoming relatively large. As we expected, PRIMA and SPRIM in Fig. 4.6 produced reliable results close to the expansion point, in this case $s = 0$, but the error is immediately increasing for the rest of the oscillation part, see Fig. 4.2, and then smoothly decreases for higher frequencies. In the first example the PMTBR method matches a bit worse for the low frequencies as the error decreases just for a short interval and immediately starts to increase again. But PMTBR also cannot cover the oscillation part of the transfer function although it resolves the higher frequencies well. The order in PMTBR results from a prescribed tolerance.

For the second example the SPRIM and PRIMA produced a nice match around the expansion point, $s = 0$, like the first example, but for a larger interval, see Fig. 4.7. The peaks of error for both PRIMA and SPRIM are around -50 and -80 dB, respectively, which are much lower than in Example 1 where the peaks are around 0 dB for both PRIMA and SPRIM. We allowed PMTBR to reduce the system by a factor of 20 in this case although we keep the order of the reduced system the same as for the first example for the PRIMA and SPRIM. Despite the lower dimension for the reduced system PMTBR produced much better results for this test case compared to the first example as the error starts from -50 dB and smoothly decreases for low frequencies and suddenly falls to -300 dB for larger frequencies.

As we expected, the SPRIM produces a better approximation than PRIMA, especially for the second example, since it matches twice as much moments. Although both methods have a good agreement around the expansion point $s = 0$, the error increases as we are far from the expansion point. Since the Hankel singular values for the first example do not decay, the PMTBR method cannot produce an accurate model for low frequencies in that case. In the second example where the Hankel singular values rapidly decay PMTBR produced a more reliable result with a better match. This shows that we cannot stick to one method for reduction in general and the method should be chosen depending on the circuit's behavior.

4.1.8 *Summary*

In industrial applications of different disciplines, model order reduction is gaining more and more interest. As there is not the one and only type of model to describe all kinds of dynamics of different physical problems there is not and will never be the one and only MOR technique that fits best to all problems. Hence, research on MOR techniques is an ongoing process.

In the following contributions in this chapter you will find different approaches to different questions, aiming to attack different facets of reduced order models. This introductory contribution was ment to give an overview of the basic ideas and the motivation of some MOR techniques that are applied and refined throughout this chapter.

4.2 Eigenvalue Methods and Model Order Reduction

Physical structures and processes are modeled by dynamical systems in a wide range of application areas.[13] The increasing demand for complex components and large structures, together with an increasing demand for detail and accuracy, makes the models larger and more complicated. To be able to simulate these large-scale systems, there is need for reduced-order models of much smaller size, that approximate the behavior of the original model and preserve the important characteristics.

In order to preserve the important characteristics, one usually first has to know *what* are the important characteristics. For linear dynamical systems, two important characteristics are the dominant dynamics and stability. The dominant dynamics are determined by the dominant modes of the system, while stability of the system is determined by the location of the eigenvalues. Hence, both characteristics can be computed by solving eigenvalue problems: the dominant dynamics can be found by computing the dominant eigenvalues (poles) and corresponding eigenvectors, while stability can be assessed by determining whether the system has no eigenvalues in the right half-plane (the system is stable if there are no eigenvalues with real part greater than zero).

A large-scale dynamical system can have a large number of modes. Like a general square matrix can be approximated by its largest eigenvalues, i.e. by projecting it onto the space spanned by the eigenvectors corresponding to the largest eigenvalues, a dynamical system can be approximated by its dominant modes: a reduced order model, called the *modal equivalent*, can be obtained by projecting the state space on the subspace spanned by the dominant eigenvectors. This technique, *modal approximation* or *modal model reduction*, has been successfully applied to scalar and multivariable transfer functions of large-scale power systems, with applications such as stability analysis and controller design, see [81, 82].

The dominant eigenvectors, and the corresponding dominant poles of the system transfer function, are specific eigenvectors and eigenvalues of the state matrix. Because the systems are very large in practice, it is not feasible to compute all eigenvectors and to select the dominant ones.

Section 4.2 is concerned with the efficient computation of the dominant poles and eigenvectors specifically, and their use in model order reduction. The section is organized as follows. In Sect. 4.2.1 the concept of dominant poles and modal approximation is explained in more detail. Dominant poles can be computed with specialized eigensolution methods, as is described in Sect. 4.2.2. Some generalizations of the presented algorithms are shown in Sect. 4.2.3. Ideas on how to improve Krylov based MOR methods by using dominant poles are discussed in Sect. 4.2.4. Numerical examples are presented in Sect. 4.2.5. Section 4.2.6 concludes.

[13] Section 4.2 has been written by: Joost Rommes and Nelson Martins.

For general introductions to model order reduction we refer to the previous Sect. 4.1 and to [58, 60, 61, 88]; for eigenvalue problems, see [87, 93]. More detailed publications on the contents of this section are [80–85]. The pseudocode algorithms presented in this section are written using Matlab-like [92] notation.

4.2.1 Transfer Functions, Dominant Poles and Modal Equivalents

In Sect. 4.2, the dynamical systems $(E, A, \mathbf{b}, \mathbf{c}, d)$ are of the form

$$\begin{cases} E\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \mathbf{b}u(t) \\ y(t) \quad = \mathbf{c}^*\mathbf{x}(t) + du(t), \end{cases} \tag{4.29}$$

where $A, E \in \mathbb{R}^{n \times n}$, E may be singular, $\mathbf{b}, \mathbf{c}, \mathbf{x}(t) \in \mathbb{R}^n$, $u(t), y(t), d \in \mathbb{R}$. The vectors $\mathbf{b}$ and $\mathbf{c}$ are called the input, and output map, respectively. The transfer function $H : \mathbb{C} \to \mathbb{C}$ of (4.29) is defined as

$$H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d. \tag{4.30}$$

The poles of the transfer function in (4.30) are a subset of the eigenvalues $\lambda_i \in \mathbb{C}$ of the matrix pencil (A, E). An eigentriplet $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$ is composed of an eigenvalue λ_i of (A, E) and corresponding right and left eigenvectors $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{C}^n$:

$$\begin{aligned} A\mathbf{x}_i &= \lambda_i E\mathbf{x}_i, \qquad \mathbf{x}_i \neq 0, \\ \mathbf{y}_i^* A &= \lambda_i \mathbf{y}_i^* E, \qquad \mathbf{y}_i \neq 0, \qquad (i = 1, \ldots, n). \end{aligned}$$

The actual occurring poles in (4.30) are identified by the components of the eigenvectors in in $\mathbf{b}$ and $\mathbf{c}$. Assuming that the pencil is nondefective, the right and left eigenvectors corresponding to the same finite eigenvalue can be scaled so that $\mathbf{y}_i^* E\mathbf{x}_i = 1$. Furthermore, it is well known that left and right eigenvectors corresponding to distinct eigenvalues are E-orthogonal: $\mathbf{y}_i^* E\mathbf{x}_j = 0$ for $i \neq j$. The transfer function $H(s)$ can be expressed as a sum of residues $R_i \in \mathbb{C}$ over the $\tilde{n} \leq n$ finite first order poles [68]:

$$H(s) = \sum_{i=1}^{\tilde{n}} \frac{R_i}{s - \lambda_i} + R_\infty + d, \tag{4.31}$$

where the residues R_i are

$$R_i = (\mathbf{c}^*\mathbf{x}_i)(\mathbf{y}_i^*\mathbf{b}),$$

and R_∞ is the constant contribution of the poles at infinity (often zero).

Although there are different indices of modal dominance [57, 64, 94], the following will be used in this chapter.

Definition 4.1 A pole λ_i of $H(s)$ with corresponding right and left eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ ($\mathbf{y}_i^* E\mathbf{x}_i = 1$) is called the dominant pole if $|R_i|/|\mathrm{Re}(\lambda_i)| > |R_j|/|\mathrm{Re}(\lambda_j)|$, for all $j \neq i$.

More generally, a pole λ_i is called dominant if $|R_i|/|\mathrm{Re}(\lambda_i)|$ is not small compared to $|R_j|/|\mathrm{Re}(\lambda_j)|$, for all $j \neq i$. A dominant pole is well observable and controllable in the transfer function. This can also be seen in the corresponding Bode-plot, which is a plot of the magnitude $|H(i\omega)|$ against $\omega \in \mathbb{R}$: peaks occur at frequencies ω close to the imaginary parts of the dominant poles of $H(s)$. In practise one also plots the corresponding phase of $H(i\omega)$. An approximation of $H(s)$ that consists of $k < n$ terms with $|R_j|/|\mathrm{Re}(\lambda_j)|$ above some value, determines the effective transfer function behavior [90] and is also known as transfer function modal equivalent:

Definition 4.2 A transfer function modal equivalent $H_k(s)$ is an approximation of a transfer function $H(s)$ that consists of $k < n$ terms:

$$H_k(s) = \sum_{j=1}^{k} \frac{R_j}{s - \lambda_j} + d. \tag{4.32}$$

A modal equivalent that consists of the most dominant terms determines the effective transfer function behavior [90]. If $X \in \mathbb{C}^{n\times k}$ and $Y \in \mathbb{C}^{n\times k}$ are matrices having the left and right eigenvectors $\mathbf{y}_i$ and $\mathbf{x}_i$ of (A, E) as columns, such that $Y^*AX = \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$, with $Y^*EX = I$, then the corresponding (complex) reduced system follows by setting $\mathbf{x} = X\tilde{\mathbf{x}}$ and multiplying from the left by Y^*:

$$\begin{cases} \dot{\tilde{\mathbf{x}}}(t) = \Lambda\tilde{\mathbf{x}}(t) + (Y^*\mathbf{b})u(t) \\ \tilde{y}(t) = (\mathbf{c}^*X)\tilde{\mathbf{x}}(t) + du(t). \end{cases}$$

In practice, it is advisable to make a real reduced model in the following way: for every complex pole triplet $(\lambda, \mathbf{x}, \mathbf{y})$, construct real bases for the right and left eigenspace via $[\mathrm{Re}(\mathbf{x}), \mathrm{Im}(\mathbf{x})]$ and $[\mathrm{Re}(\mathbf{y}), \mathrm{Im}(\mathbf{y})]$, respectively. Let the columns of X_r and Y_r be such bases, respectively. Because the complex conjugate eigenvectors are also in this space, the real bases for the eigenspaces are still (at most) k dimensional. The real reduced model can be formed by using X_r and Y_r in $(Y_r^*EX_r, Y_r^*AX_r, Y_r^*\mathbf{b}, X_r^*\mathbf{c}, d)$.

For stable nondefective systems, the error in the modal equivalent can be quantified as [64]

$$\begin{aligned}\|H - H_k\|_\infty &= \| \sum_{j=k+1}^{n} \frac{R_j}{s-\lambda_j} \|_\infty \\ &\le \sum_{j=k+1}^{n} \frac{|R_j|}{|\mathrm{Re}(\lambda_j)|},\end{aligned}$$

where $\|H\|_\infty$ is the operator norm induced by the 2-norm in the frequency domain [58, 64]. An advantage of modal approximation is that the poles of the modal equivalent are also poles of the original system.

The dominant poles are specific (complex) eigenvalues of the pencil (A, E) and usually form a small subset of the spectrum of (A, E), so that rather accurate modal equivalents may be possible for $k \ll n$. Since the dominant poles can be located anywhere in the spectrum, specialized eigensolution methods are needed. Because the dominance of a pole is independent of d, without loss of generality $d = 0$ in the following.

4.2.2 *Specialized Eigensolution Methods*

In this section we describe the Dominant Pole Algorithm and its extension with deflation and subspace acceleration.

4.2.2.1 The Dominant Pole Algorithm (DPA)

The poles of the transfer function (4.30) are the $\lambda \in \mathbb{C}$ for which $\lim_{s\to\lambda} |H(s)| = \infty$ and can be computed via the roots of $G(s) = 1/H(s)$. Applying Newton's method leads to the scheme

$$s_{k+1} = s_k - \frac{\mathbf{c}^*\mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}, \tag{4.33}$$

where $\mathbf{v}_k = (s_k E - A)^{-1}\mathbf{b}$ and $\mathbf{w}_k = (s_k E - A)^{-*}\mathbf{c}$. The algorithm based on this scheme, also known as the Dominant Pole Algorithm (DPA) [72], is shown in Algorithm 4.1. Note that strictly speaking the definition of dominance used here is based on $|R_j|$ (and not on $|R_j|/|\mathrm{Re}(\lambda_j)|$ as in Definition 4.1); observe that in (4.32) $R_j = (\mathbf{c}^*\mathbf{x}_j)(\mathbf{y}_j^*\mathbf{b})$. The subsequent algorithms offer refinements that may lead to additional candidates, in any user-specified dominance criterion, including Definition 4.1.

Algorithm 4.1 The Dominant Pole Algorithm (DPA)

```
INPUT: System $(E, A, \mathbf{b}, \mathbf{c})$, initial pole estimate $s_0$, tolerance $\epsilon \ll 1$
OUTPUT: Approximate dominant pole $\lambda$ (close to $s_0$) and corresponding right and left eigenvectors $\mathbf{x}$ and $\mathbf{y}$
1: Set $k = 0$
2: while not converged do
3:     Solve $\mathbf{v}_k \in \mathbb{C}^n$ from $(s_k E - A)\mathbf{v}_k = \mathbf{b}$
4:     Solve $\mathbf{w}_k \in \mathbb{C}^n$ from $(s_k E - A)^* \mathbf{w}_k = \mathbf{c}$
5:     Compute the new pole estimate
           $s_{k+1} = s_k - \frac{\mathbf{c}^* \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k} = \frac{\mathbf{w}_k^* A \mathbf{v}_k}{\mathbf{w}_k^* E \mathbf{v}_k}$
6:     The pole $\lambda = s_{k+1}$ with $\mathbf{x} = \mathbf{v}_k$ and $\mathbf{y} = \mathbf{w}_k$ has converged if
           $\|A\mathbf{v}_k - s_{k+1} E \mathbf{v}_k\|_2 < \epsilon$
7:     Set $k = k + 1$
8: end while
```

The Dominant Pole Algorithm is closely related to Rayleigh Quotient Iteration [76, 77]: the only difference is that in DPA the right hand-sides in Step 3 and 4 remain fixed, while in Rayleigh Quotient Iterations these are updated with $\mathbf{b} = E\mathbf{v}_{k-1}$ and $\mathbf{c} = E^* \mathbf{w}_{k-1}$ every iteration. See [85] for a detailed comparison.

The two linear systems that need to be solved in step 3 and 4 of Algorithm 4.1 to compute the Newton update in (4.33) can be efficiently solved using one LU-factorization $LU = s_k E - A$, by noting that $U^* L^* = (s_k E - A)^*$. If an exact LU-factorization is not available, one has to use inexact Newton schemes, such as inexact Rayleigh Quotient Iteration and Jacobi-Davidson style methods [67, 89, 91]. In the next section, extensions of DPA are presented that are able to compute more than one eigenvalue in an effective and efficient way.

4.2.2.2 Deflation and Subspace Acceleration

In practical applications often more than one dominant pole is wanted: one is interested in all the dominant poles, no matter what definition of dominance is used. Simply running the single pole algorithm DPA for a number of different initial shifts will most likely result in duplicate dominant poles. A well known strategy to avoid computation of already found eigenvalues is deflation, see for instance [87]. It is also known that subspace acceleration may improve the global convergence: for an $n \times n$ problem, the subspace accelerated algorithm converges within at most n iterations, although in practice it may need only $k \ll n$ iterations and will almost never build a full search space of dimension n, but restart every $k_{max} \ll n$ iterations. The use of subspace acceleration requires that every iteration an approximate pole has to be selected from the available approximations. This also may improve the global convergence, since better approximations than the initial estimate, which may be a rather crude approximation, become available during the process.

In the next subsections, variants of DPA for the computation of more than one pole without and with subspace acceleration are discussed. This variant that does not use subspace acceleration can be implemented efficiently with only constant costs for deflation, while the subspace accelerated variant has better global convergence.

Throughout the rest of this chapter, let the $(n \times k)$ matrices X_k and Y_k have as their columns the normalized (found) right and left eigenvectors $\mathbf{x}_i$ and $\mathbf{y}_i$ $(i = 1, \dots, k)$ of (A, E), respectively, and let Λ_k be a diagonal $(k \times k)$ matrix with the corresponding eigenvalues on its diagonal, i.e. $\Lambda_k = \operatorname{diag}(\lambda_1, \dots, \lambda_k)$, $Y_k^* A X_k = \Lambda_k$ and $Y_k^* E X_k = I$. For ease of notation, the subscript k will be omitted if this does not lead to confusion.

4.2.2.3 DPA with Deflation by Restriction

It can be verified that if $X \equiv X_k$ and $Y \equiv Y_k$ have as their columns exact eigenvectors (normalized so that $Y^*EX = I$), then the system $(E_d, A_d, \mathbf{b}_d, \mathbf{c}_d)$, where

$$\begin{aligned} E_d &= (I - EXY^*)E(I - XY^*E), \\ A_d &= (I - EXY^*)A(I - XY^*E), \\ \mathbf{b}_d &= (I - EXY^*)\mathbf{b}, \\ \mathbf{c}_d &= (I - E^*YX^*)\mathbf{c}, \end{aligned}$$

has the same poles, eigenvectors and residues, but with the found λ_i $(i = 1, \dots, k)$ and corresponding R_i transformed to 0. So in order to avoid recomputing the same pole, DPA could be applied to the deflated system $(E_d, A_d, \mathbf{b}_d, \mathbf{c}_d)$ after having found one or more poles. This would require solves with $(sE_d - A_d)$ and $(sE_d - A_d)^*$ in step 4 and 5 of Algorithm 4.2,[14] but the following theorem shows that it is sufficient to only replace $\mathbf{b}$ by $\mathbf{b}_d$ and $\mathbf{c}$ by $\mathbf{c}_d$ to ensure deflation.

Theorem 4.2 ([80, Thm. 3.3.1]) *The deflated transfer function $H_d(s) = \mathbf{c}_d^*(sE - A)^{-1}\mathbf{b}_d$, where*

$$\mathbf{b}_d = (I - EXY^*)\mathbf{b} \quad \textit{and} \quad \mathbf{c}_d = (I - E^*YX^*)\mathbf{c},$$

has the same poles λ_i and corresponding residues R_i as $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$, but with the residues R_i corresponding to the found poles λ_i $(i = 1, \dots, k)$ transformed to $R_i = 0$.

[14] Note that $(sE_d - A_d)$ would never be computed explicitly, and that sparse systems $(sE_d - A_d)\mathbf{x} = \mathbf{b}_d$ can be solved efficiently.

Algorithm 4.2 Dominant Pole Algorithm with deflation (DPAd)

```
INPUT: System $(E, A, \mathbf{b}, \mathbf{c})$, initial pole estimates $s_0^1, \ldots, s_0^p$, tolerance $\epsilon \ll 1$
OUTPUT: Approximate dominant poles $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$, and corresponding right and left
    eigenvectors $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p]$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_p]$
Set $k = 0, i = 0, s_k = s_0^1$
while $i < p$ do
                                        ▷ Continue until $p$ poles have been found
    Solve $\mathbf{v}_k \in \mathbb{C}^n$ from $(s_k E - A)\mathbf{v}_k = \mathbf{b}$
    Solve $\mathbf{w}_k \in \mathbb{C}^n$ from $(s_k E - A)^*\mathbf{w}_k = \mathbf{c}$
    Compute the new pole estimate
        $$s_{k+1} = s_k - \frac{\mathbf{c}^*\mathbf{v}_k}{\mathbf{w}_k^* E\mathbf{v}_k} = \frac{\mathbf{w}_k^* A\mathbf{v}_k}{\mathbf{w}_k^* E\mathbf{v}_k}$$
    if $\|A\mathbf{v}_k - s_{k+1}E\mathbf{v}_k\|_2 < \epsilon$ (with $\|\mathbf{v}_k\|_2 = 1$) then
        Set $i = i + 1$
        Set $\lambda_{ii} = s_{k+1}$
        Set $\mathbf{v}_k = \mathbf{v}_k/(\mathbf{w}_k^* E\mathbf{v}_k)$
        Set $X = [X, \mathbf{v}_k]$ and $Y = [Y, \mathbf{w}_k]$
        Deflate: $\mathbf{b} = \mathbf{b} - E\mathbf{v}_k\mathbf{w}_k^*\mathbf{b}$
        Deflate: $\mathbf{c} = \mathbf{c} - E^*\mathbf{w}_k\mathbf{v}_k^*\mathbf{c}$
        Set $s_{k+1} = s_0^i$
    end if
    Set $k = k + 1$
end while
```

Proof The proof follows by using the definition of residues and basic linear algebra [80, Thm. 3.3.1]. □

In other words, by using $\mathbf{b}_d$ and $\mathbf{c}_d$ the found dominant poles are degraded to non dominant poles of $H_d(s)$, while not changing the dominance of the remaining poles. Hence these poles will not be recomputed by DPA applied to $H_d(s)$. Graphically, the peaks caused by the found poles are 'flattened' in the Bode plot (see also Fig. 4.8).

Note that if $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d$ with $d = 0$, then the deflated poles in fact become zeros of $H_d(s)$. It can be shown that DPA applied to $H_d(s) = \mathbf{c}_d^*(sE - A)^{-1}\mathbf{b}_d$ and DPA applied to $H_{\tilde{d}}(s) = \mathbf{c}_d^*(sE_d - A_d)^{-1}\mathbf{b}_d$ produce the same results [85].

The important result is that the single pole DPA can easily be extended, see Algorithm 4.2, to an algorithm that is able to compute more than one pole, while maintaining constant costs per iteration, except for iterations in which a pole is found. The only change to be made to Algorithm 4.1, is when a dominant pole triplet $(\lambda, \mathbf{x}, \mathbf{y})$ is found: in that case, the algorithm continues with $\mathbf{b}$ and $\mathbf{c}$ replaced by $(I - E\mathbf{x}\mathbf{y}^*)\mathbf{b}$ and $(I - E^*\mathbf{y}\mathbf{x}^*)\mathbf{c}$, respectively.

This approach has a number of advantages. The implementation is straightforward and efficient: search spaces, selection strategies and orthogonalization procedures are not needed, so that the computational costs per iteration remain constant, even if the number of found poles increases. For every found pole only two skew projections are needed once to compute the new $\mathbf{b}_d$ and $\mathbf{c}_d$, so the costs

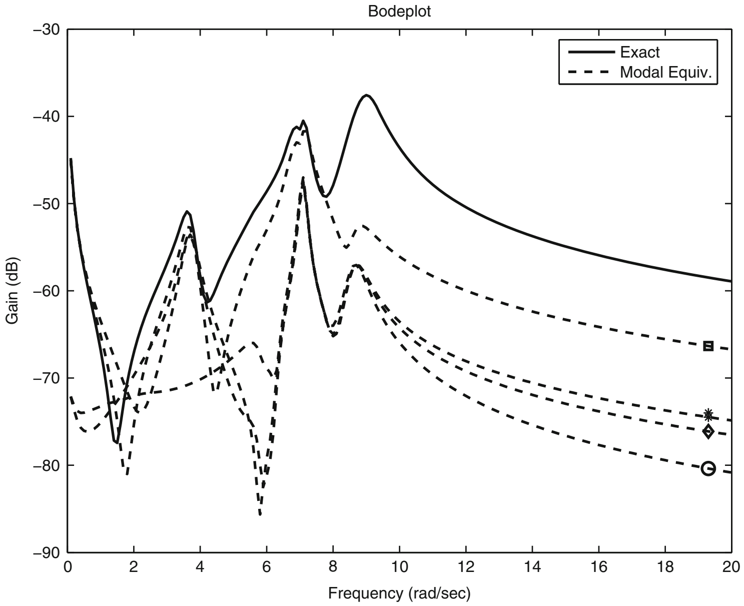


Fig. 4.8 Exact transfer function (*solid*) of the New England test system [72], and modal equivalents where the following dominant pole (*pairs*) are removed one by one: $-0.467 \pm 8.96i$ (*square*), $-0.297 \pm 6.96i$ (*asterisk*), -0.0649 (*diamond*), and $-0.249 \pm 3.69i$ (*circle*). Note that the corresponding peaks are removed from the Bode plot as well (to see this, check the Bode plot at the frequencies near the imaginary part of the removed pole)

for deflation are constant. The pseudo code in Algorithm 4.2 can almost literally be used as Matlab code. The special properties of DPA ensure convergence to dominant poles (locally). Furthermore, the deflation of found poles is numerically stable in the sense that even if the corresponding transformed residues are not exactly zero, which is usually the case in finite arithmetic, this will hardly influence the effect of deflation: firstly, all the poles are left unchanged, and secondly, already a decrease of dominance of the found poles to nondominance (because of the projected in- and output vectors $\mathbf{b}_d$ and $\mathbf{c}_d$) will shrink the local convergence neighborhood of these poles significantly, again because of the convergence behavior of DPA [85].

This approach, however, may still suffer from the fact that the convergence behavior can be very local and hence may heavily depend on the initial estimates s_0^i. Although in practice one often has rather accurate initial estimates of the poles of interest, this may be problematic if accurate information is not available. It may take many iterations until convergence if the initial estimate is not in the neighborhood of a dominant pole. On the other hand, the computational complexity of this problem depends on the costs of the LU factorization, which in certain practical examples can be computed very efficiently. In the next section a subspace accelerated version of DPA is described, that improves the global convergence by using search spaces.

Algorithm 4.3 Subspace Accelerated DPA (SADPA)

```
INPUT: System $(E, A, \mathbf{b}, \mathbf{c})$, initial pole estimate $s_1$ and the number of wanted poles $p$
OUTPUT: Dominant pole triplets $(\lambda_i, \mathbf{x}_i, \mathbf{y}_i)$, $i = 1, \ldots, p$
 1: $k = 1$, $p_{found} = 0$, $\Lambda = [\,]$, $X = Y = [\,]$
 2: while $p_{found} < p$ do
 3:     Solve $\mathbf{v}$ from $(s_k E - A)\mathbf{v} = \mathbf{b}$
 4:     Solve $\mathbf{w}$ from $(s_k E - A)^*\mathbf{w} = \mathbf{c}$
 5:     $\mathbf{v} = \mathrm{MGS}(V, \mathbf{v})$, $V = [V, \mathbf{v}/||\mathbf{v}||_2]$
 6:     $\mathbf{w} = \mathrm{MGS}(W, \mathbf{w})$, $W = [W, \mathbf{w}/||\mathbf{w}||_2]$
 7:     Compute $S = W^*AV$ and $T = W^*EV$
 8:     $(\tilde{\Lambda}, \tilde{X}, \tilde{Y}) = \mathrm{Sort}(S, T, W^*\mathbf{b}, V^*\mathbf{c})$        ▷ Algorithm 4.4
 9:     Dominant approximate eigentriplet of $(A, E)$ is

        $$(\hat{\lambda}_1 = \tilde{\lambda}_1, \hat{\mathbf{x}}_1 = V\tilde{\mathbf{x}}_1/\|V\tilde{\mathbf{x}}_1\|_2, \hat{\mathbf{y}}_1 = W\tilde{\mathbf{y}}_1/\|W\tilde{\mathbf{y}}_1\|_2)$$

10:     if $||A\hat{\mathbf{x}}_1 - \hat{\lambda}_1 E\hat{\mathbf{x}}_1||_2 < \epsilon$ then
11:         $(\Lambda, X, Y, V, W, \mathbf{b}, \mathbf{c}) =$
12:       Deflate$(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1, \Lambda, X, Y, V\tilde{X}_{2:k}, W\tilde{Y}_{2:k}, E, \mathbf{b}, \mathbf{c})$        ▷ Algorithm 4.5
13:         $p_{found} = p_{found} + 1$
14:         Set $\tilde{\lambda}_1 = \tilde{\lambda}_2$, $k = k - 1$
15:     end if
16:     Set $k = k + 1$
17:     Set the new pole estimate $s_k = \tilde{\lambda}_1$
18: end while
```

4.2.2.4 Subspace Accelerated DPA

A drawback of DPA is that information obtained in the current iteration is discarded at the end of the iteration. The only information that is preserved is contained in the new pole estimate s_{k+1}. The current right and left approximate eigenvectors $\mathbf{v}_k$ and $\mathbf{w}_k$, however, may also contain components in the direction of eigenvectors corresponding to other dominant poles. Instead of discarding these approximate eigenvectors, they are kept in search spaces spanned by the columns of V and W, respectively. This idea is known as subspace acceleration.

A global overview of SADPA is shown in Algorithm 4.3. Starting with a single shift s_1, the first iteration is equivalent to the first iteration of the DPA (step 3–4). The right and left eigenvector approximations $\mathbf{v}_1$ and $\mathbf{w}_1$ are kept in spaces V and W. In the next iteration, these spaces are expanded orthogonally, by using modified Gram-Schmidt (MGS) [63], with the approximations $\mathbf{v}_2$ and $\mathbf{w}_2$ corresponding to the new shift s_2 (step 5–6). Hence the spaces grow and will contain better approximations.

It can be shown that subspace accelerated DPA, under certain conditions, is equivalent to subspace accelerated Rayleigh Quotient Iteration and the Jacobi-Davidson method, see [80, 85] for more details.

Algorithm 4.4 $(\tilde{\Lambda}, \tilde{X}, \tilde{Y}) = \text{Sort}(S, T, \mathbf{b}, \mathbf{c})$

```
INPUT: $S, T \in \mathbb{C}^{k\times k}$, $\mathbf{b}, \mathbf{c} \in \mathbb{C}^k$
OUTPUT: $\tilde{\Lambda} \in \mathbb{C}^k$, $\tilde{X}, \tilde{Y} \in \mathbb{C}^{k\times k}$ with $\lambda_1$ the pole with largest (scaled) residue magnitude and
    $\tilde{\mathbf{y}}_1$ and $\tilde{\mathbf{x}}_1$ the corresponding right and left eigenvectors.
1: Compute eigentriplets of the pair $(S, T)$:
       $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i), \quad \tilde{\mathbf{y}}_i^* T \tilde{\mathbf{x}}_i = 1, \quad i = 1, \dots, k$
2: $\tilde{\Lambda} = [\tilde{\lambda}_1, \dots, \tilde{\lambda}_k]$
3: $\tilde{X} = [\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_k]$
4: $\tilde{Y} = [\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_k]$
5: Compute residues $R_i = (\mathbf{c}^* \tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^* \mathbf{b})$
6: Sort $\tilde{\Lambda}$, $\tilde{X}$, $\tilde{Y}$ in decreasing $|R_i|/|\text{Re}(\tilde{\lambda}_i)|$ order
```

Selection Strategy

In iteration k the Petrov-Galerkin approach leads (step 7) to the projected eigenproblem

$$W^*AV\tilde{\mathbf{x}} = \tilde{\lambda} W^*EV\tilde{\mathbf{x}},$$
$$\tilde{\mathbf{y}}W^*AV = \tilde{\lambda}\tilde{\mathbf{y}}W^*EV.$$

Since the interaction matrices $S = W^*AV$ and $T = W^*EV$ are of low dimension $k \ll n$, the eigentriplets $(\tilde{\lambda}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ of this reduced problem can be computed using the QZ method (or the QR method in the bi-E-orthogonal case) (step 1 of Algorithm 4.4). This provides k approximate eigentriplets $(\hat{\lambda}_i = \tilde{\lambda}_i, \hat{\mathbf{x}}_i = V\tilde{\mathbf{x}}_i, \hat{\mathbf{y}}_i = W\tilde{\mathbf{y}}_i)$ for (A, E). The most natural thing to do is to choose the triplet $(\hat{\lambda}_j, \hat{\mathbf{x}}_j, \hat{\mathbf{y}}_j)$ with the most dominant pole approximation (step 8–9): compute the corresponding residues $\hat{R}_i = (\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b})$ of the k pairs and select the pole with the largest $|\hat{R}_j|/|\text{Re}(\hat{\lambda}_j)|$ (see Algorithm 4.4). The SADPA then continues with the new shift $s_{k+1} = \hat{\lambda}_j$ (step 16).

The residues $\hat{R}_i$ can be computed without computing the approximate eigenvectors explicitly (step 5 of Algorithm 4.4): if the $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ are scaled so that $\tilde{\mathbf{y}}_i^* T \tilde{\mathbf{x}}_i = 1$ $(= \hat{\mathbf{y}}_i^* E \hat{\mathbf{x}}_i)$, then it follows that the $\hat{R}_i$ can be computed as $\hat{R}_i = ((\mathbf{c}^*V)\tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^*(W^*\mathbf{b}))$ $(= (\mathbf{c}^*\hat{\mathbf{x}}_i)(\hat{\mathbf{y}}_i^*\mathbf{b}))$.

Instead of $\hat{\mathbf{y}}_i^* E \hat{\mathbf{x}}_i = 1$ one can also use the scaling $\|\hat{\mathbf{y}}_i\|_2 = \|\hat{\mathbf{x}}_i\|_2 = 1$ when computing approximate residues. In that case the product of the angles $\angle(\hat{\mathbf{x}}_i, \mathbf{c})$ and $\angle(\hat{\mathbf{y}}_i, \mathbf{b})$ is used in the computation of the approximate residues (see also [85]), which numerically may be more robust.

Algorithm 4.5 $(\Lambda, X, Y, \tilde{V}, \tilde{W}, \mathbf{b}_d, \mathbf{c}_d) = \text{Deflate}(\lambda, \mathbf{x}, \mathbf{y}, \Lambda, X, Y, V, W, E, \mathbf{b}, \mathbf{c})$

```
INPUT: $\lambda \in \mathbb{C}, \mathbf{x}, \mathbf{y} \in \mathbb{C}^n, \Lambda \in \mathbb{C}^p, X, Y \in \mathbb{C}^{n \times p}, V, W \in \mathbb{C}^{n \times k}, E \in \mathbb{C}^{n \times n}, \mathbf{b}, \mathbf{c} \in \mathbb{C}^n$
OUTPUT: $\Lambda \in \mathbb{C}^q, X, Y \in \mathbb{C}^{n \times q}, \tilde{V}, \tilde{W} \in \mathbb{C}^{n \times k-1}, \mathbf{b}_d, \mathbf{c}_d \in \mathbb{C}^n$, where $q = p + 1$ if $\lambda$ has zero
    imaginary part and $q = p + 2$ if $\lambda$ has nonzero imaginary part.
 1: $\Lambda = [\Lambda, \lambda]$
 2: Set $\mathbf{x} = \mathbf{x}/(\mathbf{y}^* E\mathbf{x})$
 3: $X = [X, \mathbf{x}]$
 4: $Y = [Y, \mathbf{y}]$
 5: Deflate: $\mathbf{b}_d = \mathbf{b} - E\mathbf{x}(\mathbf{y}^*\mathbf{b})$
 6: Deflate: $\mathbf{c}_d = \mathbf{c} - E^*\mathbf{y}(\mathbf{x}^*\mathbf{c})$
 7: if imag($\lambda$) $\neq 0$ then
 8:                                              ▷ Also deflate complex conjugate
 9:     $\Lambda = [\Lambda, \bar{\lambda}]$
10:     $\mathbf{x} = \bar{\mathbf{x}}$, $X = [X, \mathbf{x}]$
11:     $\mathbf{y} = \bar{\mathbf{y}}$, $Y = [Y, \mathbf{y}]$
12:     Deflate: $\mathbf{b}_d = \mathbf{b}_d - E\mathbf{x}(\mathbf{y}^*\mathbf{b}_d)$
13:     Deflate: $\mathbf{c}_d = \mathbf{c}_d - E^*\mathbf{y}(\mathbf{x}^*\mathbf{c}_d)$
14: end if
15: $\tilde{V} = \tilde{W} = [\,]$
16: for $j = 1, \ldots, k$ do
17:     $\tilde{V} = \text{Expand}(\tilde{V}, X, Y, E, \mathbf{v}_j)$          ▷ Algorithm 4.6
18:     $\tilde{W} = \text{Expand}(\tilde{W}, Y, X, E^*, \mathbf{w}_j)$        ▷ Algorithm 4.6
19: end for
```

Deflation

In each iteration step a convergence test (step 10) is done like in DPAd (Algorithm 4.2): if for the selected eigentriplet $(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{y}}_1)$ the norm of the residual $||A\hat{\mathbf{x}}_1 - \hat{\lambda}_1 E\hat{\mathbf{x}}_1||_2$ is smaller than some tolerance ϵ, it is converged. In general more than one dominant eigentriplet is wanted and it is desirable to avoid repeated computation of the same eigentriplet. The same deflation technique as used in DPAd can be applied here (steps 5–6 and 12–13 of Algorithm 4.5, see also Sect. 4.2.2.3), and since SADPA continues with $\mathbf{b}_d$ and $\mathbf{c}_d$, no explicit E-orthogonalization of expansion vectors against found eigenvectors is needed in step 3 and 4. The effect is similar to the usual deflation in Jacobi-Davidson methods [62]: found eigenvectors are hard-locked, i.e. once deflated, they do not participate and do not improve during the rest of the process (contrary to soft-locking, where deflated eigenvectors still participate in the Rayleigh-Ritz (Ritz-Galerkin) procedure and may be improved, at the cost of additional computations and administration, see [69, 70]). In fact, there is cheap explicit deflation without the need for implicit deflation (cf. [62, remark 5, p. 106], where a combination of explicit and implicit deflation is used).

Algorithm 4.6 $V = \text{Expand}(V, X, Y, E, \mathbf{v})$

```
INPUT: $V \in \mathbb{C}^{n\times k}$ with $V^*V = I$, $X, Y \in \mathbb{C}^{n\times p}$, $E \in \mathbb{C}^{n\times n}$, $\mathbf{v} \in \mathbb{C}^n$, $Y^*EX$ diagonal,
    $Y^*EV = 0$
OUTPUT: $V \in \mathbb{C}^{n\times(k+1)}$ with $V^*V = I$ and
  $\mathbf{v}_{k+1} = \prod_{j=1}^{p}(I - \frac{\mathbf{x}_j\mathbf{y}_j^*E}{\mathbf{y}_j^*E\mathbf{x}_j}) \cdot \mathbf{v}$
  $\mathbf{v} = \prod_{j=1}^{p}(I - \frac{\mathbf{x}_j\mathbf{y}_j^*E}{\mathbf{y}_j^*E\mathbf{x}_j}) \cdot \mathbf{v}$
  $\mathbf{v} = \text{MGS}(V, \mathbf{v})$
  $V = [V, \mathbf{v}/||\mathbf{v}||_2]$
```

If an eigentriplet has converged (steps 11–13 of Algorithm 4.3), the eigenvectors are deflated from the search spaces by reorthogonalizing the search spaces against the found eigenvectors. This can be done by using modified Gram-Schmidt (MGS) and by recalling that, if the exact vectors are found, the pencil

$$((I - EXY^*)A(I - XY^*E), \quad (I - EXY^*)E(I - XY^*E))$$

has the same eigentriplets as (A, E), but with the found eigenvalues transformed to zero (Algorithm 4.6, see also [62, 67]). Since in finite arithmetic only *approximations* to exact eigentriplets are available, the computed eigenvalues are transformed to $\eta \approx 0$. The possible numerical consequences of this, however, are limited, since SADPA continues with $\mathbf{b}_d$ and $\mathbf{c}_d$, and as argued in Sect. 4.2.2.3, the residues of the found poles are transformed to (approximately) zero.

If a complex pole has converged, its complex conjugate is also a pole and the corresponding complex conjugate right and left eigenvectors can also be deflated. A complex conjugate pair is counted as one pole. The complete deflation procedure is shown in Algorithm 4.5.

After deflation of the found pole(s), SADPA continues with the second most dominant approximate pole (steps 13–16 of Algorithm 4.3).

Further Improvements and Remarks

It may happen that the search spaces V and W become high-dimensional, especially when a large number of dominant poles is wanted. A common way to deal with this is to do a thick restart [62, 87]: if the subspaces V and W reach a certain maximum dimension $k_{max} \ll n$, they are reduced to a dimension $k_{min} < k_{max}$ by keeping the k_{min} most dominant approximate eigentriplets; the process is restarted with the reduced V and W (already converged eigentriplets are not part of the active subspaces V and W). This procedure is repeated until all poles are found.

Furthermore, as more eigentriplets have converged, approximations of new eigentriplets may become poorer or convergence may be hampered, due to rounding errors in the orthogonalization phase and the already converged eigentriplets. It is therefore advised to take a small tolerance $\epsilon \leq 10^{-10}$. Besides that, as the estimate converges to a dominant pole, the right and left eigenvectors computed in step 3 and 4 of Algorithm 4.3 are usually more accurate than the approximations computed in the selection procedure: if the estimate s_k is close to an eigenvalue λ, then $s_k E - A$ may become ill-conditioned, but, as is discussed in [79] and [78, Section 4.3], the solutions $\mathbf{v}_k$ and $\mathbf{w}_k$ have large components in the direction of the corresponding right and left eigenvectors (provided $\mathbf{b}$ and $\mathbf{c}$ have sufficiently large components in those directions). In the deflation phase, it is therefore advised to take the most accurate of both, i.e., the approximate eigenvector with smallest residual. It may also be advantageous to do an additional step of two-sided Rayleigh quotient iteration to improve the eigenvectors.

SADPA requires only one initial estimate. If rather accurate initial estimates are available, one can take advantage of this in SADPA by setting the next estimate after deflation to a new initial estimate (step 16 of Algorithm 4.3).

Every iteration, two linear systems are to be solved (step 3 and 4). As was already mentioned, this can efficiently be done by computing one LU-factorization and solving the systems by using L and U, and U^* and L^*, respectively. Because in practice the system matrices A and E are often very sparse and structured, computation of the LU-factorizations can be relatively inexpensive.

The selection criterion can easily be changed to another of the several existing indices of modal dominance [57, 64, 94]. Furthermore, the strategy can be restricted to considering only poles in a certain frequency range. Also, instead of providing the number of wanted poles, the procedure can be automated even further by providing the desired maximum error $|H(s) - H_k(s)|$ for a certain frequency range: the procedure continues computing new poles until the error bound is reached. Note that such an error bound requires that the transfer function of the complete model can be evaluated efficiently for the frequency range of interest.

A Numerical Example

For illustrational purposes, SADPA was applied to a transfer function of the New England test system, a model of a power system. This small benchmark system has 66 state variables (for more information, see [72]). The tolerance used was $\epsilon = 10^{-10}$ and no restarts were used. Every iteration, the pole approximation $\hat{\lambda}_j$ with largest $|\hat{R}_j|/|\mathrm{Re}(\hat{\lambda}_j)|$ was selected. Table 4.1 shows the found dominant poles and the iteration number for which the pole satisfied the stopping criterion. Bodeplots of two modal equivalents are shown in Fig. 4.9. The quality of the modal equivalent increases with the number of found poles, as can be observed from the better match of the exact and reduced transfer function.

Table 4.1 Results for SADPA applied to the New England test system ($s_1 = 1i$)

#Poles	#States	New pole	Iteration	Bodeplot
1	2	$-0.4672 \pm 8.9644i$	13	–
2	4	$-0.2968 \pm 6.9562i$	18	–
3	5	-0.0649	21	Fig. 4.9 (left)
4	7	$-0.2491 \pm 3.6862i$	25	–
5	9	$-0.1118 \pm 7.0950i$	26	–
6	11	$-0.3704 \pm 8.6111i$	27	Fig. 4.9 (right)

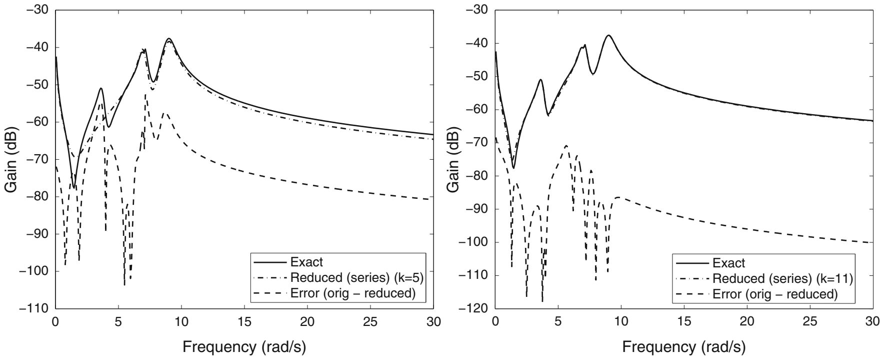


Fig. 4.9 Bode plot of 5th order (*left*) and 11th order (*right*) modal equivalent, complete model and error for the transfer function of the New England test system (66 states in the complete model)

4.2.3 Generalizations to Other Eigenvalue Problems

In this section, four variants of the dominant pole algorithm presented in the previous section are briefly discussed. In Sect. 4.2.3.1, the theory is extended to multi-input multi-output systems. A variant of DPA that computes the dominant zeros of a transfer function is described in Sect. 4.2.3.2. Section 4.2.3.3 describes the extension to higher-order dynamical systems. Finally, in Sect. 4.2.3.4 it is shown how DPA can be used for the computation of eigenvalues most sensitive to parameter changes.

4.2.3.1 MIMO Systems

For a multi-input multi-output (MIMO) system

$$\begin{cases} E\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t) \\ \mathbf{y}(t) \quad = C^*\mathbf{x}(t) + D\mathbf{u}(t), \end{cases}$$

where $A, E \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{n\times p}$, $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{y}(t) \in \mathbb{R}^p$ and $D \in \mathbb{R}^{p\times m}$, the transfer function $H(s) : \mathbb{C} \to \mathbb{C}^{p\times m}$ is defined as

$$H(s) = C^*(sE - A)^{-1}B + D. \tag{4.34}$$

The dominant poles of (4.34) are those $s \in \mathbb{C}$ for which the largest singular value $\sigma_{\max}(H(s)) \to \infty$. For square transfer functions ($m = p$), there is an equivalent criterion: the dominant poles are those $s \in \mathbb{C}$ for which the absolute smallest eigenvalue $|\lambda_{\min}(H^{-1}(s))| \to 0$. This leads, for square transfer functions, to the following Newton scheme:

$$s_{k+1} = s_k - \frac{1}{\mu_{\min}} \frac{1}{\mathbf{v}^* C^*(s_k E - A)^{-2} B\mathbf{u}},$$

where $(\mu_{\min}, \mathbf{u}, \mathbf{v})$ is the eigentriplet of $H^{-1}(s_k)$ corresponding to $\lambda_{\min}(H^{-1}(s_k))$. For a dominant pole, the mountain spreads of $\sigma_{\max}$ are larger and, therefore, the neighborhood of convergence attraction is larger than for a less dominant pole, which would show just a spike. An algorithm for computing the dominant poles of a MIMO transfer function can be readily derived from Algorithm 4.1. The reader is referred to [74] for the initial MIMO DPA algorithm and to [81] for an algorithm SAMDP, similar to SADPA, generalizations to non-square MIMO systems and more details.

4.2.3.2 Computing the Zeros of a Transfer Function

The zeros of a transfer function $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b} + d$ are those $s \in \mathbb{C}$ for which $H(s) = 0$. An algorithm, similar to Algorithm 4.1, can be derived by noting that a Newton scheme for computing the zeros of a transfer function is given by

$$s_{k+1} = s_k + \frac{\mathbf{c}^*(s_k E - A)^{-1}\mathbf{b} + d}{\mathbf{c}^*(s_k E - A)^{-2}\mathbf{b}}. \tag{4.35}$$

In fact, it can be shown that the dominant zeros can be computed as the dominant poles of the inverse transfer function $[H(s)]^{-1} = \mathbf{c}_z^*(sE_z - A_z)^{-1}\mathbf{b}_z + d_z$, which has the realization

$$A_z = \begin{bmatrix} A & \mathbf{b} \\ \mathbf{c}^T & d \end{bmatrix}, \; E_z = \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix},$$

$$\mathbf{b}_z = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \quad \mathbf{c}_z = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad d_z = 0,$$

In other words, the dominant zeros of $H(s)$ can be computed by applying DPA to $[H(s)]^{-1}$. See [73] for further details.

4.2.3.3 Polynomial Eigenvalue Problems

The main idea of using Newton's method to find dominant poles can be generalized to higher order systems [84]. For the second-order transfer function $H(s) = \mathbf{c}^*(s^2M + sC + K)^{-1}\mathbf{b}$, for instance, the scheme becomes

$$s_{k+1} = s_k - \frac{\mathbf{c}^*\mathbf{v}}{\mathbf{w}^*(2s_kM + C)\mathbf{v}},$$

where $\mathbf{v} = (s_k^2M + s_kC + K)^{-1}\mathbf{b}$ and $\mathbf{w} = (s_k^2M + s_kC + K)^{-*}\mathbf{c}$. The efficient use of subspace acceleration on large scale second-order eigenvalue problems is described in [84].

4.2.3.4 Computing Eigenvalues Sensitive to Parameter Changes

Let $p \in \mathbb{R}$ be a system parameter (e.g., a resistor value R, or $1/R$, in an electric circuit), and let $A(p)$ and $E(p)$ be matrices that depend on p. The derivative of an eigenvalue λ of the pencil $(A(p), E(p))$, with left and right eigenvectors $\mathbf{y} \equiv \mathbf{y}(p)$ and $\mathbf{x} \equiv \mathbf{x}(p)$, to a parameter p is given by [66, 75]

$$\frac{\partial \lambda}{\partial p} = \frac{\mathbf{y}^*(\frac{\partial A}{\partial p} - \lambda\frac{\partial E}{\partial p})\mathbf{x}}{\mathbf{y}^*E\mathbf{x}}. \tag{4.36}$$

The derivative (4.36) is often called the sensitivity (coefficient) of λ. Assuming that $\frac{\partial E}{\partial p} = 0$, with $\mathbf{y}$ and $\mathbf{x}$ scaled so that $\mathbf{y}^*E\mathbf{x} = 1$, the eigenvalue derivative (4.36) becomes

$$\frac{\partial \lambda}{\partial p} = \mathbf{y}^*\frac{\partial A}{\partial p}\mathbf{x}. \tag{4.37}$$

The larger the magnitude of the derivative (4.37), the more sensitive eigenvalue λ is to changes in parameter p. In practical applications such information is useful when, for instance, a system needs to be stabilized by moving poles from the right half-plane to the left half-plane [83, 95].

Suppose that the derivative of A to parameter p has rank 1 and hence can be written as

$$\frac{\partial A}{\partial p} = \mathbf{b}\mathbf{c}^*, \tag{4.38}$$

where $\mathbf{b}, \mathbf{c} \in \mathbb{R}^n$ are vectors. Then the sensitivity of an eigenvalue λ with left and right eigenvectors $\mathbf{y}$ and $\mathbf{x}$ (with $\mathbf{y}^*E\mathbf{x} = 1$) becomes

$$\frac{\partial \lambda}{\partial p} = \mathbf{y}^*\frac{\partial A}{\partial p}\mathbf{x} = (\mathbf{y}^*\mathbf{b})(\mathbf{c}^*\mathbf{x}) = (\mathbf{c}^*\mathbf{x})(\mathbf{y}^*\mathbf{b}). \tag{4.39}$$

In the right-hand side of (4.39) one recognizes the residues of the transfer function $H(s) = \mathbf{c}^*(sE - A)^{-1}\mathbf{b}$. Consequently, the most sensitive eigenvalues of the pencil $(A(p), E)$ can be computed by applying DPA to $(E, A, \mathbf{b}, \mathbf{c})$, with $\mathbf{b}$ and $\mathbf{c}$ defined by (4.38).

If $\frac{\partial A}{\partial p}$ has rank higher than 1, one can change Algorithm 4.1 as follows to compute the most sensitive eigenvalues: replace $\mathbf{b}$ and $\mathbf{c}$ by $\frac{\partial A}{\partial p}\mathbf{v}_{k-1}$ and $\left(\frac{\partial A}{\partial p}\mathbf{w}_{k-1}\right)^*$, respectively. The algorithm based on this is called SASPA. For more details and generalizations to higher rank derivatives and multiparameter systems, see [83].

Having obtained, with the use of SADPA [82] or SAMDP [81], a reduced model for a large scale system incorporating feedback controllers at nominal parameters, one may want to find other reduced models for off-nominal parameters in these controllers. The SADPA and SAMDP are ideal algorithms for this application, since they benefit from the reduced model information for the nominal parameters. Note that only a true modal equivalent can benefit from this sensitivity feature, through the use of the SASPA [83].

4.2.4 Improving Krylov Models by Using Dominant Poles

It is well known that for some examples moment matching works well, while reduced order models computed by modal approximation are of low quality, and the other way around [58, 80]. Generally speaking, modal approximation performs best if there are $k \ll n$ dominant poles with residues much larger than the residues of the non-dominant poles. In other words, there is a $k \ll n$ for which one has $|R_1| \geq |R_2| \geq \ldots \geq |R_k| \gg |R_{k+1}| \geq |R_{n-1}| \geq |R_n|$, so that truncation at the kth pole does not give a large error [64]. Moment matching based approaches, on the other hand, perform best if the moments show a similar steep decay. There is, however, one additional complication for Krylov based moment matching approaches, that is best explained by an example. Figure 4.10 shows the Bode magnitude plots of an exact transfer function and of two reduced order models: one modal approximation and a moment matching approximation. While the modal approximation captures the dominant dynamics, the moment matching model deviates for $\omega > 4$ rad/s.

The modal approximation matches the original transfer function well because it is built from the 7 most dominant poles. The moment matching Arnoldi model ($k = 30$) was built using left and right Krylov subspaces with shift $s_0 = 0$. Therefore, the match for frequencies up to $\omega = 4$ rad/s is good. For higher frequencies, however, this approach suffers from a well known property of Arnoldi methods, that were originally developed for the computation of eigenvalues: the eigenvalue approximations, or Ritz values, tend to approximate the eigenvalues at the outside of the spectrum [93]. This can also be seen in Fig. 4.11, where the circles denote the poles of the moment matching model (note the inverses of the poles are shown): they match the eigenvalues at the outside. The dominant poles, however,

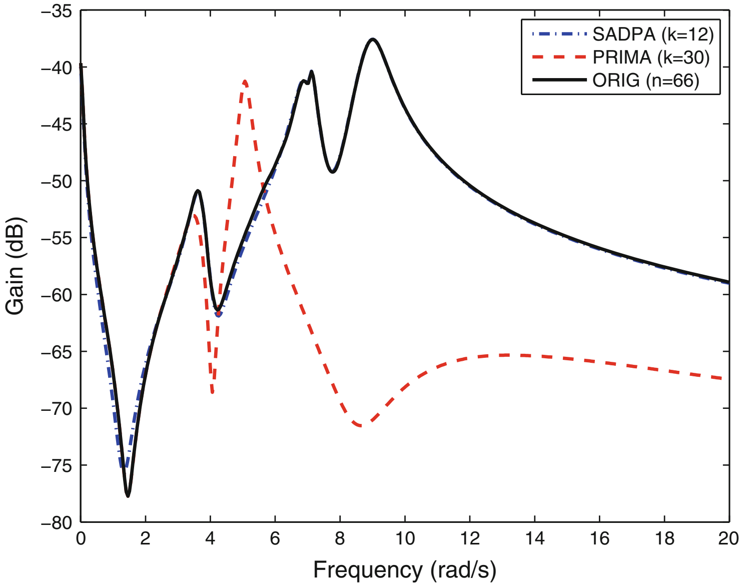


Fig. 4.10 Frequency response of complete system ($n = 66$), modal approximation ($k = 12$), and dual Arnoldi model ($k = 30$)

may be located anywhere in the spectrum, as can also be seen in Fig. 4.11 (squares). This explains why the Arnoldi model fails to capture the peaks.

Based on the above observations and theory in [65], the idea is to use the imaginary parts of dominant poles as shifts for the rational Krylov approach, so that resonance peaks located well within the system frequency bandwidth can also be captured by Krylov methods. The combined dominant pole – Krylov approach can be summarized as follows:

1. Compute $k \ll n$ dominant poles $\lambda_j = \alpha_j \pm \beta_j i$, with $j = 1, \dots k$ and $i = \sqrt{-1}$.
2. Choose interpolation points $s_j = \beta_j i$.
3. Construct $V_j, W_j \in \mathbb{C}^{n \times k_j}$ such that their columns are bases for the rational Krylov [86] spaces

$$\operatorname{colspan}(V_j) = \mathcal{K}^{k_j}((s_j E - A)^{-1} E, (s_j E - A)^{-1} \mathbf{b})$$

and

$$\operatorname{colspan}(W_j) = \mathcal{K}^{k_j}((s_j E - A)^{-*} E^*, (s_j E - A)^{-*} \mathbf{c}),$$

respectively.

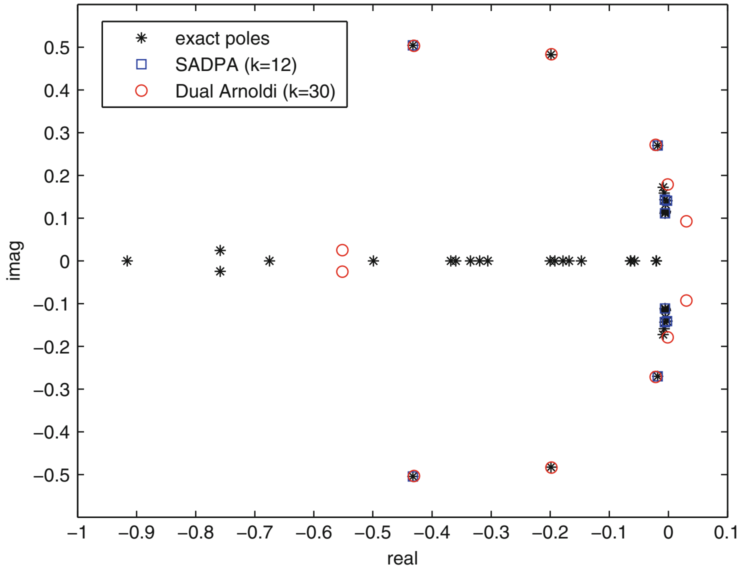


Fig. 4.11 Relevant part of pole spectrum of complete system ($n = 66$), modal approximation ($k = 12$), and dual Arnoldi model ($k = 30$)

4. Project with $V = \mathrm{orth}([V_1, \ldots, V_k])$ and $W = \mathrm{orth}([W_1, \ldots, W_k])$, where orth constructs an orthogonal basis for the spaces. The size of the reduced model is at most $K = \sum_{j=1}^{k} k_j$, matching $2K$ moments.

4.2.5 *Numerical Examples*

4.2.5.1 Brazilian Interconnected Power System (BIPS)

The Brazilian Interconnected Power System (BIPS) is a year 1999 planning model that has been used in practice (see [82] for more technical details). The size of the sparse matrices A and E is $n = 13{,}251$ (the number of states in the dense state space realization is 1,664). The corresponding transfer function has a non-zero direct transmission term d. Figure 4.12 shows the frequency response of the complete model and the reduced model (41 states) together with the error. Both the magnitude and the phase plots show good matches of the exact and the reduced transfer functions (a relative error of approximately $||H(s) - H_k(s)||/||H_k(s)|| = 0.1$, also for the DC-gain $H(0)$). Figure 4.13 shows the corresponding step response

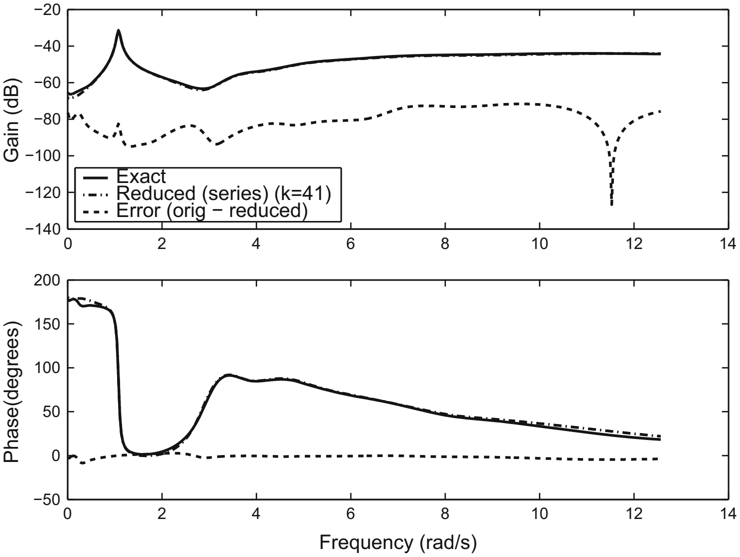


Fig. 4.12 Bode plot (with modulus and phase) of the modal equivalent, the complete model and the error for the transfer function $P_{sc}(s)/B_{sc}(s)$ of BIPS (41 in the modal equivalent, 1664 in the complete model)

(step $u = 0.01$).[15] The reduced model nicely captures the system oscillations. The reduced model (30 poles, 56 states) was computed by SADPA in 341 *LU*-factorizations ($k_{min} = 1$, $k_{max} = 10$). This reduced model could be reduced further to 41 states (22 poles) by removing less dominant contributions, without decreasing the quality of the reduced model much.

Sensitivity of BIPS

To study the sensitivity of the dominant poles and system stability of BIPS, the gain (Kpss) of one of the generators is varied between 0 and 30, with increments of 0.5. Figure 4.14 shows the traces for the most sensitive poles as computed by SASPA (Sect. 4.2.3.4, see also [83]). The CPU time needed for the 60 runs was 1,450 s. A root-locus plot for all poles, computed using the QR method, confirmed that the most sensitive poles were found, but needed 33,600 s. Hence, for large-scale systems, SASPA is a very effective and efficient way to produce relevant root-locus plots.

[15] If $h_k(t)$ is the inverse Laplace transform of $H_k(s)$, the step response for step $u(t) = c$ of the reduced model is given by $y(t) = \int_0^t h(t)u(t) = c(\sum_{i=1}^{k}(\frac{R_i}{\lambda_i}(\exp(\lambda_i t) - 1)) + d)$.

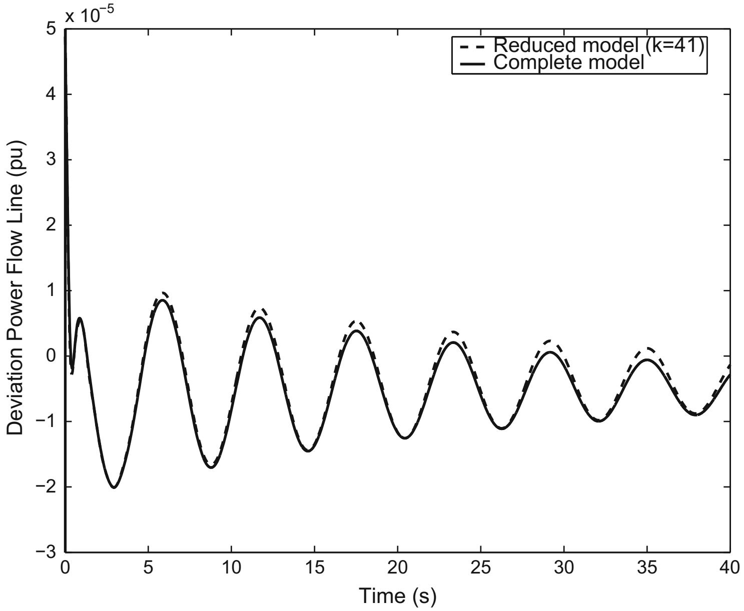


Fig. 4.13 Step responses for transfer function $P_{sc}(s)/B_{sc}(s)$ of BIPS, complete model and modal equivalent (41 states in the modal equivalent, 1664 in the complete model, step disturbance of 0.01 pu)

4.2.5.2 The Breathing Sphere

Figure 4.15 shows the frequency response of a 70th order Second-Order Arnoldi [59] reduced model of vibrating body from sound radiation analysis ($n = 17{,}611$ degrees of freedom, see [71]), that was computed using the complex parts $i\beta$ of five dominant poles $\lambda = \alpha + i\beta$ (computed by Quadratic DPA [84]) as interpolation points, as described in Sect. 4.2.4. This model is more accurate than reduced order models based on standard Krylov methods and matches the peaks up to $\omega = 1$ rad/s, because of use of shifts near the resonance frequencies. This model is a good example of the combined dominant pole – rational Krylov approach, since modal approximations of similar quality require too much CPU time, while Krylov models with uniformly spaced shifts do not capture the peaks.

4.2.6 Concluding Remarks

In this chapter eigenvalue methods, based on the Dominant Pole Algorithm, for the computation of a few specific eigenvalues were discussed. The methods can be used to solve large-scale eigenvalue problems arising in real-life applications

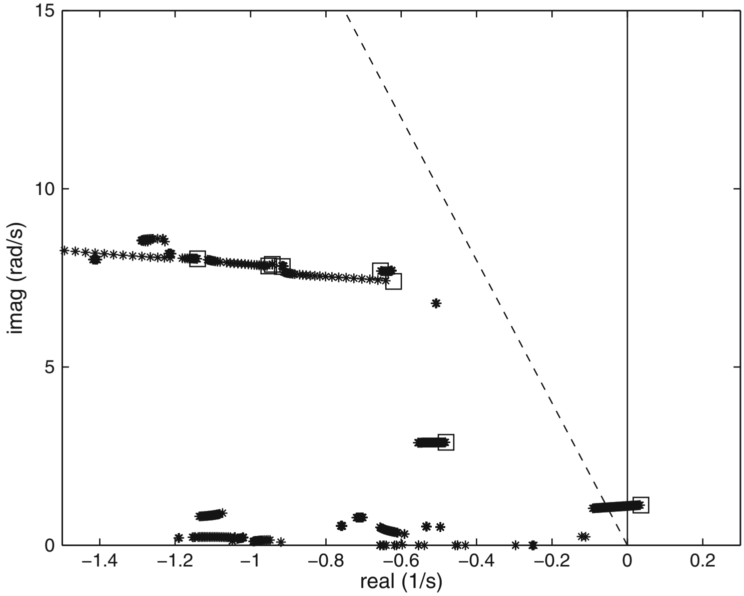


Fig. 4.14 Root locus plot of sensitive poles computed by SASPA. As the gain increases, the critical rightmost pole crosses the imaginary axis and the 5 % damping ratio boundary. Squares denote initial pole locations

and simulation of dynamical systems, for instance for the computation of transfer function dominant poles and zeros, and eigenvalues most sensitive to parameter changes. Furthermore, the corresponding eigenvectors can be used for construction of reduced-order models (modal equivalents) or to improve Krylov-based models. The dominant poles can be used to determine shifts in rational Krylov methods for computing reduced-order models. The practical application of the algorithms was illustrated by numerical experiments with real-life examples.

4.3 Passivity Preserving Model Order Reduction

In this Section we are concerned with dynamical systems $\sum = (\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ of the form

$$\begin{cases} \mathbf{E}\dot{\mathbf{x}}(t) = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \ \mathbf{y}(t) & = \mathbf{C}^{*}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{cases} \tag{4.40}$$

where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n \times n}$, $\mathbf{E}$ may be singular (we assume $\mathbf{E}$ is symmetric and positive (semi) definite), $\mathbf{B} \in \mathbb{R}^{n \times m}$, $\mathbf{C} \in \mathbb{R}^{n \times p}$, $\mathbf{D} \in \mathbb{R}^{p \times m}$, $\mathbf{x}(t) \in \mathbb{R}^{n}$, $\mathbf{y}(t) \in \mathbb{R}^{p}$ and

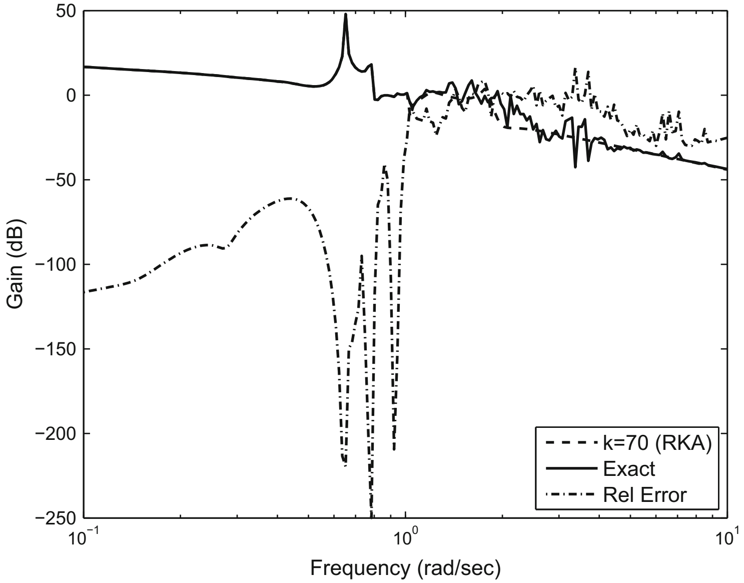


Fig. 4.15 Exact and reduced system transfer functions for a vibrating body, computed by a rational Krylov method with resonance frequencies as complex interpolation points

$\mathbf{u}(t) \in \mathbb{R}^m$.[16] The matrix **E** is called the *descriptor matrix*, the matrix **A** is called the *state space matrix*, the matrices **B** and **C** are called the *input* and *output map*, respectively, and **D** is the *direct transmission map*. The vectors $\mathbf{u}(t)$ and $\mathbf{x}(t)$ are called the *input* and the *state vector*, respectively, and $\mathbf{y}(t)$ is called the *output of the system*. The dimension n of the state is defined as the *complexity* of the system $\sum$. These systems often arise in circuit simulation, for instance, and in this application the system $\sum$ is often *passive*.[17]

The transfer function $\mathbf{G} : \mathbb{C}^m \to \mathbb{C}^p$, of (4.40),

$$\mathbf{G}(s) = \mathbf{C}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D},$$

can be obtained by applying the Laplace transform to (4.40) under the condition $\mathbf{x(0)} = \mathbf{0}$. The transfer function relates outputs to inputs in the frequency domain via $\mathbf{Y}(s) = \mathbf{G}(s)\mathbf{U}(s)$, where $\mathbf{Y}(s)$ and $\mathbf{U}(s)$ are the Laplace transforms to $\mathbf{y}(t)$ and $\mathbf{u}(t)$, respectively.

[16] Section 4.3 has been written by: Maryam Saadvandi and Joost Rommes. For further details see the MSc-Thesis of the first author [108]. Her further research is found in her Ph.D.-Thesis [109].

[17] Passivity condition is one of the important concepts and many researches have been studying it, [97–101, 104, 106, 107].

We want to reduce the original system $\sum$ to a reduced order model $\hat{\sum} = (\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \mathbf{D})$

$$\begin{cases} \hat{E}\dot{\hat{\mathbf{x}}}(t) = & \hat{A}\hat{\mathbf{x}}(t) + \hat{B}\mathbf{u}(t) \\ \hat{\mathbf{y}}(t) & = \hat{\mathbf{C}}^*\hat{\mathbf{x}}(t) + D\mathbf{u}(t), \end{cases} \tag{4.41}$$

where $\hat{\mathbf{A}}, \hat{\mathbf{E}} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{B}} \in \mathbb{R}^{k\times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{k\times p}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$, $\hat{\mathbf{x}}(t) \in \mathbb{R}^k$, $\hat{\mathbf{y}}(t) \in \mathbb{R}^p$, $\mathbf{u}(t) \in \mathbb{R}^m$ and $k \ll n$.

It is important to produce a reduced model that preserves stability and passivity.

Remark 4.1 Throughout the reminder of this chapter it is assumed that:

- $m = p$ such that $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$ and $\mathbf{D} \in \mathbb{R}^{p\times p}$.
- $\mathbf{A}$ is a stable matrix i.e. $Re(\lambda_i) < 0$ with $\lambda_i \in \sigma(\mathbf{A}), i = 1, \cdots, n$.
- The system $\sum$ is observable and controllable [112] and it is passive.

Spectral zeros play an important role in guaranteeing passivity as will be explained in the next sections. In Section 4.3.3 the spectral zeros and the method for computing them are introduced. In the following we describe two projection reduced order methods from literature for reducing the system, that aim to produce a reduced transfer function, which has the specified roots at selected spectral zeros. These methods have been developed by Sorensen [110] and Antoulas [96].

4.3.1 Model Reduction via Projection Matrices

We assume that $\mathcal{M}$ and $\mathcal{N}$ are k-dimensional subspaces of $\mathbb{R}^n$. $\mathbf{V}$ and $\mathbf{W}$ are built for reducing the system by a projection method. So we construct $\mathbf{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_k\} \in \mathbb{R}^{n\times k}$, of which the column vectors $\mathbf{v}_i$ form a basis of $\mathcal{M}$, and $\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_k\} \in \mathbb{R}^{n\times k}$, of which the column vectors $\mathbf{w}_j$ form a basis of $\mathcal{N}$ (we are interested in $W^*V = I_k$). We assume that $\mathbf{V}$ and $\mathbf{W}$ are time-invariant.

We suppose $\mathbf{x} \in \mathcal{M}$ is an approximate solution of Σ. Hence we can write $\mathbf{x} = \mathbf{V}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} \in \mathbb{R}^k$ and $\dot{\mathbf{x}} = \mathbf{V}\dot{\hat{\mathbf{x}}}$. Then the residual is

$$\mathbf{E}\dot{\mathbf{x}} - \mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{u} = \mathbf{E}\mathbf{V}\dot{\hat{\mathbf{x}}} - \mathbf{A}\mathbf{V}\hat{\mathbf{x}} - \mathbf{B}\mathbf{u}.$$

Next, we assume that this residual is orthogonal to $\mathcal{N}$

$$\begin{aligned} \mathbf{W}^*(\mathbf{E}\mathbf{V}\dot{\hat{\mathbf{x}}} - \mathbf{A}\mathbf{V}\hat{\mathbf{x}} - \mathbf{B}\mathbf{u}) \quad &= \mathbf{0}, \\ \Rightarrow \mathbf{W}^*\mathbf{E}\mathbf{V}\dot{\hat{\mathbf{x}}} - \mathbf{W}^*\mathbf{A}\mathbf{V}\hat{\mathbf{x}} - \mathbf{W}^*\mathbf{B}\mathbf{u} &= \mathbf{0}. \end{aligned}$$

Then the reduced model $\hat{\Sigma}$ becomes:

$$\begin{cases} \hat{\mathbf{E}}\dot{\hat{\mathbf{x}}}(t) = & \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}\mathbf{u}(t), \\ \hat{\mathbf{y}}(t) & = \hat{\mathbf{C}}^*\hat{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t), \end{cases}$$

where $\hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{E}} = \mathbf{W}^*\mathbf{E}\mathbf{V} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B} \in \mathbb{R}^{k\times m}$, $\hat{\mathbf{C}} = \mathbf{C}\mathbf{V} \in \mathbb{R}^{k\times p}$, $\hat{\mathbf{x}}(t) = \mathbf{V}\hat{\mathbf{x}} \in \mathbb{R}^k$ and $\mathbf{y} = \hat{\mathbf{y}}(t) \in \mathbb{R}^p$ [105].

4.3.2 Passive Systems

We can reduce the model by **V** and **W**, which are constructed in the previous Sect. 4.3.1. With arbitrary **V** and **W**, some features of the original system may not be preserved. One of these properties, which we are interested in to preserve, is *passivity*.

The matrix **A** is assumed to be stable, which means all its eigenvalues are in the open left half-plane. Passivity is defined using an energy concept.

Definition 4.3 A system is *passive* if it does not generate energy internally, and *strictly passive* if it consumes or dissipates input energy [110].

In other words Σ is *passive if*

$$Re \int_{-\infty}^{t} \mathbf{u}(\tau)^*\mathbf{y}(\tau)d\tau \geq 0, \qquad \forall t \in \mathbb{R}, \quad \forall \mathbf{u} \in \mathcal{L}_2(\mathbb{R})$$

or *strictly passive* if

$$\exists\delta > 0 \quad \text{s.t. } Re \int_{-\infty}^{t} \mathbf{u}(\tau)^*\mathbf{y}(\tau)d\tau \geq \delta \ \cdot Re \int_{-\infty}^{t} \mathbf{u}(\tau)^*\mathbf{u}(\tau)d\tau, \ \ \forall t \in \mathbb{R}, \ \ \forall \mathbf{u} \in \mathcal{L}_2(\mathbb{R})$$

Another more practical definition of passivity is based on the transfer function $\mathbf{G}(s)$ in the frequency domain:

Definition 4.4 [110] The system Σ is passive iff the transfer function $\mathbf{G}(s)$ is positive real, which means that:

1. $\mathbf{G}(s)$ is analytic for $Re(s) > 0$,
2. $\mathbf{G}(\bar{s}) = \overline{\mathbf{G}(s)}$, $\forall s \in \mathbb{C}$,
3. $\mathbf{G}(s) + (\mathbf{G}(s))^* \geq 0$ for $Re(s) > 0$ where

$$(\mathbf{G}(s))^* = \mathbf{B}^*(s\mathbf{E}^* - \mathbf{A}^*)^{-1}\mathbf{C} + \mathbf{D}^*.$$

We try to construct the **V** and **W** in such a way that the transfer function of the reduced model has the above three properties. Property 3 implies the existence of a stable rational matrix function $\mathbf{K}(s) \in \mathbb{R}^{p\times p}$ (with stable inverse) such that

$$\mathbf{G}(s) + (\mathbf{G}(-s))^* = \mathbf{K}(s)\mathbf{K}^*(-s).$$

We prove this only for the scalar case $p = 1$ of the transfer function. Let $G(s)$ be a scalar, positive-real transfer function with real coefficients. The spectral zeros of G are defined as the zeros of $G(s) + G^*(-s)$. Since all coefficients of G are real, we have $G^*(-s) = G(-s)$. Since $G(s)$ is scalar, we can write $G(s) = \frac{n(s)}{d(s)}$, where $n(s)$ and $d(s)$ are polynomials of degree $\leq k+1$ (in this note we assume k is even; a similar explanation holds when k is odd). Note that $(G(-s))^* = \frac{n^*(-s)}{d^*(-s)}$. Now we have

$$\begin{aligned} G(s) + (G(-s))^* &= \frac{n(s)}{d(s)} + \frac{n^*(-s)}{d^*(-s)} \\ &= \frac{n(s)d^*(-s) + d(s)n^*(-s)}{d(s)d^*(-s)} \\ &= \frac{r(s)r^*(-s)}{d(s)d^*(-s)}. \end{aligned} \tag{4.42}$$

We focus on proving (4.42). We will use the following identies:

$$n(s) = \sum_{i=0}^{k/2} \nu_{2i}s^{2i} + \sum_{i=0}^{k/2} \nu_{2i+1}s^{2i+1} = A + B,$$

$$d(s) = \sum_{i=0}^{k/2} \delta_{2i}s^{2i} + \sum_{i=0}^{k/2} \delta_{2i+1}s^{2i+1} = C + D.$$

It is easy to see that

$$n(-s) = \sum_{i=0}^{k/2} \nu_{2i}s^{2i} - \sum_{i=0}^{k/2} \nu_{2i+1}s^{2i+1} = A - B,$$

$$d(-s) = \sum_{i=0}^{k/2} \delta_{2i}s^{2i} - \sum_{i=0}^{k/2} \delta_{2i+1}s^{2i+1} = C - D.$$

For the sum $n(s)d(-s) + n(-s)d(s)$ we then have

$$\begin{aligned} n(s)d(-s) + n(-s)d(s) &= (A + B)(C - D) + (A - B)(C + D) = 2AC - 2BD \\ &= 2\left[\sum_{i=0}^{k/2} \nu_{2i} s^{2i}\right]\left[\sum_{i=0}^{k/2} \delta_{2i} s^{2i}\right] \\ &\quad -2\left[\sum_{i=0}^{k/2} \nu_{2i+1} s^{2i+1}\right]\left[\sum_{i=0}^{k/2} \delta_{2i+1} s^{2i+1}\right] \\ &= \tilde{v}(s) - \tilde{w}(s). \end{aligned}$$

Note that

$$\begin{aligned} \tilde{v}(s) &= \alpha_0 + \alpha_1 s^2 + \alpha_2 s^4 + \cdots + \alpha_k s^{2k}, \\ \tilde{w}(s) &= \beta_1 s^2 + \beta_2 s^4 + \beta_3 s^6 + \cdots + \beta_{k+1} s^{2k+2}. \end{aligned}$$

So, we have

$$t(s) := \tilde{v}(s) - \tilde{w}(s) = \alpha_0 + (\alpha_1 - \beta_1)s^2 + \cdots + (\alpha_k - \beta_k)s^{2k} - \beta_{k+1} s^{2k+2}.$$

Clearly, if s_0 is a zero of $t(s)$, so is $-s_0$. Consequently, we can factorize $t(s)$ as $t(s) = r(s)r(-s)$. Summarizing, we finally have

$$n(s)d(-s) + n(-s)d(s) = \tilde{v}(s) - \tilde{w}(s) = t(s) = r(s)r(-s),$$

which proves (4.42). ■

This last result equals $K(s)K^*(-s)$, i.e., this is the *spectral factorization* of G. Here K is a called the *spectral factor* of G. The zeros of K, i.e. the λ_i, $i = 1, \cdots, n$ such that $det(K(\lambda_i)) = 0$, are the *spectral zeros* of G.

4.3.3 Spectral Zeros and Generalized Eigenvalue Problem

We start this section with explaining a generalized eigenvalue problem, which Sorensen used in [110]. It brings together the theory of positive real interpolation by Antoulas [96] and the invariant subspace method for interpolating the spectral zeros by Sorensen.

First we recall that for the transfer function $\mathbf{G}(s)$ we have

$$\begin{aligned} \mathbf{G}(s) &= \mathbf{C}^*(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \quad \text{and thus,} \\ (\mathbf{G}(-s))^* &= \mathbf{B}^*(-s\mathbf{E}^* - \mathbf{A}^*)^{-1}\mathbf{C} + \mathbf{D}^*, \\ &= \mathbf{B}^*(s\mathbf{E}^* - (-\mathbf{A}^*))^{-1}(-\mathbf{C}) + \mathbf{D}^*. \end{aligned}$$

Then we compute $\mathbf{G}+\mathbf{G}^*$,[18]

$$\begin{aligned}\mathbf{G}(s)+(\mathbf{G}(-s))^* = \quad & (\mathbf{C}^*(s\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}+\mathbf{D})+(\mathbf{B}^*(s\mathbf{E}^*-(-\mathbf{A}^*))^{-1}(-\mathbf{C})+\mathbf{D}^*)\\ &= \begin{bmatrix}\mathbf{C}^* & \mathbf{B}^*\end{bmatrix}\begin{bmatrix}(s\mathbf{E}-\mathbf{A})^{-1} & 0\\ 0 & (s\mathbf{E}^*-(-\mathbf{A}^*))^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{B}\\ -\mathbf{C}\end{bmatrix}+(\mathbf{D}+\mathbf{D}^*)\\ &= \begin{bmatrix}\mathbf{C}^* & \mathbf{B}^*\end{bmatrix}\left(s\begin{bmatrix}\mathbf{E} & 0\\ 0 & \mathbf{E}^*\end{bmatrix}-\begin{bmatrix}\mathbf{A} & 0\\ 0 & -\mathbf{A}^*\end{bmatrix}\right)^{-1}\begin{bmatrix}\mathbf{B}\\ -\mathbf{C}\end{bmatrix}+(\mathbf{D}+\mathbf{D}^*).\end{aligned}$$

Note that this is the transfer function of the following system:

$$\begin{cases}\begin{bmatrix}\mathbf{E} & 0\\ 0 & \mathbf{E}^*\end{bmatrix}\dot{\mathbf{x}}(t) = \begin{bmatrix}\mathbf{A} & 0\\ 0 & -A^*\end{bmatrix}\mathbf{x}(t)+\begin{bmatrix}\mathbf{B}\\ -C\end{bmatrix}\mathbf{u}(t)\\ \\ \qquad\mathbf{y}(t) \qquad = \begin{bmatrix}\mathbf{C} & \mathbf{B}\end{bmatrix}^*\mathbf{x}(t)+(\mathbf{D}+\mathbf{D}^*)u(t)\end{cases} \tag{4.43}$$

Let

$$\mathcal{A}=\begin{bmatrix}\mathbf{A} & \mathbf{0} & \mathbf{B}\\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C}\\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^*\end{bmatrix} \quad \text{and} \quad \mathcal{E}=\begin{bmatrix}\mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0}\end{bmatrix}.$$

The finite *spectral zeros* of G are the set of all finite complex numbers λ such that

$$\operatorname{Rank}(\mathcal{A}-\lambda\mathcal{E}) < 2n+p,$$

i.e., the finite generalized eigenvalues $\sigma(\mathcal{A},\mathcal{E})$. The set of spectral zeros is denoted as $\mathcal{S}_G$.

Lemma 4.1 *If λ is a generalized eigenvalue $\sigma(\mathcal{A}-\lambda\mathcal{E})$ in $\mathcal{S}_G$ then $-\bar{\lambda}$ also belongs to $\mathcal{S}_G$, i.e.,*

$$\lambda\in\mathcal{S}_G \Rightarrow -\bar{\lambda}\in\mathcal{S}_G \quad \textit{since} \quad \mathcal{A}\mathbf{q}=\lambda\mathcal{E}\mathbf{q} \Rightarrow \tilde{\mathbf{q}}^*\mathcal{A}=(-\bar{\lambda})\tilde{\mathbf{q}}^*\mathcal{E},$$

where $\mathbf{q}^*=[\mathbf{x}^*,\mathbf{y}^*,\mathbf{z}^*]$ is a right eigenvector and $\tilde{\mathbf{q}}^*=[\mathbf{y}^*,-\mathbf{x}^*,\mathbf{z}^*]$. Also

$$\lambda\in\mathcal{S}_G \Rightarrow -\bar{\lambda}\in\mathcal{S}_G \quad \textit{since} \quad \mathbf{r}\mathcal{A}=\lambda\mathbf{r}\mathcal{E} \Rightarrow \mathcal{A}\tilde{\mathbf{r}}^*=(-\bar{\lambda})\mathcal{E}\tilde{\mathbf{r}}^*,$$

where $\mathbf{r}^*=[\mathbf{x1}^*,\mathbf{y1}^*,\mathbf{z1}^*]$ is a left eigenvector and $\tilde{\mathbf{r}}^*=[-\mathbf{y1}^*,\mathbf{x1}^*,\mathbf{z1}^*]$.

[18] Block wise inversion:

$$\begin{bmatrix}A & B\\ C & D\end{bmatrix}^{-1}=\begin{bmatrix}A^{-1}+A^{-1}B(D-CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D-CA^{-1}B)^{-1}\\ -(D-CA^{-1}B)^{-1}CA^{-1} & (D-CA^{-1}B)^{-1}\end{bmatrix}$$

Proof If $\lambda \in \sigma(\mathcal{A} - \lambda\mathcal{E})$ and $\mathbf{q}$ is the corresponding eigenvector then

$$\mathcal{A}\mathbf{q} = \lambda\mathcal{E}\mathbf{q}$$

or

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}$$

By taking conjugates and changing rows one obtains

$$\begin{bmatrix} \mathbf{y}^* & -\mathbf{x}^* & \mathbf{z}^* \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} = -\bar{\lambda} \begin{bmatrix} \mathbf{y}^* & -\mathbf{x}^* & \mathbf{z}^* \end{bmatrix} \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix}, \quad \text{or}$$

$$\tilde{\mathbf{q}}^*\mathcal{A} = -\bar{\lambda}\tilde{\mathbf{q}}^*\mathcal{E}.$$

Now we can conclude that $-\bar{\lambda} \in \mathcal{S}_G$ and that $\tilde{\mathbf{q}}^*$ is its corresponding eigenvector. The proof is similar for the left eigenvectors [110]. ■

If specified spectral zeros are preserved (interpolated) in the reduced model with $\mathcal{S}_{\hat{G}}$ then a passive reduced model will result. For real systems, $\mathcal{S}_{\hat{G}}$ must include conjugate pairs of spectral zeros. This result is based on Antoulas' theorem [96]:

Theorem 4.3 (Antoulas) *Suppose $\mathcal{S}_{\hat{G}} \subset \mathcal{S}_G$ and also that $\hat{G}(\lambda) = G(\lambda)$ for all $\lambda \in \mathcal{S}_{\hat{G}}$ and that $\hat{G}$ is a minimal degree rational interpolant of the values of G on the set $\mathcal{S}_{\hat{G}}$. Then the reduced system $\hat{\sum}$ with transfer function $\hat{G}$ is both stable and passive.*

4.3.4 Passivity Preserving Model Reduction

Theorem 4.3 indicates that Antoulas's approach [96] *preserves passivity for the reduced model when spectral zero interpolation is applied.* The interpolation is guaranteed by building the projection matrices using a Krylov subspace method [103, 111]. Antoulas' method [96] significantly differs from PRIMA [107]. For a detailed comparison between PRIMA and Antoulas's approach we refer to [105].

In Antoulas' method it is assumed that the system Σ with transfer function $\mathbf{G}(s)$ is passive. Then one defines a set $\mathcal{S}_1 \subset \mathcal{S}_{\mathbf{G}_{stable}}$ where $\mathcal{S}_{\mathbf{G}_{stable}}$ is the set of stable *spectral zeros* and one takes $\mathcal{S}_2 = -\mathcal{S}_1$. Antoulas [96] has shown that the reduced system $\hat{\Sigma}$ with transfer function $\hat{\mathbf{G}}(s)$ is passive if the set of interpolation points is $\mathcal{S}_1 \cup \mathcal{S}_2$.

A second approach has been introduced by Sorensen [110], which can be seen as an interpolatory model reduction too. It is based on invariant subspaces. In this method it is not necessary that the spectral zeros (interpolation points) are computed in advance. Sorensen's approach transfers the model reduction problem into an eigenvalue problem. In this case the eigenvalues are the spectral zeros of the transfer function of the original system. Then the projection matrices are built from a basis for a chosen invariant subspace.

Choosing different spectral zeros gives us different invariant subspaces, which return different reduced models. Although these reduced models are passive, they may not be a good approximation to the original system. So the selection of spectral zeros must guarantee that the reduced model is a good approximation to the original ones.

In large scale problems in which the eigen computation of the resulting highly-structured eigenvalue problem should be done iteratively, all selection criteria can not be satisfied. So the problem has two goals: the first one is to have a good approximation of the original model, the second one is to be suitable as an iterative scheme for large-scale dynamical systems.

4.3.5 Model Reduction by Projection

We will construct a basis for a selected invariant subspace of the pair $(\mathcal{A}, \mathcal{E})$ (Sorensen [110]). Let

$$\mathcal{A}\mathbf{Q} = \mathcal{E}\mathbf{Q}\mathbf{R}$$

be a partial real Schur decomposition for the pair $(\mathcal{A}, \mathcal{E})$. Then, $\mathbf{Q}^*\mathbf{Q} = I$ and $\mathbf{R}$ is real and quasi-upper triangular. Let $Q = [\mathbf{X}^*, \mathbf{Y}^*, \mathbf{Z}^*]^*$ be partitioned in accordance with the block structure of $\mathcal{A}$:

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} \mathbf{R}$$

$$\Rightarrow \begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C} \\ \mathbf{C}^* & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{E}\mathbf{X} \\ \mathbf{E}^*\mathbf{Y} \\ \mathbf{0} \end{bmatrix} \mathbf{R} \tag{4.44}$$

The projection will be constructed from $\mathbf{X}$ and $\mathbf{Y}$ and the reduced model will be obtained out of these. Here it will be useful to have the following lemma [110].

Lemma 4.2 *Suppose that* $\mathbf{R}$ *in* (4.44) *satisfies* $Re(\lambda) > 0, \ \forall\lambda \in \sigma(\mathbf{R})$. *Then* $\mathbf{X}^*\mathbf{E}^*\mathbf{Y} = \mathbf{Y}^*\mathbf{E}\mathbf{X}$ *is symmetric.*

Proof We start with

$$\mathcal{A}\mathbf{Q} = \mathcal{E}\mathbf{Q}\mathbf{R}. \tag{4.45}$$

By (4.45) and according to the previous proof we have

$$\hat{\mathbf{Q}}^*\mathcal{A} = (-\mathbf{R}^*)\hat{\mathbf{Q}}^*\mathcal{E} \quad \text{where} \quad \hat{\mathbf{Q}}^* = \begin{bmatrix} \mathbf{Y}^* & -\mathbf{X}^* & \mathbf{Z}^* \end{bmatrix}, \tag{4.46}$$

If we multiply equation (4.45) with $\hat{\mathbf{Q}}^*$ from the left, then we get

$$\hat{\mathbf{Q}}^*\mathcal{A}\mathbf{Q} = \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q}\mathbf{R}. \tag{4.47}$$

We substitute the right part of equation (4.46) in the left part of equation (4.47), giving

$$\begin{aligned} (-\mathbf{R}^*)\hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q} &= \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q}\mathbf{R} \\ \Rightarrow \quad \mathbf{R}^*\hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q} + \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q}\mathbf{R} &= \mathbf{0}. \end{aligned} \tag{4.48}$$

Here

$$\begin{aligned} \hat{\mathbf{Q}}^*\mathcal{E}\mathbf{Q} &= \begin{bmatrix} \mathbf{Y}^* & -\mathbf{X}^* & \mathbf{Z}^* \end{bmatrix} \begin{bmatrix} \mathbf{E} & & \\ & \mathbf{E}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} \\ &= \mathbf{Y}^*\mathbf{E}\mathbf{X} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y}. \end{aligned} \tag{4.49}$$

If we substitute (4.49) in (4.48) we obtain

$$\mathbf{R}^*(\mathbf{Y}^*\mathbf{E}\mathbf{X} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y}) + (\mathbf{Y}^*\mathbf{E}\mathbf{X} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y})\mathbf{R} = \mathbf{0}. \tag{4.50}$$

Therefore the equation (4.50) has the unique solution[19]:

$$\mathbf{Y}^*\mathbf{E}\mathbf{X} - \mathbf{X}^*\mathbf{E}^*\mathbf{Y} = \mathbf{0},$$

and hence

$$\mathbf{Y}^*\mathbf{E}\mathbf{X} = \mathbf{X}^*\mathbf{E}^*\mathbf{Y},$$

which completes the proof. ■

[19] Equation (4.50) is a simple form ($R^*X + XR = 0$) of a Lyapunov equation of the more general type $AX - XB = C$ (which has a unique solution if $\sigma(A) \cap \sigma(B) = \oslash$). Due to the condition $Re(\lambda) > 0$ for λ in $\sigma(R)$, we have that $\sigma(R^*) \cap \sigma(-R) = \oslash$. Hence the Lyapunov equation (4.50) has a unique (zero) solution.

For the construction of $\mathbf{V}$ and $\mathbf{W}$ as projections, we first have to find a basis for an invariant subspace [102] with all eigenvalues of $\mathbf{R}$ in the right half-plane.

Let $\mathbf{Q}_x\mathbf{S}^2\mathbf{Q}_y^* = \mathbf{X}^*\mathbf{Y}$ be the SVD of $\mathbf{X}^*\mathbf{Y}$ and note that $\mathbf{Q}_y = \mathbf{Q}_x\mathbf{J}$ where $\mathbf{J}$ is a signature matrix by virtue of the fact that $\mathbf{X}^*\mathbf{Y}$ is symmetric.

If $\mathbf{S} \geq \mathbf{0}$ is nonsingular, put

$$\begin{aligned} \mathbf{V} &= \mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1} \\ \mathbf{W} &= \mathbf{Y}\mathbf{Q}_y\mathbf{S}^{-1}. \end{aligned} \tag{4.51}$$

It follows that

$$\begin{aligned} \mathbf{W}^*\mathbf{V} &= (\mathbf{Y}\mathbf{Q}_y\mathbf{S}^{-1})^*\mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1} \\ &= \mathbf{S}^{-*}\mathbf{Q}_y^*\mathbf{Y}^*\mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1} \\ \text{(include the SVD form of } \mathbf{X}^*\mathbf{Y}) &= \mathbf{S}^{-*}\mathbf{Q}_y^*\mathbf{Q}_y(\mathbf{S}^2)^*\mathbf{Q}_x^*\mathbf{Q}_x\mathbf{S}^{-1} \\ (\mathbf{Q}_x \text{ and } \mathbf{Q}_y \text{ are unitary matrices}) &= \mathbf{S}^{-*}\mathbf{S}^*\mathbf{S}^*\mathbf{S}^{-1} \\ &= \mathbf{I}. \end{aligned}$$

and also we have

$$\mathbf{V}^*\mathbf{W} = (\mathbf{W}^*\mathbf{V})^* = \mathbf{I}.$$

Now from the SVD of $\mathbf{X}^*\mathbf{Y}$, let

$$\begin{aligned} \hat{\mathbf{X}} &= \mathbf{S}(\mathbf{Q}_x)^* \\ \hat{\mathbf{Y}} &= \mathbf{S}(\mathbf{Q}_y)^*, \end{aligned}$$

and define

$$\mathcal{V} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathcal{W} = \begin{bmatrix} \mathbf{W} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

It is obvious that $\mathcal{W}^*\mathcal{V} = \mathbf{I}$ and that

$$\begin{aligned} \mathbf{V}\hat{\mathbf{X}} &= (\mathbf{X}\mathbf{Q}_x\mathbf{S}^{-1})(\mathbf{S}\mathbf{Q}_x^*), \\ &= \mathbf{X}\mathbf{Q}_x\mathbf{Q}_x^*, \\ (\mathbf{Q}_x^* \quad \text{is unitary matrix}) &= \mathbf{X}. \end{aligned}$$

Similarly, $\mathbf{W}\hat{\mathbf{Y}} = \mathbf{Y}$, so we have

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix}.$$

Therefore

$$\hat{\mathcal{A}} = \mathcal{W}^*\mathcal{A}\mathcal{V} = \begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \hat{\mathbf{B}} \\ \mathbf{0} & -\hat{\mathbf{A}}^* & -\hat{\mathbf{C}} \\ \hat{\mathbf{C}}^* & \hat{\mathbf{B}}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \quad \text{and} \quad \hat{\mathcal{E}} = \mathcal{W}^*\mathcal{E}\mathcal{V} = \begin{bmatrix} \hat{\mathbf{E}} & & \\ & \hat{\mathbf{E}}^* & \\ & & \mathbf{0} \end{bmatrix},$$

and

$$\begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \hat{\mathbf{B}} \\ \mathbf{0} & -\hat{\mathbf{A}}^* & -\hat{\mathbf{C}} \\ \hat{\mathbf{C}}^* & \hat{\mathbf{B}}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{E}} & & \\ & \hat{\mathbf{E}}^* & \\ & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix} \mathbf{R},$$

or

$$\begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \hat{\mathbf{B}} \\ \mathbf{0} & -\hat{\mathbf{A}}^* & -\hat{\mathbf{C}} \\ \hat{\mathbf{C}}^* & \hat{\mathbf{B}}^* & \mathbf{D}+\mathbf{D}^* \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}} \\ \hat{\mathbf{Y}} \\ \hat{\mathbf{Z}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{E}}\hat{\mathbf{X}} \\ \hat{\mathbf{E}}^*\hat{\mathbf{Y}} \\ \mathbf{0} \end{bmatrix} \mathbf{R}.$$

where $\hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V}, \hat{\mathbf{E}} = \mathbf{W}^*\mathbf{E}\mathbf{V}, \hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B}$, and $\hat{\mathbf{C}} = \mathbf{V}^*\mathbf{C}$.
This shows that $\mathcal{S}_{\hat{G}} \subseteq \mathcal{S}_G$ and since $\mathcal{S}_{\hat{G}} = \sigma(\mathbf{R}) \cup \sigma(-\mathbf{R}^*)$[20] and $\sigma(\mathbf{R})$ is in the open right half-plane, the reduced model has no spectral zeros on the imaginary axis.

The previous result is also valid when $\mathbf{S}$ is nonsingular. Now we consider the case $\mathbf{S}$ is singular. Beginning with $\mathbf{X}, \mathbf{Y}$ from (4.44) and with the SVD of $\mathbf{X}^*\mathbf{Y}$,

[20] We know that if we have a real matrix $\mathbf{A}$ and $\lambda \in \sigma(\mathbf{A})$ then $\bar{\lambda} \in \sigma(\mathbf{A})$. In Lemma 4.1 we showed that if $\lambda \in \mathcal{S}_G$ then $-\bar{\lambda} \in \mathcal{S}_G$. Therefore

$$\lambda,\ \bar{\lambda},\ -\lambda \quad \text{and} -\bar{\lambda} \in \mathcal{S}_G.$$

On the other hand, $\mathbf{R}$ is a selected invariant subspace of $(\mathcal{A}, \mathcal{E})$, which means that $\sigma(\mathbf{R}) \subset \mathcal{S}_G$. Now, we need to find a basis for an invariant subspace with eigenvalues of $\mathbf{R}$ in the open right half-plane. As we mentioned above $\sigma(\mathbf{R})$ and $\sigma(-\mathbf{R}^*)$ are a subset of $\mathcal{S}_G$. Thus take

$$\mathcal{S}_{\hat{G}} = \sigma(\mathbf{R}) \cup \sigma(-\mathbf{R}^*).$$

Algorithm 4.7 Sorensen's Algorithm [110]

```
INPUT: System $(E, A, B, C, D)$,
OUTPUT: Reduced System $(\hat{E}, \hat{A}, \hat{B}, \hat{C}, D)$
Compute $\mathcal{A}, \mathcal{E}$
$[\mathcal{A}_1, \mathcal{E}_1, Z, Q, V, W] = \mathtt{qz}(\mathcal{A}, \mathcal{E})$;
Find spectral zeros, $\Lambda = \mathtt{eig}(\mathcal{A}, \mathcal{E})$;
Find the real basis for the right eigenvector matrix $V$,
Find the positive real spectral zeros and corresponding eingenvectors, $\Lambda_1 = [\,]$; $V_1 = [\,]$;
for $i = 1 : \mathtt{length}(\Lambda)$ do
    if $(\mathtt{real}(\Lambda(i)) > 0$ and $\Lambda(i)$ are chosen spectral zeros) or $\mathtt{imag}\Lambda(i) = 0$ then
        $\Lambda_1 = [\Lambda_1 \quad \Lambda(i)]$; $V_1 = [V_1 \quad V(:, i)]$;
    end if
end for
$X = V_1(1 : n, :)$; $Y = V_1(n + 1 : 2n, :)$;
$[Q_x, S^2, Q_y] = \mathtt{svd}(X^* Y)$;
Construct the projection matrices, $V = XQ_x S^{-1}$; $W = YQ_y S{-}1$;
$\hat{E} = W^* E V$; $\hat{A} = W^* A V$; $\hat{B} = W^* B$; $\hat{C} = CV$;
```

where $\mathbf{Q}_x \mathbf{S}^2 \mathbf{Q}_y = \mathbf{X}^* \mathbf{Y}$, specify a cut-off tolerance $\tau_c \in (0, 1)$ and let j be the largest positive integer such that

$$\sigma_j \geq \tau_c \sigma_1 \qquad \text{where} \quad \sigma_j = \mathbf{S}(j, j).$$

Define $\mathbf{Q}_j = \mathbf{Q}_x(:, 1 : j)$, $\mathbf{S}_j = \mathbf{S}(\mathbf{1} : \mathbf{j}, \mathbf{1} : \mathbf{j})$ and then let $(\mathbf{X}_j)^I = \mathbf{Q}_j(\mathbf{S}_j)^{-1}$. Replace $\hat{\mathbf{X}}^{-1} = (\mathbf{X}_j)^I$, $\mathbf{V} = \mathbf{X}(\mathbf{X}_j)^I$ and $\mathbf{W} = -\mathbf{Y}(\mathbf{X}_j)^I$. According to [110], in this way, the reduced system is passive and also the stability of the reduced model is obtained if $\mathbf{Z}$ is full rank.

Sorenson's Algorithm is described in Algorithm 4.7.

4.3.6 Model Reduction by Projection

We want to reduce the original system $\sum$ to $\hat{\sum}$ where the complexity k of $\hat{\sum}$ is (much) less than that of $\sum$ ($k \ll n$) (Antoulas [96]). This reduction must preserve both stability and passivity and it must be numerically efficient. Antoulas' Algorithm is described in Algorithm 4.8.

We will look for $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times k}$ such that $\mathbf{V}\mathbf{W}^*$ is a projection with the additional condition $\mathbf{W}^*\mathbf{V} = \mathbf{I}_k$ (recall that P is a projection matrix if $P^2 = P$). So, if we have $\mathbf{V}$ and $\mathbf{W}$ with $\mathbf{W}^*\mathbf{V} = \mathbf{I}_k$, then indeed

$$(\mathbf{V}\mathbf{W}^*)^2 = \mathbf{V}\mathbf{W}^*.$$

Algorithm 4.8 Antoulas's Algorithm [96]

```
INPUT: System $(E, A, B, C, D)$,
OUTPUT: Reduced System $(\hat{E}, \hat{A}, \hat{B}, \hat{C}, D)$
Compute $\mathcal{A}, \mathcal{E}$
Find spectral zeros, $\Lambda = \texttt{eig}(\mathcal{A}, \mathcal{E})$;
$\Lambda R = [\,]; \Lambda C = [\,]$;
while $n \geq \texttt{length}(\Lambda)$ do
    if $\Lambda(n)$ is positive real, $\Lambda R = [\Lambda R \quad \Lambda(n)]$;
    if $\Lambda(n)$ is complex and in right half-plane, $\Lambda C = [\Lambda C \quad \Lambda(n)]$;
end while
for $m = 1 : \texttt{length}(\Lambda C)$ do
    if $\Lambda C(m)$ chosen spectral zeros then
        $\Lambda R = [\Lambda R \quad \Lambda C(m)]$;
    end if
end for
$V = [\,]; W = [\,]$;
for $q = 1 : \texttt{length}(\Lambda R)$ do
    $v = (\Lambda R(q)E - A)^{-1}B; w = (-\Lambda R(q)E^* - A^*)^{-1}C^*$;
    $V = [V \quad v]; W = [W \quad w]$;
end for
Make a real basis for $V$ and $W$
$W = (W^*V)^{-1}W$;
$\hat{E} = W^*EV; \hat{A} = W^*AV; \hat{B} = W^*B; \hat{C} = CV$;
```

Given $2k$ distinct points $s_1, \cdots, s_{2k}$, let

$$\begin{aligned}\tilde{\mathbf{V}} &= \left[\, (s_1\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \cdots (s_k\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B} \,\right], \\ \tilde{\mathbf{W}} &= \left[\, (s_{k+1}\mathbf{I}_n - \mathbf{A}^*)^{-1}\mathbf{C} \cdots (s_{2k}\mathbf{I}_n - \mathbf{A}^*)^{-1}\mathbf{C} \,\right].\end{aligned} \tag{4.52}$$

Now take $\mathbf{V} = \tilde{\mathbf{V}}$ and $\mathbf{W} = \tilde{\mathbf{W}}(\tilde{\mathbf{V}}^*\tilde{\mathbf{W}})^{-1}$. We define

$$\hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V}, \quad \hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{V}^*\mathbf{C}. \tag{4.53}$$

Then we have the following theorem (Antoulas [96])

Proposition 4.1 *Assuming that det*$(\tilde{\mathbf{W}}^*\tilde{\mathbf{V}}) \neq \mathbf{0}$, *the projected system* $\hat{\sum}$, *defined by* (4.53), *interpolates the transfer function of* $\sum$ *at the points* s_i*:*

$$\hat{\mathbf{G}}(s_i) = \mathbf{G}(s_i) \qquad i = 1, 2, \cdots, 2k.$$

where s_i *are the spectral zeros.*

4.3.7 Numerical Results

In [108] several numerical results are presented for an RLC-circuit that is also found in [96, 110]. The transfer function is a scalar function $G(s)$. The starting point is to compute the spectral zeros (using a generalized eigenvalue method) and then to try to categorize them related to their magnitude, like distance from the real and the imaginary axis in order to have a good match in low or high frequency. The reduced method was obtained by Algorithm 4.8 of Antoulas. A large distance from the real axis results in a good approximation at high frequencies. A large distance from the imaginary axis results in a good approximation at low frequencies. In both situations including the real spectral zeros plays an important role for having a good reduced model at low frequencies.

One should check if a spectral zero also occurs as a pole and as a zero, both, in which case the factors $(\lambda\mathbf{I} - \mathbf{A})$ are singular. These spectra zeros should be left out of the reduction.

In this section we study a circuit which has a descriptor matrix $\mathbf{E} \neq \mathbf{I}_n$. We consider the circuit shown in Fig. 4.16. We assume that all capacitors and inductors have a unit value, $R_1 = \frac{1}{2}\Omega$, $R_2 = \frac{1}{5}\Omega$, $R_{2k} = \frac{1}{3}\Omega$, where $k = 2, 3, \cdots, n$ and $R_{2k+1} = \frac{1}{4}\Omega$, where $k = 1, 2, \cdots, n$.

The order of the original system is 1003 and the selected spectral zeros close to the real axis are shown in Fig. 4.17. In this case, like before, the reduced model has a good match at low and at high frequencies, as shown in Fig. 4.18.

4.3.8 Conclusion

We have considered two approaches for passive and stable reduction of dynamical systems in circuit simulation, based on the methods by Antoulas [96] and Sorenson [110] that both exploit interpolating spectral zeros. The reduced models preserve passivity and stability. The original system is reduced by projection matrices, which are built via spectral zero interpolation. Different selections of spectral zeros give us different approximations of the original model, which may/may not produce

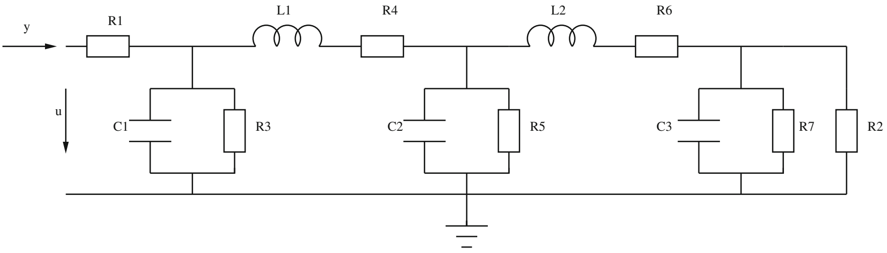


Fig. 4.16 RLC Circuit of Order 7

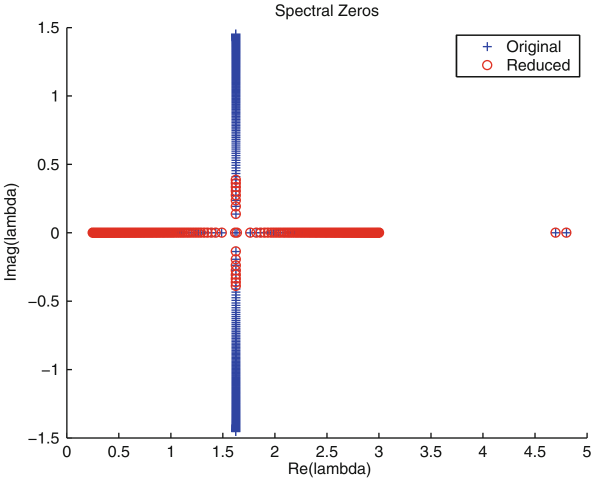


Fig. 4.17 Spectral zeros of the original model (+), and spectral zeros of the reduced model (o). For interpolation, the spectral zeros close to the real axis are chosen. All selected spectral zeros are preserved after reduction. The order of the original model is 1003 and it is reduced to 341

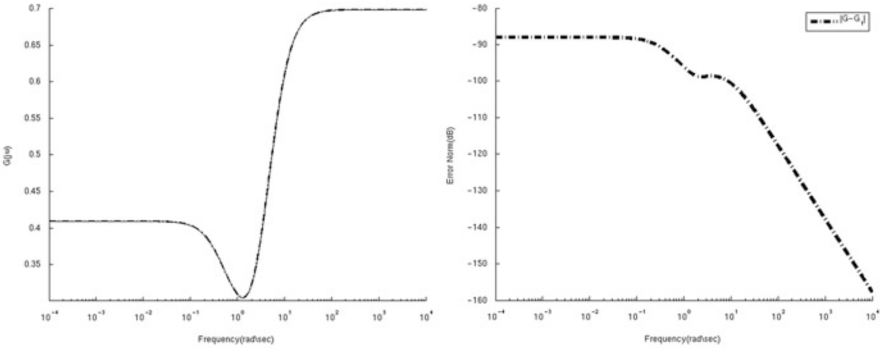


Fig. 4.18 Effect of several real spectral zeros, *Left*: Frequency responses of the original system and reduced model. The spectral zeros close to the real axis are interpolated. *Right*: Frequency response of the error $\|\Sigma - \hat{\Sigma}\|_2$

acceptable reduction. We have considered criteria for selecting the spectral zeros and also to approximate the original system well in low and high frequency. When the spectral zeros are chosen close to the real axis, the reduced model matches the original response well for low frequencies. On the other hand, when they are far from the real axis, the reduced model is more accurate for high frequencies. As already shown preserving the real spectral zero plays an important role for having a good reduction in the whole frequency domain, specially in low frequency. It means that one should try to save all the real spectral zeros of the system.

The approaches of Antoulas and Sorensen are equivalent but as Sorensen's algorithm works directly with eigenvalues and eigenvectors, it is more usable for constructing the projection matrices. For the same reason Sorensen's approach is more suitable for large scale systems.

4.4 Passivity Preserving Model Reduction Using the Dominant Spectral Zero Method

The design of integrated circuits has become increasingly complex, thus electromagnetic couplings between components on a chip are no longer negligible.[21] To verify coupling effects, on-chip interconnections are modeled as RLC circuits and simulated. As these circuits contain millions of electrical components, the underlying dynamical systems have millions of internal variables and cannot be simulated in full dimension. Model order reduction (MOR) aims at approximating the mathematical description of a large scale circuit with a model of smaller dimension, which replaces the original model during verification and speeds up simulation. The reduction method should preserve important properties of the original model (i.e., stability, passivity) and have an efficient, robust implementation, suitable for large-scale applications. RLC circuits describing the interconnect are *passive* systems, with *positive real* transfer functions [113, 116], thus reduced models should also be passive. A passive reduced model can be synthesized back into an RLC circuit [113], which is placed instead of the original in the simulation flow. Passive reduced circuits also guarantee stable simulations when integrated with the overall nonlinear macro-model [117, 128, 133] during later simulation stages.

The proposed *Dominant Spectral Zero Method (dominant SZM)* is a model reduction method which preserves passivity and stability, and is efficiently implemented using *the subspace accelerated dominant pole algorithm (SADPA)* [130, 131]. Passivity preservation is ensured via a new approach, that of interpolation at *dominant spectral zeros*, a subset of spectral zeros of the original model. Dominant SZM reduces automatically all passive systems, including those with formulations unsuitable for PRIMA (first order susceptance-based models for inductive couplings (RCS circuits) [140] or models involving controlled sources, such as vector potential equivalent circuit (VPEC) [139] and partial element equivalent circuit (PEEC) models [136]). In comparison to *positive real balanced truncation (PRBT)* [129], dominant SZM efficiently handles systems with a possibly singular $\mathbf{E}$ matrix [see (4.54)]. Unlike *modal approximation (MA)* [131, 135] where interpolation is at dominant poles, our method matches the dominant spectral zeros of the original system, guaranteeing passivity.

[21] Section 4.4 has been written by: Roxana Ionutiu, Joost Rommes and Athanasios C. Antoulas. For an extended treatment on the topics of this Section see also the Ph.D. Thesis of the first author [121].

The remainder of this section is structured as follows. The introduction continues with the mathematical setup of MOR in Sect. 4.4.1, and with a brief description of MOR via spectral zero interpolation in Sect. 4.4.2. Dominant SZM is presented concisely in Sect. 4.4.3.1 (following [125]). It is extended with the concept of dominance at ∞ (Sect. 4.4.3.2), and with an approach for converting the reduced models to circuit representations (Sect. 4.4.3.3). Numerical results follow in Sect. 4.4.4 and the section concludes with Sect. 4.4.5. Algorithmic pseudocode for the dominant SZM – SADPA implementation is given in the Appendix 4.4.6.

4.4.1 Background on MOR

The model reduction framework involves approximation of an original dynamical system described by a set of differential algebraic equations in the form:

$$\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \tag{4.54}$$

where the entries of $\mathbf{x}(t)$ are the system's internal variables, $\mathbf{u}(t)$ is the system input and $\mathbf{y}(t)$ is the system output, with dimensions $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{y}(t) \in \mathbb{R}^p$. Correspondingly, $\mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $(\mathbf{A}, \mathbf{E})$ is a regular pencil, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. The original system $\Sigma(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is stable and passive and has *dimension* n, usually very large. We seek a reduced order model $\hat{\Sigma}(\hat{\mathbf{E}}, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}, \mathbf{D})$, which satisfies: $\hat{\mathbf{E}}\dot{\hat{\mathbf{x}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{B}}\mathbf{u}(t)$, $\hat{\mathbf{y}}(t) = \hat{\mathbf{C}}\hat{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t)$, where $\hat{\mathbf{x}} \in \mathbb{R}^k$, $\hat{\mathbf{E}} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{A}} \in \mathbb{R}^{k\times k}$, $\hat{\mathbf{B}} \in \mathbb{R}^{k\times m}$, $\hat{\mathbf{C}} \in \mathbb{R}^{p\times k}$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. $\hat{\Sigma}$ is obtained by projecting the internal variables of the original system $\mathbf{x}$ onto a subspace $\mathit{ColSpan}(\mathbf{V}) \subset \mathbb{R}^{n\times k}$, along $\mathit{Null}(\mathbf{W}^*) \subset \mathbb{R}^{k\times n}$. The goal is to construct $\mathbf{V}$ and $\mathbf{W}$, such that $\hat{\Sigma}$ is stable and passive. Additionally, $\mathbf{V}$ and $\mathbf{W}$ should be computed efficiently. The reduced matrices are obtained as follows:

$$\hat{\mathbf{E}} = \mathbf{W}^*\mathbf{E}\mathbf{V},\ \hat{\mathbf{A}} = \mathbf{W}^*\mathbf{A}\mathbf{V},\ \hat{\mathbf{B}} = \mathbf{W}^*\mathbf{B},\ \hat{\mathbf{C}} = \mathbf{C}\mathbf{V}. \tag{4.55}$$

4.4.2 MOR by Spectral Zero Interpolation

We revise the spectral zero interpolation approach for model reduction as proposed in [114, 134]. The ingredient for passivity preservation are the *spectral zeros* of $\Sigma(\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$, defined as follows:

Definition 4.5 For system Σ with transfer function: $\mathbf{H}(s) := \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$, the spectral zeros are all $s \in \mathbb{C}$ such that $\mathbf{H}(s) + \mathbf{H}^*(-s) = 0$, where $\mathbf{H}^*(-s) = \mathbf{B}^*(-s\mathbf{E}^* - \mathbf{A}^*)^{-1}\mathbf{C}^* + \mathbf{D}^*$.

According to [114, 134], model reduction via spectral zero interpolation involves forming rational Krylov subspaces:

$$\begin{aligned} \mathbf{V} &= [(s_1\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}, \ \cdots, \ (s_k\mathbf{E}-\mathbf{A})^{-1}\mathbf{B}], \\ \mathbf{W} &= [(-s_1^*\mathbf{E}^*-\mathbf{A}^*)^{-1}\mathbf{C}^*, \ \cdots, \ (-s_k^*\mathbf{E}^*-\mathbf{A}^*)^{-1}\mathbf{C}^*], \end{aligned} \tag{4.56}$$

where $s_1 \dots s_k, -s_1^* \dots -s_k^*$ are a subset of the spectral zeros of Σ. By projecting the original system with matrices (4.56) according to (4.55), the reduced $\hat{\Sigma}$ interpolates Σ at the chosen s_i and their mirror images $-s_i^*$, $i = 1, \dots, k$ [113, 114]. Projection matrices $\mathbf{V}$ and $\mathbf{W}$ insure that the reduced system satisfies the positive real lemma [113, 114, 116, 134], thus passivity is preserved. If in the original system $\mathbf{D} \neq \mathbf{0}$, the reduced system is strictly passive, and realizable with RLC circuit elements. In Sect. 4.4.3.2 we show one way of obtaining strictly passive reduced systems also when $\mathbf{D} = \mathbf{0}$.

4.4.3 *The Dominant Spectral Zero Method*

The new Dominant Spectral Zero Method (dominant SZM) is presented. The spectral zero method [114, 134] is extended with a dominance criterion for selecting finite spectral zeros. These are computed efficiently and automatically using the subspace accelerated dominant pole algorithm (SADPA) [130, 131]. We show in addition how, for certain RLC models, dominant spectral zeros at ∞ can also be easily interpolated.

4.4.3.1 Dominant Spectral Zeros and Implementation

In [134] it was shown that spectral zeros are solved efficiently from an associated Hamiltonian eigenvalue problem [127, 137]. In [114, 134] however, the selection of spectral zeros was still an open problem. We propose a solution as follows: we extend the concept of *dominance* from poles [130] to spectral zeros, and adapt the iterative solver SADPA for the computation of *dominant spectral zeros*. The corresponding invariant subspaces are obtained as a by-product of SADPA, and are used to construct the passivity preserving projection matrices $\mathbf{V}$ and $\mathbf{W}$. Essentially, dominant SZM is the SADPA-based implementation of modal approximation for the Hamiltonian system associated with $\mathbf{G}(s) = [\mathbf{H}(s)+\mathbf{H}^*(-s)]^{-1}$. Recalling Def. 4.5, the spectral zeros of Σ are the poles of $\mathbf{G}(s)$, with partial fraction expansion: $\mathbf{G}(s) = \sum_{j=1}^{2n} \frac{\mathscr{R}_j}{s-s_j}$, where s_i are the poles of $\mathbf{G}$ with associated residues $\mathscr{R}_j$ [126, 131]. The modal approximate of $\mathbf{G}(s)$ is obtained by truncating this sum: $\hat{\mathbf{G}}(s) = \sum_{j=1}^{2k} \frac{\mathscr{R}_j}{s-s_j}$. The procedure is outlined next.

1. Given $\Sigma(\mathbf{E},\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D})$, construct the associated Hamiltonian system Σ_s, associated with transfer function $\mathbf{G}(s)$:

   a. Σ_s when $\mathbf{D}+\mathbf{D}^*$ is invertible:

$$\mathbf{A}_s = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C}^* \\ \mathbf{C} & \mathbf{B}^* & \mathbf{D}+\mathbf{D}^* \end{pmatrix}, \mathbf{E}_s = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \mathbf{B}_s = \begin{pmatrix} \mathbf{B} \\ -\mathbf{C}^* \\ \mathbf{0} \end{pmatrix} \Delta,$$

$$\mathbf{C}_s = -\Delta \begin{pmatrix} \mathbf{C} & \mathbf{B}^* & \mathbf{0} \end{pmatrix}, \; \mathbf{D}_s = \Delta = (\mathbf{D}+\mathbf{D}^*)^{-1} \tag{4.57}$$

   b. Σ_s when $\mathbf{D} = \mathbf{0}$:

$$\mathbf{A}_s = \begin{pmatrix} \mathbf{A} & \mathbf{0} & \mathbf{B} \\ \mathbf{0} & -\mathbf{A}^* & -\mathbf{C}^* \\ \mathbf{C} & \mathbf{B}^* & \mathbf{0} \end{pmatrix}, \; \mathbf{E}_s = \begin{pmatrix} \mathbf{E} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{E}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}, \; \mathbf{B}_s = \begin{pmatrix} \mathbf{B} \\ -\mathbf{C}^* \\ \mathbf{I} \end{pmatrix}, \; \mathbf{C}_s = -\begin{pmatrix} \mathbf{C} & \mathbf{B}^* & \mathbf{I} \end{pmatrix} \tag{4.58}$$

2. Solve the Hamiltonian eigenvalue problem $(\Lambda, \mathbf{R}, \mathbf{L}) = \mathrm{eig}(\mathbf{A}_s, \mathbf{E}_s)$, i.e., $\mathbf{A}_s\mathbf{R} = \mathbf{E}_s\mathbf{R}\Lambda$, $\mathbf{L}^*\mathbf{A}_s = \Lambda\mathbf{L}^*\mathbf{E}_s$. $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_{2n}]$, $\mathbf{L} = [\mathbf{l}_1, \ldots, \mathbf{l}_{2n}]$ and eigenvalues $\Lambda = \mathrm{diag}(s_1, \ldots, s_n, -s_1^*, \ldots, -s_n^*)$ are the spectral zeros of Σ.
3. Compute residues $\mathscr{R}_j$ associated with the stable[22] spectral zeros s_j, $j = 1 \ldots n$ as follows: $\mathscr{R}_j = \gamma_j \beta_j$, $\gamma_j = \mathbf{C}_s \mathbf{r}_j (\mathbf{l}_j^* \mathbf{E}_s \mathbf{r}_j)^{-1}$, $\beta_j = \mathbf{l}_j^* \mathbf{B}_s$.
4. Sort spectral zeros descendingly according to *dominance criterion* $\frac{\|\mathscr{R}_j\|}{|Re(s_j)|}$ [130, Chapter 3], and reorder right eigenvectors $\mathbf{R}$ accordingly.
5. Retain the right eigenspace $\hat{\mathbf{R}} = [\mathbf{r}_1, \; \ldots, \; \mathbf{r}_k] \in \mathbb{C}^{2n \times k}$, corresponding to the stable k most dominant spectral zeros.
6. Construct passivity projection matrices $\mathbf{V}$ and $\mathbf{W}$ from the rows of $\hat{\mathbf{R}}$: $\mathbf{V} = \hat{\mathbf{R}}_{[1:n,1:k]}$, $\mathbf{W} = \hat{\mathbf{R}}_{[n+1:2n,1:k]}$, and reduce Σ according to (4.55).

As explained in [114, 125, 134], by projecting with (4.55), $\hat{\Sigma}$ interpolates the k most dominant spectral zeros of Σ, guaranteeing passivity and stability. For large-scale applications, a full solution to the eigenvalue problem in step 2, followed by the dominant sort 3–4 is computationally unfeasible. Instead, the iterative solver SADPA [130, Chapter 3] is applied with appropriate adaptations for spectral zero computation (see Appendix 4.4.6 for the pseudocode). SADPA implements steps 2–4 efficiently and automatically gives the k most dominant spectral zeros and associated $2n \times k$ right eigenspace $\hat{\mathbf{R}}$. The implementation requires performing an LU factorization of $(s_j\mathbf{E} - \mathbf{A})$ at each iteration. The relevant s_j are nevertheless computed automatically in SADPA, which may have several advantages over other methods (see [125] for a more detailed cost analysis).

[22] $s \in \mathbb{C}$ is stable if $Re(s) < 0$.

4.4.3.2 $\mathbf{D} = \mathbf{0}$ and Dominance at $s \to \infty$

Systems arising in circuit simulation often satisfy $\mathbf{D} = \mathbf{0}$ in (4.54). In this case, the projection (4.55), with $\mathbf{W}$ and $\mathbf{V}$ obtained in step 6 in Sect. 4.4.3.1, gives a lossless system [125]. This is because $\mathbf{W}$ and $\mathbf{V}$ only interpolate dominant finite spectral zeros, whereas the original system has spectral zeros at ∞, some of which may be dominant [120]. A strictly passive system (with all poles in the left half plane) can nevertheless be obtained by recovering this dominant behavior. For systems often occurring in circuit simulation this is achieved as follows. Consider the modified nodal analysis (MNA) description of an RLC circuit:

$$\underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathscr{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathscr{L} \end{pmatrix}}_{\mathbf{E}} \underbrace{\frac{d}{dt} \begin{pmatrix} v_p \\ \boldsymbol{v}_i \\ \boldsymbol{i}_L \end{pmatrix}}_{\dot{\mathbf{x}}} + \underbrace{\begin{pmatrix} \mathscr{G}_{11} & \mathscr{G}_{12} & \mathscr{E}_1 \\ \mathscr{G}_{12}^* & \mathscr{G}_{22} & \mathscr{E}_2 \\ -\mathscr{E}_1^* & -\mathscr{E}_2^* & \mathbf{0} \end{pmatrix}}_{-\mathbf{A}} \underbrace{\begin{pmatrix} \boldsymbol{v}_p \\ \boldsymbol{v}_i \\ \boldsymbol{i}_L \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} \mathscr{B}_1 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{B}} \mathbf{u}, \qquad (4.59)$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ are input currents and $\mathbf{y}(t) = \mathbf{Cx} \in \mathbb{R}^m$ are output voltages, $\mathbf{C} = \mathbf{B}^*$. The states are $\mathbf{x}(t) = [\mathbf{v}_p(t),\ \mathbf{v}_i(t),\ \mathbf{i}_L(t)]^T$, with $\mathbf{v}_p(t) \in \mathbb{R}^{n_p}$ the voltages at the input nodes (circuit terminals), $\mathbf{v}_i(t) \in \mathbb{R}^{n_i}$ the voltages at the internal nodes, and $\mathbf{i}_L(t) \in \mathbb{R}^{n_{i_L}}$ the currents through the inductors, $n_p + n_i + n_{i_L} = n$. $\mathscr{C}$ and $\mathscr{L}$ are the capacitor and inductor matrix stamps, respectively. With (4.59) it is assumed that no capacitors or inductors are directly connected to the input nodes, thus $\mathbf{B} \in Null(\mathbf{E})$ and $\mathbf{C}^* \in Null(\mathbf{E}^*)$. As $\mathbf{B}$ and $\mathbf{C}$ are right and left eigenvectors corresponding to dominant poles (and spectral zeros) at ∞ [120], the modified projection matrices are:

$$\tilde{\mathbf{W}} = [\mathbf{W}, \mathbf{C}^*],\ \tilde{\mathbf{V}} = [\mathbf{W}, \mathbf{B}], \qquad (4.60)$$

where $\mathbf{W}$ and $\mathbf{V}$ are obtained from step 6 in Sect. 4.4.3.1. With (4.60), the finite dominant spectral zeros are interpolated as well as the dominant spectral zeros at ∞, and the reduced system is strictly passive [120]. In [125] two alternatives were proposed for ensuring strict passivity for systems in the more general form (4.54) with $\mathbf{D} = \mathbf{0}$.

4.4.3.3 Circuit Representation of Reduced Impedance Transfer Function

Reduced models obtained with dominant SZM and other Krylov-type methods (PRIMA [128], SPRIM [117, 118], SPRIM/IOPOR [115, 138]) are mathematical abstractions of an underlying small RLC circuit. Circuit simulators however can only handle mathematical representations to a limited extent, and reduced models have to be synthesized with RLC circuit elements. We reduce all circuits with respect to the input impedance transfer function (i.e., the inputs are the currents injected into the circuit terminals and the outputs are the voltages measured at the terminals) [123]. After converting the reduced input impedance transfer function to netlist format, the reduced circuit can be driven easily by currents or voltages when simulated. Thus both the input impedance and admittance of an original model can be reproduced (see Sect. 4.4.4). Here, models obtained with dominant SZM are converted to netlist representations using the Foster impedance realization approach [119, 122]. Netlist formats for the SPRIM/IOPOR [115, 117, 138] reduced models are obtained via the RLCSYN unstamping procedure in [123, 138]. With both approaches, the resulting netlists may still contain circuit elements with negative values, nevertheless this does not impede the circuit simulation. Obtaining realistic synthesized models with positive circuit elements only is still an open problem.

4.4.4 Numerical Results

Two transmission line models are reduced with the proposed dominant spectral zero method and compared with the input-output structure preserving method SPRIM/IOPOR [115, 117, 138]. For both circuits, the circuit simulators[23] yield systems in the form (4.59), thus the dominant SZM projection is (4.60). RLC netlist representations for the reduced models are obtained (see Sect. 4.4.3.3) and simulated with Pstar.

The RLC transmission line with connected voltage controlled current sources (VCCSs) from [125] is reduced with dominant SZM, SPRIM/IOPOR [117, 138] and modal approximation (MA). The transfer function is an input impedance i.e., the circuit is current driven. Matlab simulations of the original and reduced models, as well as the Pstar netlist simulations are shown in Fig. 4.19: the model reduced with Dominant SZM gives the best approximation. Table 4.2 summarizes the reduction: the number of circuit elements and the number of states were reduced significantly and the simulation time was sped up.

In [125], the voltage driven *input admittance* of an RLC transmission line (consisting of cascaded RLC blocks) was reduced directly as shown in Fig. 4.20. Here we reduce and synthesize the underlying *input impedance* of the same transmission line (see Figs. 4.21 and 4.22). When driving the reduced netlist by an input voltage during the actual circuit simulation, the same input admittance is obtained as if the input admittance had been reduced directly, as seen in Figs. 4.20 and 4.23. Table 4.3 summarizes the reduction results. Although the reduced mathematical

[23] Pstar and Hstar are in-house simulators at NXP Semiconductors, Eindhoven, The Netherlands

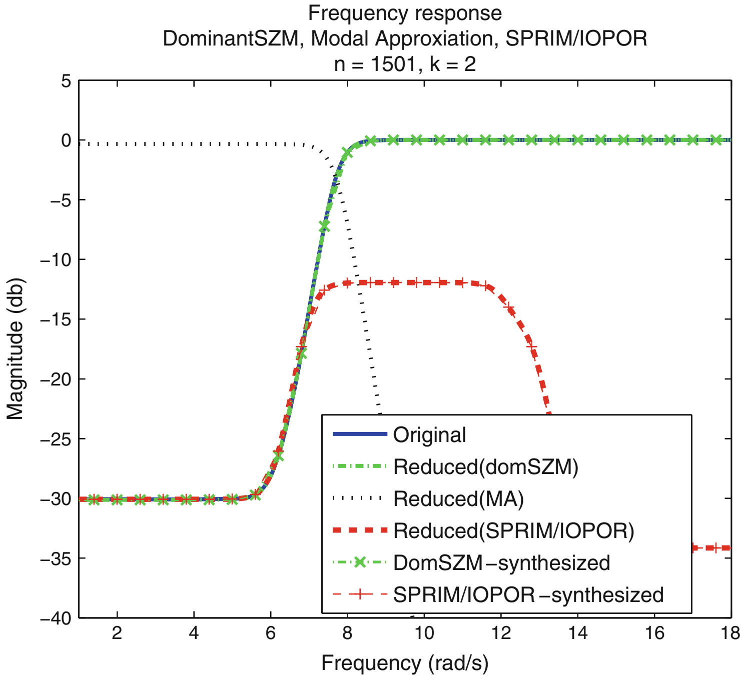


Fig. 4.19 Original, reduced and synthesized systems: Dominant SZM, SPRIM/IOPOR

Table 4.2 Transmission line with VCCSs: reduction and synthesis summary

System	Dimension	R	C	L	VCCs	States	Simulation time
Original	1501	1001	500	500	500	1,500	0.5 s
Dominant SZM	2	3	2	0	–	4	0.01 s
SPRIM/IOPOR	2	6	3	1	–	4	0.01 s

models have the same dimension ($k = 23$), the reduction effect can only be determined after obtaining the netlist representations. Although the SPRIM/IOPOR synthesized model has fewer states, it has more circuit elements than the dominant SZM model, i.e., the matrix stamp of the model is more dense. This suggests that simulation time is jointly determined by the number of states and the number of circuit elements. Thus for practical purposes it is critical to synthesize reduced models with RLC components.

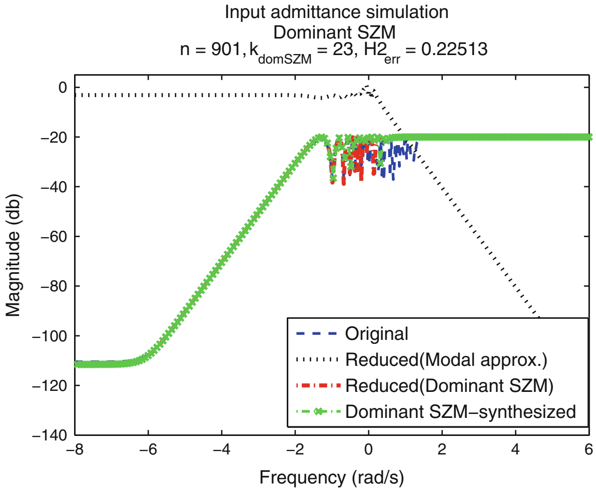


Fig. 4.20 Input admittance transfer function: original, synthesized Dominant SZM model

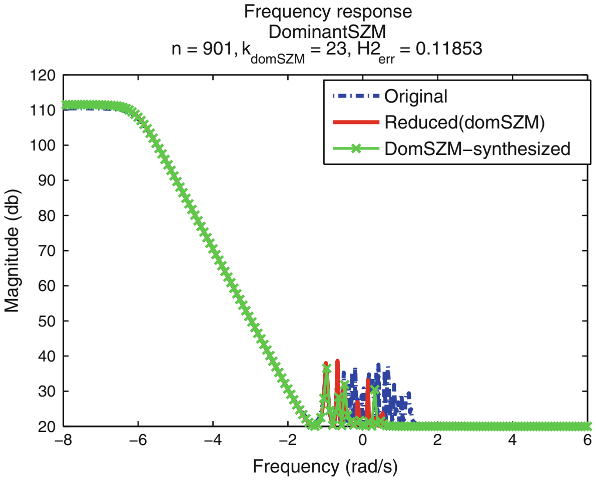


Fig. 4.21 Input impedance transfer function: original and reduced with Dominant SZM

4.4.5 Concluding Remarks

A novel passivity preserving model reduction method is presented, which is based on interpolation of dominant spectral zeros. Implemented with the SADPA eigenvalue solver, the method computes the partial eigenvalue decomposition of an

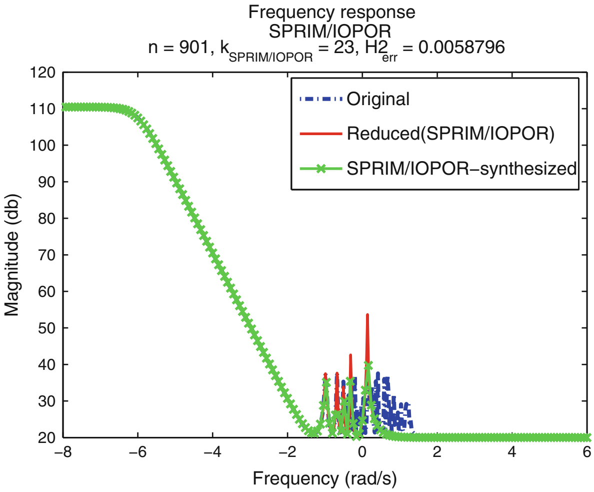


Fig. 4.22 Input impedance transfer function: original, reduced with SPRIM/IOPOR

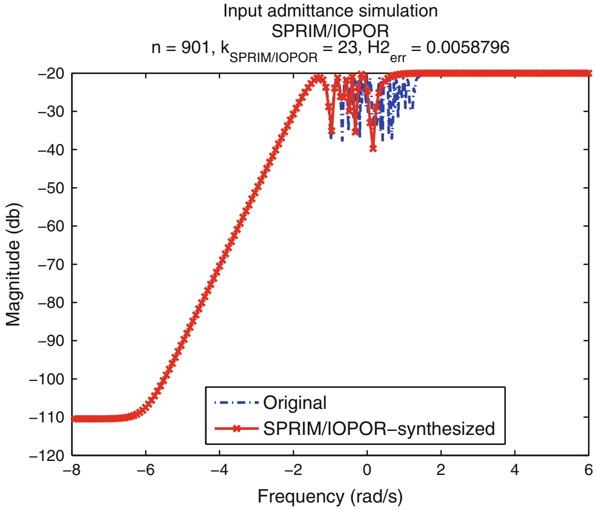


Fig. 4.23 Input admittance transfer function: original, synthesized SPRIM/IOPOR model

associated Hamiltonian matrix pair, and constructs the passivity preserving projection. Netlist equivalents for the reduced models are simulated and directions for future work are revealed. Especially in model reduction of multi-terminal circuits, achieving structure preservation, sparsity and small dimensionality simultaneously

Table 4.3 RLC transmission line: Input impedance reduction and synthesis summary

System	Dimension	R	C	L	States	Simulation time
Original	901	500	300	300	901	1.5 s
Dominant SZM	23	22	11	10	34	0.02 s
SPRIM/IOPOR	23	78	66	6	18	0.02 s

is an open question. New developments on sparsity-preserving model reduction for multi-terminal *RC* circuits can be found in [124]. In this context, RLC synthesis with positive circuit elements will also be addressed.

4.4.6 Appendix: SADPA for Computing Dominant Spectral Zeros

We outline SADPA for SISO systems; the MIMO implementation is similar and the code for computing dominant poles can be found in [132] or online [130]. The following pseudocode is extracted from [130, Chapter 3] and [131], with efficient modifications to automatically account for the four-fold symmetry $(\lambda, -\lambda^*, \lambda^*, -\lambda)$ of spectral zeros. In particular, as soon as a Hamiltonian eigenvalue (spectral zero) λ has converged, we immediately deflate the right/left eigenvectors corresponding to $-\lambda^*$ as well. It turns out that the right/left eigenvectors corresponding to $-\lambda^*$ need not be solved for explicitly. Rather, due to the structure of the Hamiltonian matrices [127, 137], they can be written down directly from the already converged left/right eigenvectors for λ, as shown in steps 14–17 of Algorithm 4.9. As for modal approximation [131], [130, Chapter 3] deflation for λ^* and $-\lambda$ is automatically handled in Algorithm 4.11. To summarize, once the right/left eigenvectors corresponding to an eigenvalue λ have converged, the right/left eigenvectors corresponding to $-\lambda^*, \lambda^*, -\lambda$ are also readily available at no additional computational cost, and can be immediately deflated.

In Algorithm 4.10, the MATLAB `qz` routine is proposed for solving the small, projected eigenvalue problem in step 1. This reveals the right/left eigenvectors $\tilde{\mathbf{X}}, \tilde{\mathbf{V}}$ of the projected pencil directly, however they are neither orthogonal nor bi-$\mathbf{G}$-orthogonal. Thus the normalization in step 3 is needed when computing the residues.

A modified Gram-Schimdt procedure (MGS) is used for orthonormalization. We used the implementation in [130, Algorithm 1.4]. For complete mathematical and algorithmic details of SADPA we refer to [130, Chapter 3] and [131].

Algorithm 4.9 $(\Lambda, \mathbf{R}, \mathbf{L})$ =SADPA$(\mathbf{E}_h, \mathbf{A}_h, \mathbf{B}_h, \mathbf{C}_h, s_1, \ldots p_{max}, k_{min}, k_{max})$

INPUT: $(\mathbf{E}_h, \mathbf{A}_h, \mathbf{B}_h, \mathbf{C}_h)$, $\mathbf{E}_h \in \mathbb{C}^{2n\times 2n}$, $\mathbf{A}_h \in \mathbb{C}^{2n\times 2n}$, $\mathbf{B}_h \in \mathbb{C}^{2n\times 1}$, $\mathbf{C}_h \in \mathbb{C}^{1\times 2n}$ an initial pole estimate s_1 and number of desired poles p_{max} (in the restarted version, k_{min} and k_{max} are also specified)

OUTPUT: Λ, the p_{max} most dominant eigenvalues and associated right, left eigenspaces **R**, **L** of $(\mathbf{A}_h, \mathbf{E}_h)$

```
$k = 1,\ p_{found} = 0,\ \Lambda = [],\ \mathbf{R} = [],\ \mathbf{L} = []$
while $p_{found} < p_{max}$ do
    Solve for $\mathbf{x}$ from $(s_k\mathbf{E}_h - \mathbf{A}_h)\mathbf{x} = \mathbf{B}_h$
    Solve for $\mathbf{v}$ from $(s_k\mathbf{E}_h - \mathbf{A}_h)^*\mathbf{v} = \mathbf{C}_h^*$
    $\mathbf{x}$ =MGS$(\mathbf{X}, \mathbf{x})$, $\mathbf{X} = [\mathbf{X}, \mathbf{x}/\|\mathbf{x}\|]$
    $\mathbf{v}$ =MGS$(\mathbf{V}, \mathbf{v})$, $\mathbf{V} = [\mathbf{V}, \mathbf{v}/\|\mathbf{v}\|]$
    Compute $\mathbf{G} = \mathbf{V}^*\mathbf{E}_h\mathbf{X}$ and $\mathbf{T} = \mathbf{V}^*\mathbf{A}_h\mathbf{X}$
    $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ = DomSort$(\mathbf{T}, \mathbf{G}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$    ▷ ▷ Algorithm 4.10
    Compute dominant approximate eigentriplet $(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{v}}_1)$:
        $\hat{\lambda}_1 = \tilde{\lambda}_1, \hat{\mathbf{x}}_1 = (\mathbf{X}\tilde{\mathbf{x}}_1)/\|\mathbf{X}\tilde{\mathbf{x}}_1\|, \hat{\mathbf{v}}_1 = (\mathbf{V}\tilde{\mathbf{v}}_1)/\|\mathbf{V}\tilde{\mathbf{v}}_1\|$
    if $\|\mathbf{A}_h\hat{\mathbf{x}}_1 - \mathbf{E}_h\hat{\mathbf{x}}_1\hat{\lambda}_1\| < \epsilon$ then
        $(\Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$ = Deflate$(\hat{\lambda}_1, \hat{\mathbf{x}}_1, \hat{\mathbf{v}}_1, \Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}\tilde{\mathbf{X}}_{(:,2:k)}, \mathbf{V}\tilde{\mathbf{V}}_{(:,2:k)}, \mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h)$
                                                            ▷ ▷ Algorithm 4.11
        $p_{found}++$ ▷ ▷ Also find eigenvectors for the antistable spectral zero $-\hat{\lambda}_1^*$ and deflate
        $\mathbf{x} = [\ -\hat{\mathbf{v}}_{1(n+1:2n,:)};\ \hat{\mathbf{v}}_{1(1:n,:)}\ ]$
        $\mathbf{v} = [\ \hat{\mathbf{x}}_{1(n+1:2n,:)};\ -\hat{\mathbf{x}}_{1(1:n,:)}\ ]$
        $(\Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$ = Deflate$(-\hat{\lambda}_1^*, \mathbf{x}, \mathbf{v}, \Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h)$    ▷ ▷ Algorithm 4.11
        $p_{found}++$
        $\tilde{\lambda}_1 = \tilde{\lambda}_2$
    else if ncols$(\tilde{\mathbf{X}}) > k_{max}$ then
                                                            ▷ ▷ Possible restart
        ▷ ▷ Retain first $k_{min}$ most dominant approximate eigenvectors and re-orthonormalize
        $\mathbf{X}$ =MGS$(\mathbf{X}\tilde{\mathbf{X}}_{(:,1:k_{min})})$    ▷ ▷ Orthornormalize all columns sequentially
        $\mathbf{V}$ =MGS$(\mathbf{V}\tilde{\mathbf{V}}_{(:,1:k_{min})})$
    end if
    Increment $k = k + 1$
    Select new most dominant pole estimate $s_k = \tilde{\lambda}_1$
end while
```

Algorithm 4.10 $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ = DomSort$(\mathbf{T}, \mathbf{G}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h)$

INPUT: $(\mathbf{T}, \mathbf{G}), \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h$

OUTPUT: $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$, k dominant approximate eigenvalues and associated right, left eigenvectors of $(\mathbf{T}, \mathbf{G})$, sorted such that $\tilde{\lambda}_1$ is most dominant

```
$(AA, BB, Q, Z, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ = QZ$(\mathbf{T}, \mathbf{G})$
$\tilde{\Lambda}$ = diag$(AA)$./diag$(BB)$ and $|\tilde{\lambda}_i| \neq \infty, i = 1 \ldots k$
$R_i = \frac{[\mathbf{C}_h\tilde{x}_i][\tilde{v}_i^*\mathbf{B}_h]}{\tilde{v}_i^*\mathbf{G}\tilde{x}_i}$    ▷ ▷ Compute residues
Sort $(\tilde{\Lambda}, \tilde{\mathbf{X}}, \tilde{\mathbf{V}})$ in decreasing $|R_i|/|Re(\tilde{\lambda}_i)|$ order
```

Algorithm 4.11 $(\Lambda, \mathbf{R}, \mathbf{L}, \mathbf{X}, \mathbf{V}, \mathbf{B}_h, \mathbf{C}_h) = \text{Deflate}(\hat{\lambda}, \hat{\mathbf{x}}, \hat{\mathbf{v}}, \ldots \Lambda, \mathbf{R}, \mathbf{L}, \hat{\mathbf{X}}, \hat{\mathbf{V}}, \mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h)$

INPUT: $(\hat{\lambda}, \hat{\mathbf{x}}, \hat{\mathbf{v}})$: the newly converged most dominant eigentriplet, $(\Lambda, \mathbf{R}, \mathbf{L})$: the dominant eigentriplets already found correctly, $\hat{\mathbf{X}}, \hat{\mathbf{V}}$: the approximate right/left eigenvectors not yet checked for convergence, $\mathbf{E}_h, \mathbf{B}_h, \mathbf{C}_h$

OUTPUT: $(\Lambda, \mathbf{R}, \mathbf{L})$: updated converged eigentriplets, $\mathbf{X}, \mathbf{V}$: deflated approximate eigenspaces, $\mathbf{B}_h, \mathbf{C}_h$: deflated matrices

```
$\Lambda = [\Lambda, \hat{\lambda}]$
$\hat{\mathbf{r}} = \hat{\mathbf{x}}/(\hat{\mathbf{v}}^* \mathbf{E}_h \hat{\mathbf{x}})$        ▷ ▷ For keeping converged eigenvectors bi-E-orthogonal
$\hat{\mathbf{l}} = \hat{\mathbf{v}}$
$\mathbf{R} = [\mathbf{R}, \hat{\mathbf{r}}], \mathbf{L} = [\mathbf{L}, \hat{\mathbf{l}}]$
Deflate $\mathbf{B}_h = \mathbf{B}_h - \mathbf{E}_h \hat{\mathbf{r}} (\hat{\mathbf{l}}^* \mathbf{B}_h)$
Deflate $\mathbf{C}_h = \mathbf{C}_h - (\mathbf{C}_h \hat{\mathbf{r}}) \hat{\mathbf{l}}^* \mathbf{E}_h$
if imag($\hat{\lambda} \neq 0$) then
                                                                ▷ ▷ Also deflate complex conjugate
    $\Lambda = [\Lambda, \hat{\lambda}^*]$
    $\hat{\mathbf{r}} = \hat{\mathbf{r}}^*, \hat{\mathbf{l}} = \hat{\mathbf{l}}^*$
    $\mathbf{R} = [\mathbf{R}, \hat{\mathbf{r}}], \mathbf{L} = [\mathbf{L}, \hat{\mathbf{l}}]$
    Deflate $\mathbf{B}_h = \mathbf{B}_h - \mathbf{E}_h \hat{\mathbf{r}} (\hat{\mathbf{l}}^* \mathbf{B}_h)$
    Deflate $\mathbf{C}_h = \mathbf{C}_h - (\mathbf{C}_h \hat{\mathbf{r}}) \hat{\mathbf{l}}^* \mathbf{E}_h$
end if
$\mathbf{X} = \mathbf{Y} = []$
for $j = 1 \ldots \#\text{cols}(\hat{\mathbf{X}})$ do
    $\mathbf{X} = \text{Expand}(\mathbf{X}, \mathbf{R}, \mathbf{L}, \mathbf{E}_h, \hat{\mathbf{x}}_j)$        ▷ ▷ Algorithm 4.12
    $\mathbf{V} = \text{Expand}(\mathbf{V}, \mathbf{R}, \mathbf{L}, \mathbf{E}_h^*, \hat{\mathbf{v}}_j)$        ▷ ▷ Algorithm 4.12
end for
```

Algorithm 4.12 $\mathbf{X} = \text{Expand}(\mathbf{X}, \mathbf{R}, \mathbf{L}, \mathbf{E}_h, \hat{\mathbf{x}})$

INPUT: $\mathbf{X} \in \mathbb{C}^{2n \times k}$ such that $\mathbf{X}\mathbf{X}^* = \mathbf{I}$, $(\mathbf{R}, \mathbf{L}) \in \mathbb{C}^{2n \times p}$: the correctly found right/left eigenvectors such that: $\mathbf{L}^* \mathbf{E}_h \mathbf{R}$ is diagonal and $\mathbf{L}^* \mathbf{E}_h \mathbf{X} = \mathbf{0}$, $\hat{\mathbf{x}}$: approximate eigenvector not yet checked for convergence, $\mathbf{E}_h$

OUTPUT: $\mathbf{X} \in \mathbb{C}^{2n \times (k+1)}$ expanded such that $\mathbf{X}\mathbf{X}^* = \mathbf{I}$

```
$\mathbf{x}_{k+1} = \prod_{j=1}^{p} \left( \mathbf{I} - \frac{\mathbf{r}_j \mathbf{l}_j^* \mathbf{E}_h}{\mathbf{l}_j^* \mathbf{E}_h \mathbf{r}_j} \right) \hat{\mathbf{x}}$
$\mathbf{x} = \text{MGS}(\mathbf{X}, \mathbf{x}_{k+1})$
$\mathbf{X} = [\mathbf{X}, \mathbf{x}/\|\mathbf{x}\|]$
```

4.5 A Framework for Synthesis of Reduced Order Models

The main motivation for this section comes from the need for a general framework for the (re)use of reduced order models in circuit simulation.[24] Although many model order reduction methods have been developed and evolved since the 1990s (see for instance [141, 146] for an overview), it is usually less clear how to use these

[24] Section 4.5 has been written by: Roxana Ionutiu and Joost Rommes. For an extended treatment on the topics of this section see also the Ph.D.-Thesis of the first author [156] and [159].

methods efficiently in industrial practice, e.g., in a circuit simulator. One reason can be that the reduced order model does not satisfy certain physical properties, for instance, it may not be stable or passive while the original system is. Failing to preserve these properties is typically inherent to the reduced order method used (or its implementation). Passivity (and stability implicitly) can nowadays be preserved via several methods [142, 151, 157, 166, 169, 173, 174], but none address the practical aspect of (re)using the reduced order models with circuit simulation software (e.g., SPICE [150]). While the original system is available in *netlist* format, the reduced order model is in general only available in *numerical* format. Typically, circuit simulators are not prepared for inputs of this form and would require additional software architecture to handle them. In contrast, a reduced model in *netlist* representation could be easily coupled to bigger systems and simulated.

Synthesis is the realization step needed to map the reduced order model into a netlist consisting of electrical circuit components [154, 170]. In [148] it was shown that passive systems (with positive real transfer functions) can be synthesized with positive R,L,C elements and transformers. Later developments [147] propose a method to circumvent the introduction of transformers, however the resulting realization is non-minimal (i.e., the number of electrical components generated during synthesis is too large). Allowing for possibly negative R, L, C values, other methods have been proposed via e.g. direct stamping [163, 166] or full realization [155, 167]. These mostly model the input/output connections of the reduced model with controlled sources.

In this section we consider two synthesis methods that do not involve controlled sources: (1) *Foster synthesis* [154], where the realization is done via the system's transfer function and (2) *RLCYSN synthesis by unstamping* [176], which exploits input-output structure preservation in the reduced system matrices [provided that the original system matrices are written in *modified nodal analysis (MNA)* representation]. The focus of this section is on structure preservation and RLCSYN, especially because synthesis by unstamping is simple to implement for both SISO and MIMO systems. Strengthening the result of [176], we give a simple procedure to reduce either current- or voltage-driven circuits directly in impedance form by removing all the sources. Such an impedance-based reduction enables synthesis without controlled sources. The reduced order model is available as a netlist, making it suitable for simulation and reuse in other designs. Similar software [149] is commercially available.

The material in this section is organized as follows. A brief mathematical formulation of model order reduction is given in Sect. 4.5.1. The Foster synthesis is presented in Sect. 4.5.2. In Sect. 4.5.3 we focus on reduction and synthesis with structure (and input/output) preservation. Section 4.5.3.1 describes the procedure to convert admittance models to impedance form, so that synthesized models are easily (re)used in simulation. Based on [176], Sect. 4.5.3.2 is an outline of SPRIM/IOPOR reduction and RLCSYN synthesis. Examples follow in Sect. 4.5.4, and Sect. 4.5.5 concludes.

4.5.1 Problem Formulation

In this section the dynamical systems $\Sigma(\mathbf{A}, \mathbf{E}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are of the form $\mathbf{E}\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$, where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n\times n}$, $\mathbf{E}$ may be singular but the pencil $(\mathbf{A}, \mathbf{E})$ is regular, $\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{p\times n}$, $\mathbf{x}(t) \in \mathbb{R}^n$, and $\mathbf{u}(t) \in \mathbb{R}^m$, $\mathbf{y}(t) \in \mathbb{R}^p$, $\mathbf{D} \in \mathbb{R}^{p\times m}$. If $m, p > 1$, the system is called multiple-input multiple-output (MIMO), otherwise it is called single-input single-output (SISO). The frequency domain transfer function is defined as: $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$. For systems in MNA form arising in circuit simulation see Sect. 4.5.3.

The model order reduction problem is to find, given an n-th order (descriptor) dynamical system, a k-th order system: $\tilde{\mathbf{E}}\dot{\tilde{\mathbf{x}}}(t) = \tilde{\mathbf{A}}\tilde{\mathbf{x}}(t) + \tilde{\mathbf{B}}\mathbf{u}(t)$, $\tilde{\mathbf{y}}(t) = \tilde{\mathbf{C}}\tilde{\mathbf{x}}(t) + \mathbf{D}\mathbf{u}(t)$ where $k < n$, and $\tilde{\mathbf{E}}, \tilde{\mathbf{A}} \in \mathbb{R}^{k\times k}$, $\tilde{\mathbf{B}} \in \mathbb{R}^{k\times m}$, $\tilde{\mathbf{C}} \in \mathbb{R}^{p\times k}$, $\tilde{\mathbf{x}}(t) \in \mathbb{R}^k$, $\mathbf{u}(t) \in \mathbb{R}^m$, $\tilde{\mathbf{y}}(t) \in \mathbb{R}^p$, and $\mathbf{D} \in \mathbb{R}^{p\times m}$. The number of inputs and outputs is the same as for the original system, and the corresponding transfer function becomes: $\tilde{\mathbf{H}}(s) = \tilde{\mathbf{C}}(s\tilde{\mathbf{E}} - \tilde{\mathbf{A}})^{-1}\tilde{\mathbf{B}} + \mathbf{D}$. For an overview of model order reduction methods, see [141, 145, 146, 172]. Projection based model order reduction methods construct a reduced order model via Petrov-Galerkin projection:

$$\tilde{\Sigma}(\tilde{\mathbf{E}}, \tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}, \mathbf{D}) \equiv (\mathbf{W}^*\mathbf{E}\mathbf{V}, \mathbf{W}^*\mathbf{A}\mathbf{V}, \mathbf{W}^*\mathbf{B}, \mathbf{V}^*\mathbf{C}, \mathbf{D}), \tag{4.61}$$

where $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n\times k}$ are matrices whose $k < n$ columns form bases for relevant subspaces of the state-space. There are several projection methods, that differ in the way the matrices $\mathbf{V}$ and $\mathbf{W}$ are chosen. These also determine which properties are preserved after reduction. Some stability preserving methods are: *modal approximation* [171], *Poor Man's TBR* [168]. Among *moment matching* [152] methods, the following preserve passivity: *PRIMA* [166], *SPRIM* [151], *spectral zero interpolation*, [142, 157, 161, 173]. From the balancing methods, *balanced truncation* [144] preserves stability, and *positive real balanced truncation* [169, 174] preserves passivity.

4.5.2 Foster Synthesis of Rational Transfer Functions

This section describes the Foster synthesis method, which was developed in the 1930s by Foster and Cauer [154] and involves realization based on the system's transfer function. The Foster approach can be used to realize any reduced order model that is computed by standard projection based model order reduction techniques. Realizations will be described in terms of SISO impedances (Z-parameters). For equivalent realizations in terms of admittances (Y-parameters), see for instance [154, 175]. Given the reduced system (4.61) consider the partial fraction expansion [162] of its transfer function:

$$\tilde{\mathbf{H}}(s) = \sum_{i=1}^{k} \frac{\tilde{r}_i}{s - \tilde{p}_i} + \mathbf{D}, \tag{4.62}$$

The residues are $\tilde{r}_i = \frac{(\tilde{\mathbf{C}}\tilde{\mathbf{x}}_i)(\tilde{\mathbf{y}}_i^*\tilde{\mathbf{B}})}{\tilde{\mathbf{y}}_i^*\tilde{\mathbf{E}}\tilde{\mathbf{x}}_i}$ and the poles are $\tilde{p}_i$. An eigentriplet $(\tilde{p}_i, \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ is composed of an eigenvalue $\tilde{p}_i$ of $(\tilde{\mathbf{A}}, \tilde{\mathbf{E}})$ and the corresponding right and left eigenvectors $\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i \in \mathbb{C}^k$. The expansion (4.62) consists of basic summands of the form:

$$Z(s) = r_1 + \frac{r_2}{s - p_2} + \frac{r_3}{s} + \left(\frac{r_4}{s - p_4} + \frac{\bar{r}_4}{s - \bar{p}_4}\right) + s r_6 + \left(\frac{r_7}{s - p_7} + \frac{r_7}{s - \bar{p}_7}\right), \tag{4.63}$$

where for completeness we can assume that any kind of poles may appear, i.e., either purely real, purely imaginary, in complex conjugate pairs, at ∞ or at 0 (see also Table 4.4). The Foster realization converts each term in (4.63) into the corresponding circuit block with R, L, C components, and connects these blocks in series in the final netlist. This is shown in Fig. 4.24. Note that any reordering of the circuit blocks in the realization of (4.63) in Fig. 4.24 still is a realization of (4.63). The values for the circuit components in Fig. 4.24 are determined according to Table 4.4.

The realization in netlist form can be implemented in any language such as SPICE [150], so that it can be reused and combined with other circuits as well. The advantages of Foster synthesis are: (1) its straightforward implementation for single-input-single-output (SISO) transfer functions, via either the impedance or the admittance transfer function, (2) for purely RC or RL circuits, netlists obtained from reduction via modal approximation [171] are guaranteed to have positive

Table 4.4 Circuit element values for Fig. 4.24 for the Foster impedance realization of (4.63)

Pole	Residue	R(Ohm)	C(F)	L(H)	G(Ohm^{-}1)
$p_1 = \infty$	$r_1 \in \mathbb{R}$	r_1			
$p_2 \in \mathbb{R}$	$r_2 \in \mathbb{R}$	$-\frac{r_2}{p_2}$	$\frac{1}{r_2}$		
$p_3 = 0$	$r_3 \in \mathbb{R}$		$\frac{1}{r_3}$		
$p_4 = \sigma + i\omega \in \mathbb{C}$ $p_5 \equiv \bar{p}_4$	$r_4 = \alpha + i\beta \in \mathbb{C}$ $r_5 \equiv \bar{r}_4$	$\frac{a_0}{a_1} L$	$\frac{1}{a_1}$	$\frac{a_1^3}{a_1^2 b_0 - a_0(a_1 b_1 - a_0)}$	$\frac{a_1 b_1 - a_0}{a_1^2}$
$a_0 = -2(\alpha\sigma + \beta\omega), \quad a_1 = 2\alpha, \quad b_0 = \sigma^2 + \omega^2, \quad b_1 = -2\sigma$					
$p_6 = \infty$	$r_6 \in \mathbb{R}$			r_6	
$p_7 \in i\mathbb{R}$ $p_8 \equiv \bar{p}_7$	$r_7 \in \mathbb{R}$ $r_8 \equiv \bar{r}_7$		$\frac{1}{r_7}$	$\frac{2r_7}{p_7 \bar{p}_7}$	

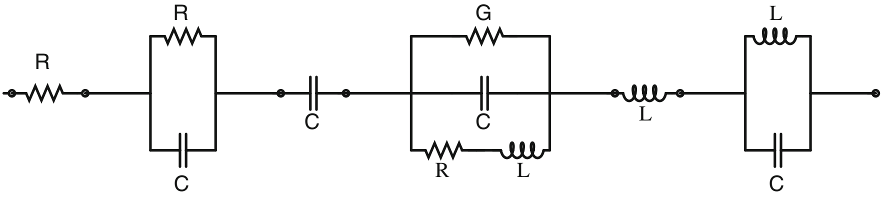


Fig. 4.24 Realization of a general impedance transfer function as a series *RLC* circuit

RC or *RL* values respectively [158]. The main disadvantage is that for multi-input-multi-output transfer functions, finding the Foster realization (see for instance [175]) is cumbersome and may also give dense reduced netlists (i.e., all nodes are interconnected). This is because the Foster synthesis of a k-dimensional reduced system with p terminals, will generally yield $O(p^2k)$ circuit elements. A method based on partitioning of an RLC circuit is found in [164]. The method produces a positive-valued, passive and stable reduced RLC circuit.

4.5.3 *Structure Preservation and Synthesis by Unstamping*

This section describes the second synthesis approach, which is based on *unstamping* the reduced matrix data into an *RLC* netlist and is denoted by RLCSYN [176]. It is suitable for obtaining netlist representations for models reduced via methods that preserve the MNA structure and the circuit terminals, such as the *input-output structure preserving* method SPRIM/IOPOR [176]. The strength of the result in [176] is that the input/output connectivity is synthesized after reduction without controlled sources, provided that the system is in *impedance form* (i.e., inputs are currents injected into the circuit terminals, and outputs are voltages measured at the terminals). Here, we interpret the input-output preservation as preserving the external nodes[25] of the original model during reduction. This way the reduced netlist can easily be coupled to other circuitry in place of the original netlist, and (re)using the reduced model in simulation becomes straightforward. The main drawback is that, when the reduced system matrices are dense and the number of terminals is large $[O(10^3)]$, the netlist obtained from RLCSYN will be dense. For a k dimensional reduced network with p terminals, the RLCSYN synthesized netlist will generally have $O(p^2k^2)$ circuit elements. The density of the reduced netlist however is not a result of the synthesis procedure, but a consequence of the fact that the reduced system matrices are dense. Developments for sparsity preserving model reduction for multi-terminal circuits can be found in [160], where sparse netlists are obtained by synthesizing sparse reduced models via RLCSYN.

First, we motivate reduction and synthesis in impedance form, and show how it also applies for systems that are originally in admittance form. Then we explain model reduction via SPRIM/IOPOR, followed by RLCSYN synthesis.

4.5.3.1 A Simple Admittance to Impedance Conversion

In [176] it was shown how SPRIM/IOPOR preserves the structure of the input/output connectivity when the original model is in impedance form, allowing for

[25] A *terminal (external node)* is a node that is visible on the outside, i.e., a node in which currents can be injected. The other nodes are called *internal*.

synthesis via RLCSYN without controlled sources. The emerging question is: How to ensure synthesis without controlled sources, if the original model is in admittance form (i.e., it is voltage driven)? We show that reduction and synthesis via the input impedance transfer function is possible by removing any voltage sources from the original circuit a priori and re-inserting them in the reduced netlist if needed.

To this end, consider the modified nodal analysis (MNA) description of an input *admittance*[26] type *RLC* circuit, driven by voltage sources:

$$\underbrace{\begin{pmatrix} \mathscr{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathscr{L} \end{pmatrix}}_{\mathbf{E}_Y} \underbrace{\frac{d}{dt}\begin{pmatrix} \boldsymbol{v}(t) \\ \boldsymbol{i}_S(t) \\ \boldsymbol{i}_L(t) \end{pmatrix}}_{\dot{\mathbf{x}}_Y} + \underbrace{\begin{pmatrix} \mathscr{G} & \mathscr{E}_v & \mathscr{E}_l \\ -\mathscr{E}_v{}^* & \mathbf{0} & \mathbf{0} \\ -\mathscr{E}_l^* & \mathbf{0} & \mathbf{0} \end{pmatrix}}_{-\mathbf{A}_Y} \underbrace{\begin{pmatrix} \boldsymbol{v}(t) \\ \boldsymbol{i}_S(t) \\ \boldsymbol{i}_L(t) \end{pmatrix}}_{\mathbf{x}_Y} = \underbrace{\begin{pmatrix} \mathbf{0} \\ \mathscr{B} \\ \mathbf{0} \end{pmatrix}}_{\mathbf{B}_Y} \mathbf{u}(t), \quad (4.64)$$

where $\mathbf{u}(t) \in \mathbb{R}^{n_1}$ are input voltages and $\mathbf{y}(t) = \mathbf{C}_Y\mathbf{x}(t) \in \mathbb{R}^{n_1}$ are output currents, $\mathbf{C}_Y = \mathbf{B}_Y^*$. The states are $\mathbf{x}_Y(t) = [\mathbf{v}(t),\ \mathbf{i}_S(t),\ \mathbf{i}_L(t)]^T$, with $\mathbf{v}(t) \in \mathbb{R}^{n_v}$ the node voltages, $\mathbf{i}_S(t) \in \mathbb{R}^{n_1}$ the currents through the voltage sources, and $\mathbf{i}_L(t) \in \mathbb{R}^{n_l}$ the currents through the inductors, $n_v + n_1 + n_l = n$. The $n_v = n_1 + n_2$ node voltages correspond to the n_1 external nodes (i.e., the number of inputs/terminals) and the n_2 internal nodes.[27] Assuming without loss of generality that (4.64) is permuted such that the *first* n_1 nodes are the external nodes, we have: $\mathbf{v}_{1:n_1}(t) = \mathbf{u}(t)$. The dimensions of the underlying matrices are: $\mathscr{C} \in \mathbb{C}^{n_v \times n_v}$, $\mathscr{G} \in \mathbb{C}^{n_v \times n_v}$, $\mathscr{E}_v \in \mathbb{C}^{n_v \times n_1}$, $\mathscr{L} \in \mathbb{C}^{n_l \times n_l}$, $\mathscr{E}_l \in \mathbb{C}^{n_v \times n_l}$, $\mathscr{B} \in \mathbb{C}^{n_1 \times n_1}$. Recalling that $\mathbf{v}_{1:n_1}(t) = \mathbf{u}(t)$, the following holds:

$$\mathscr{E}_v = \begin{pmatrix} \mathscr{B}_v \\ \mathbf{0}_{n_2 \times n_1} \end{pmatrix} \in \mathbb{C}^{n_v \times n_1}, \quad \mathscr{B}_v \in \mathbb{C}^{n_1 \times n_1}, \quad \mathscr{B} = -\mathscr{B}_v. \qquad (4.65)$$

We derive the first order impedance-type system associated with (4.64). Note that by definition, $\mathbf{i}_S(t)$ flows **out of** the circuit terminals into the voltage source (i.e., from the + to the − terminal of the voltage source, see also [166, Figure 3] [158]). We can define new input currents as the currents flowing **into** the circuit terminals: $\mathbf{i}_{in}(t) = -\mathbf{i}_S(t)$. Since $\mathbf{u}(t) = \mathbf{v}_{1:n_1}(t)$ are the terminal voltages, they

[26] The subscript Y refers to quantities associated with a system in admittance form.

[27] For the pencil $(\mathbf{A}_Y, \mathbf{E}_Y)$ to be regular, in (4.64) one node must be chosen as a ground (reference) node; this is however only a numerical requirement.

describe the new output equations, and it is straightforward to rewrite (4.64) in the impedance form:

$$\begin{cases}\underbrace{\begin{pmatrix}\mathscr{C} & \mathbf{0}\\ \mathbf{0} & \mathscr{L}\end{pmatrix}}_{\mathbf{E}}\underbrace{\frac{d}{dt}\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\dot{\mathbf{x}}}+\underbrace{\begin{pmatrix}\mathscr{G} & \mathscr{E}_l\\ -\mathscr{E}_l{}^* & \mathbf{0}\end{pmatrix}}_{-\mathbf{A}}\underbrace{\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix}\mathscr{E}_v\\ \mathbf{0}\end{pmatrix}}_{\mathbf{B}}\mathbf{i}_{in}(t)\\ \underbrace{\begin{pmatrix}\mathscr{E}_v^* & \mathbf{0}\end{pmatrix}}_{\mathbf{C}}\underbrace{\begin{pmatrix}\boldsymbol{v}(t)\\ \boldsymbol{i}_L(t)\end{pmatrix}}_{\mathbf{x}} = \mathbf{y}(t) = \mathscr{B}_v\mathbf{v}_{1:n_1}(t), \quad \mathscr{E}_v^* = \begin{pmatrix}\mathscr{B}_v^* & \mathbf{0}_{n_1\times n_2}\end{pmatrix}\end{cases}\tag{4.66}$$

where $\mathbf{B}$ describes the new *input incidence matrix* corresponding the input currents, $\mathbf{i}_{in}$. The new *output incidence matrix* is $\mathbf{C}$, corresponding to the voltages at the circuit terminals. We emphasize that (4.66) has fewer unknowns than (4.64), since the currents $\mathbf{i}_S$ have been eliminated. The transfer function associated to (4.66) is an input impedance: $\mathbf{H}(s) = \frac{\mathbf{y}(s)}{\mathbf{i}_{in}(s)}$. In Sect. 4.5.3.2 we explain how to obtain an impedance type reduced order model in input/output structure preserved form:

$$\begin{cases}\underbrace{\begin{pmatrix}\tilde{\mathscr{C}} & \mathbf{0}\\ \mathbf{0} & \tilde{\mathscr{L}}\end{pmatrix}}_{\tilde{\mathbf{E}}}\underbrace{\frac{d}{dt}\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\dot{\tilde{\mathbf{x}}}}+\underbrace{\begin{pmatrix}\tilde{\mathscr{G}} & \tilde{\mathscr{E}}_l\\ -\tilde{\mathscr{E}}_l^* & \mathbf{0}\end{pmatrix}}_{-\tilde{\mathbf{A}}}\underbrace{\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\tilde{\mathbf{x}}} = \underbrace{\begin{pmatrix}\tilde{\mathscr{E}}_v\\ \mathbf{0}\end{pmatrix}}_{\tilde{\mathbf{B}}}\mathbf{i}_{in}(t)\\ \underbrace{\begin{pmatrix}\tilde{\mathscr{E}}_v^* & \mathbf{0}\end{pmatrix}}_{\tilde{\mathbf{C}}}\underbrace{\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\tilde{\mathbf{x}}} = \mathbf{y}(t) = \mathscr{B}_v\mathbf{v}_{1:n_1}(t), \quad \tilde{\mathscr{E}}_v^* = \begin{pmatrix}\mathscr{B}_v^* & \mathbf{0}_{n_1\times k_2}\end{pmatrix}\end{cases}\tag{4.67}$$

where $\tilde{\mathscr{C}}$, $\tilde{\mathscr{L}}$, $\tilde{\mathscr{G}}$, $\tilde{\mathscr{E}}_v$ are the reduced MNA matrices, and the reduced input impedance transfer function is: $\tilde{\mathbf{H}}(s) = \frac{\tilde{\mathbf{y}}(s)}{\mathbf{i}_{in}(s)}$. Due to the input/output preservation, the circuit terminals are easily preserved in the reduced model (4.67). The simple example in Sect. 4.5.4.1 illustrates the procedure just described.

It turns out that after reduction and synthesis, the reduced model (4.67) can still be used as a voltage driven admittance block in simulation. This is shown next. We can rewrite the second equation in (4.67) as: $\left(-\tilde{\mathscr{E}}_v^* \ \mathbf{0} \ \mathbf{0}\right)\left(\tilde{\mathbf{v}}(t)^T \ \tilde{\mathbf{i}}_S(t)^T \ \tilde{\mathbf{i}}_L(t)^T\right)^T = \mathscr{B}\mathbf{u}(t)$. This result together with $\mathbf{i}_{in}(t) = -\mathbf{i}_S(t)$, reveals that (4.67) can be rewritten as:

$$\underbrace{\begin{pmatrix}\tilde{\mathscr{C}} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \tilde{\mathscr{L}}\end{pmatrix}}_{\tilde{\mathbf{E}}_Y}\underbrace{\frac{d}{dt}\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \boldsymbol{i}_S(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\dot{\tilde{\mathbf{x}}}_Y}+\underbrace{\begin{pmatrix}\tilde{\mathscr{G}} & \tilde{\mathscr{E}}_v & \tilde{\mathscr{E}}_l\\ -\tilde{\mathscr{E}}_v^* & \mathbf{0} & \mathbf{0}\\ -\tilde{\mathscr{E}}_l^* & \mathbf{0} & \mathbf{0}\end{pmatrix}}_{-\tilde{\mathbf{A}}_Y}\underbrace{\begin{pmatrix}\tilde{\boldsymbol{v}}(t)\\ \boldsymbol{i}_S(t)\\ \tilde{\boldsymbol{i}}_L(t)\end{pmatrix}}_{\tilde{\mathbf{x}}_Y} = \underbrace{\begin{pmatrix}\mathbf{0}\\ \mathscr{B}\\ \mathbf{0}\end{pmatrix}}_{\tilde{\mathbf{B}}_Y}\mathbf{u}(t), \tag{4.68}$$

which has the same structure as the original admittance model (4.64). Conceptually one could have reduced system (4.64) directly via the input admittance. In that case, synthesis by unstamping via RLCSYN [176] would have required controlled sources [155] to model the connections at the circuit terminals. As shown above, this is avoided by: applying the simple admittance-to-impedance conversion (4.64) to (4.66), reducing (4.66) to (4.67), and finally reinserting voltage sources after synthesis [if the input-output structure preserved admittance reduced admittance (4.68) is needed]. Being only a pre- and post-processing step, the proposed voltage-source removal and re-insertion can be applied irrespective of the model reduction algorithm used. For ease of understanding we relate it here to model reduction via SPRIM/IOPOR.

4.5.3.2 I/O Structure Preserving Reduction and RLCSYN Synthesis

The reduced input impedance model (4.67) is obtained via the input-output structure preserving SPRIM/IOPOR projection [176] as follows. Let $\mathbf{V} = \left(\mathbf{V}_1^T, \mathbf{V}_2^T, \mathbf{V}_3^T\right)^T \in \mathbb{C}^{((n_1+n_2+n_l)\times k)}$ be the projection matrix obtained with PRIMA [166], where $\mathbf{V}_1 \in \mathbb{C}^{(n_1\times k)}$, $\mathbf{V}_2 \in \mathbb{C}^{(n_2\times k)}$, $\mathbf{V}_3 \in \mathbb{C}^{(n_l\times k)}$, $k \geq n_1$, $i = 1\ldots3$. After appropriate orthonormalization (e.g., via Modified Gram-Schmidt [171, Chapter 1]), we obtain: $\tilde{\mathbf{V}}_i = \mathrm{orth}(\mathbf{V}_i) \in \mathbb{C}^{n_i\times k_i}$,$k_i \leq k$. The SPRIM [151] block structure preserving projection is: $\tilde{\mathbf{V}} = \mathrm{blkdiag}\left(\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_2, \tilde{\mathbf{V}}_3\right) \in \mathbb{C}^{n\times(k_1+k_2+k_3)}$, which does not yet preserve the structure of the input and output matrices. The input-output structure preserving SPRIM/IOPOR [176] projection is $\tilde{\mathbf{W}} = \begin{pmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{V}}_3 \end{pmatrix} \in \mathbb{C}^{n\times(n_1+k_2+k_3)}$ where:

$$\mathbf{W} = \begin{pmatrix} \mathbf{I}_{n_1\times n_1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{V}}_2 \end{pmatrix} \in \mathbb{C}^{(n_1+n_2)\times(n_1+k_2)}. \tag{4.69}$$

Recalling (4.65), we obtain the reduced system matrices in (4.67): $\tilde{\mathscr{C}} = \mathbf{W}^*\mathscr{C}\mathbf{W}$, $\tilde{\mathscr{G}} = \mathbf{W}^*\mathscr{G}\mathbf{W}$, $\tilde{\mathscr{L}} = \tilde{\mathbf{V}}_3^*\mathscr{L}\tilde{\mathbf{V}}_3$, $\tilde{\mathscr{E}}_l = \mathbf{W}^*\mathscr{E}_l\tilde{\mathbf{V}}_3$, $\tilde{\mathscr{E}}_v = \mathbf{W}^*\mathscr{E}_v = \left(\mathscr{B}_v^* \; \mathbf{0}_{n_1\times k_2}\right)^*$, which compared to (4.65) clearly preserve input-output structure. Therefore a netlist representation for the reduced impedance-type model can be obtained, that is driven injected currents just as the original circuit. This is done via the RLCSYN [176] unstamping procedure. To this end, we use the Laplace transform and convert (4.67) to the second order form:

$$\begin{cases} [s\tilde{\mathscr{C}} + \tilde{\mathscr{G}} + \frac{1}{s}\tilde{\Gamma}]\tilde{\mathbf{v}}(s) = \tilde{\mathscr{E}}_v\mathbf{i}_{in}(s) \\ \tilde{\mathbf{y}}(s) = \tilde{\mathscr{E}}_v^*\tilde{\mathbf{v}}(s), \end{cases} \tag{4.70}$$

where $\tilde{\mathbf{i}}_L(s) = \frac{1}{s}\tilde{\mathscr{L}}^{-1}\left(\tilde{\mathscr{E}}_l{}^*\right)\tilde{\mathbf{v}}(s)$ and $\tilde{\Gamma} = \tilde{\mathscr{E}}_l\tilde{\mathscr{L}}^{-1}\tilde{\mathscr{E}}_l^*$.

The presentation of RLCSYN follows [176, Sect. 4], [158] and is only summarized here. In circuit simulation, the process of forming the $\mathscr{C}, \mathscr{G}, \mathscr{L}$ system matrices from the individual branch element values is called "stamping". The reverse operation of "unstamping" involves decomposing entry-wise the values of the reduced system matrices in (4.70) into the corresponding R, L, and C values. When applied on reduced models, the unstamping procedure may produce negative circuit elements because the reduced system matrices are no longer diagonally dominant (while the original matrices were). Obtaining positive circuit elements only is subject to further research. The resulting Rs, Ls and Cs are connected in the reduced netlist according to the MNA topology. The reduced input/output matrices of (4.70) directly reveal the input connections in the reduced model via injected currents, without any controlling elements. The prerequisites for an unstamping realization procedure therefore are:

1. The original system is in MNA impedance form (4.66). If the system is of admittance type (4.64), apply the admittance-to-impedance conversion from Sect. 4.5.3.1.
2. In (4.66), no Ls are directly connected to the input terminals so that, after reduction, diagonalization and regularization preserve the input/output structure.
3. System (4.66) is reduced with SPRIM/IOPOR [176] to (4.67) and converted to second order form (4.70). The alternative is to obtain the second order form of the original system first, and reduce it directly with SAPOR/IOPOR [143, 176].
4. The reduced system (4.70) must be diagonalized and regularized according to [176]. Diagonalization ensures that all inductors in the synthesized model are connected to ground (i.e., there are no inductor loops). Regularization eliminates spurious over-large inductors. These steps however are not needed for purely RC circuits.

4.5.4 Numerical Examples

We apply the proposed reduction and synthesis framework on several test cases. The first is a simple circuit which illustrates the complete admittance-to-impedance formulation and the RLCSYN unstampting procedure, as described in Sect. 4.5.3. The second example is a SISO transmission line model, while the third is a MIMO model of a spiral inductor.

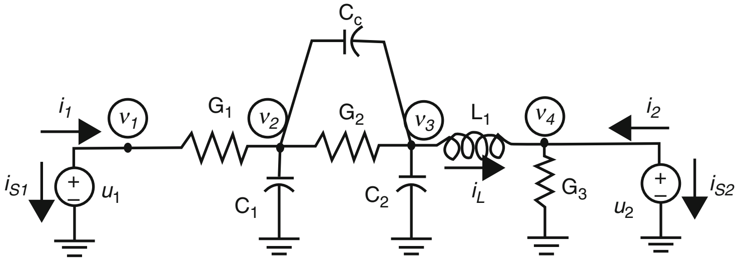


Fig. 4.25 Admittance-type circuit driven by input voltages [166]. $G_{1,2,3} = 0.1S$, $L_1 = 10^{-3}H$, $C_{1,2} = 10^{-6}$, $C_c = 10^{-4}$, $\|u_{1,2}\| = 1$

4.5.4.1 Illustrative Example

The circuit in Fig. 4.25 is voltage driven, and the MNA admittance form (4.64) is:

$$\left(\begin{array}{cccc|cc|c} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_1+C_c & -C_c & 0 & 0 & 0 \\ 0 & 0 & -C_c & C_2+C_c & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & L \end{array}\right) \overset{\cdot}{\left(\begin{array}{c} v_1 \\ v_4 \\ v_2 \\ v_3 \\ \hline i_{S_1} \\ i_{S_2} \\ \hline i_L \end{array}\right)} + \left(\begin{array}{cccc|cc|c} G_1 & 0 & -G_1 & 0 & 1 & 0 & 0 \\ 0 & G_3 & 0 & 0 & 0 & 1 & 1 \\ -G_1 & 0 & G_1+G_2 & -G_2 & 0 & 0 & 0 \\ 0 & 0 & -G_2 & G_2 & 0 & 0 & 1 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & -1 & 0 & 0 & 0 \end{array}\right) \left(\begin{array}{c} v_1 \\ v_4 \\ v_2 \\ v_3 \\ \hline i_{S_1} \\ i_{S_2} \\ \hline i_L \end{array}\right) = \left(\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \hline -1 & 0 \\ 0 & -1 \\ \hline 0 & 0 \end{array}\right) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{4.71}$$

Notice that

$$\mathbf{i}_{in} = \begin{pmatrix} i_1 \\ i_2 \end{pmatrix} = - \begin{pmatrix} i_{S_1} \\ i_{S_2} \end{pmatrix} \tag{4.72}$$

$$\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_4 \end{pmatrix}, \tag{4.73}$$

thus the external nodes (input nodes/terminals) are v_1 and v_4, and the internal nodes are v_2 and v_3. As described in Sect. 4.5.3.1, (4.71) has an equivalent impedance formulation (4.66), with:

$$\mathscr{C} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C_1+C_c & -C_c \\ 0 & 0 & -C_c & C_2+C_c \end{pmatrix}, \ \mathscr{L} = \begin{pmatrix} L \end{pmatrix}, \ \mathscr{G} = \begin{pmatrix} G_1 & 0 & -G_1 & 0 \\ 0 & G_3 & 0 & 0 \\ -G_1 & 0 & G_1+G_2 & -G_2 \\ 0 & 0 & -G_2 & G_2 \end{pmatrix}, \ \mathscr{E}_l = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \tag{4.74}$$

$$\mathscr{E}_v = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \ \mathscr{B} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}, \ \mathscr{B}_v = -\mathscr{B} \tag{4.75}$$

Matrices (4.74) and (4.75) are reduced either in first order form using SPRIM/IOPOR according to Sect. 4.5.3.2.

Here we reduce the circuit with SPRIM/IOPOR and synthesize it by unstamping via RLCSYN. Note that there is an L directly connected to the second input node v_4, thus assumption 2. from RLCSYN is not satisfied. We thus reduce and synthesize the single-input-single-output version of (4.71) only, where the second input i_2 is removed. Therefore the new incidence matrices are:

$$\mathscr{E}_{v_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \mathscr{B}_1 = \begin{pmatrix} -1 \end{pmatrix}, \ \mathscr{B}_{v_1} = -\mathscr{B}_1. \tag{4.76}$$

We choose an underlying PRIMA projection matrix $\mathbf{V} \in \mathbb{C}^{n \times k}$ spanning a $k = 2$-dimensional Krylov subspace (with expansion point $s_0 = 0$). According to Sect. 4.5.3.2, after splitting $\mathbf{V}$ and appropriate re-orthonormalization, the dimensions of the input-output structure preserving partitioning are:

$$n_1 = 1, \ n_2 = 3, \ n_l = 1, \ k_2 = 2, \ k_3 = 1, \tag{4.77}$$

and the SPRIM/IOPOR projection is:

$$\tilde{\mathbf{W}} = \left(\begin{array}{lrr|r} 1 & 0 & 0 & 0 \\ 0 & 4.082 \cdot 10^{-1} & -4.861 \cdot 10^{-1} & 0 \\ 0 & 8.164 \cdot 10^{-1} & 5.729 \cdot 10^{-1} & 0 \\ 0 & 4.082 \cdot 10^{-1} & -6.597 \cdot 10^{-1} & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \in \mathbb{C}^{5 \times 4}, \text{ with } \mathbf{W} \in \mathbb{C}^{4 \times 3}. \tag{4.78}$$

After diagonalization and regularization, the SPRIM/IOPOR reduced system matrices in (4.70) are:

$$\begin{gathered} \tilde{\mathscr{C}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1.749 \cdot 10^{-5} & -5.052 \cdot 10^{-5} \\ 0 & -5.052 \cdot 10^{-5} & 1.527 \cdot 10^{-4} \end{pmatrix}, \ \tilde{\mathscr{G}} = \begin{pmatrix} 1 & 8.165 \cdot 10^{-2} & -5.729 \cdot 10^{-2} \\ 8.165 \cdot 10^{-2} & 9.999 \cdot 10^{-2} & -7.726 \cdot 10^{-2} \\ -5.7295 \cdot 10^{-2} & -7.7265 \cdot 10^{-2} & 2.084 \cdot 10^{-1} \end{pmatrix} \\ \tilde{\Gamma} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 30.14 \end{pmatrix}, \ \tilde{\mathscr{E}}_{v_1} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \end{gathered} \tag{4.79}$$

Reduced matrices (4.79) are now unstamped individually using RLCSYN. The reduced system dimension in second order form is thus $N = 3$, indicating that the reduced netlist will have 3 nodes and an additional ground node. In the following, we denote by $M_{i,j}$ $i = 1 \ldots N$, $j = 0 \ldots N - 1$ a circuit element connected between nodes (i, j) in the resulting netlist. M represents a circuit element of the type: R,L,C or current source J.

By unstamping $\tilde{\mathscr{G}}$, we obtain the following R values (for simplicity only 4 figures behind the period are shown here, nevertheless in implementation they are computed with machine precision $\epsilon = 10^{-16}$):

$$R_{1,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{G}}_{(1,k)}\right]^{-1} = 8.0417\ \Omega,\ \ R_{1,2} = -\left[\tilde{\mathscr{G}}_{(1,2)}\right]^{-1} = -12.247\ \Omega,\ \ R_{1,3} = -\left[\tilde{\mathscr{G}}_{(1,3)}\right]^{-1} = 17.452\ \Omega,$$

$$R_{2,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{G}}_{(2,k)}\right]^{-1} = 9.5798\ \Omega,\ \ R_{2,3} = -\left[\tilde{\mathscr{G}}_{(2,3)}\right]^{-1} = 12.942\ \Omega,\ \ R_{3,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{G}}_{(3,k)}\right]^{-1} = 13.535\ \Omega.$$

By unstamping $\tilde{\mathscr{C}}$, we obtain the following C values:

$$C_{2,0} = \sum_{k=1}^{3} \tilde{\mathscr{C}}_{(2,k)} = –3.3026 \cdot 10^{-5}\ F,\ \ C_{2,3} = -\tilde{\mathscr{C}}_{(2,3)} = 5.0526 \cdot 10^{-5}\ F,$$

$$C_{3,0} = \left[\sum_{k=1}^{3} \tilde{\mathscr{C}}_{(3,k)}\right]^{-1} = 1.0221 \cdot 10^{-4}\ F.$$

By unstamping $\tilde{\Gamma}$, we obtain the following L values:

$$L_{3,0} = \left[\sum_{k=1}^{3} \tilde{\Gamma}_{(3,k)}\right]^{-1} = 3.317 \cdot 10^{-2}\ H.$$

By unstamping $\tilde{\mathscr{E}}_{v_1}$, we obtain the current source $J_{1,0}$ of amplitude 1 A.

The Pstar [165] equivalent netlist is shown below:.

```
circuit;
     r r_1_0 (1, 0) 8.0417250765565598e+000;
     r r_1_2 (1, 2) -1.2247448713915894e+001;
     r r_1_3 (1, 3) 1.7452546181796258e+001;
     r r_2_0 (2, 0) 9.5798755840972589e+000;
     r r_2_3 (2, 3) 1.2942609947762115e+001;
     r r_3_0 (3, 0) 1.3535652691596653e+001;
     l l_3_0 (3, 0) 3.3170000000000033e-002;
     c c_2_0 (2, 0) -3.3026513336014821e-005;
     c c_2_3 (2, 3) 5.0526513336014765e-005;
     c c_3_0 (3, 0) 1.0221180442099465e-004;
     j j_1 (1, 0) sw(1, 0);
     c: Set node 1 as output: vn(1);
     c: Resistors 6;
     c: Capacitors 3;
     c: Inductors 1;
end;
```

Table 4.5 summarizes the reduction and synthesis results. Even though the number of internal variables (states) generated by the simulator is smaller for the SPRIM/IOPOR model than for the original, the number of circuit elements generated by RLCSYN is larger in the reduced model than in the original. Figure 4.26 shows that approximation with SPRIM/IOPOR is more accurate than with PRIMA. The Pstar simulation of the RLCSYN synthesized model also matches the MATLAB simulation of the reduced transfer function.

Table 4.5 Input impedance reduction (SPRIM/IOPOR) and synthesis (RLCSYN)

System	Dimension	*R*	*C*	*L*	States	Inputs/Outputs
Original	5	3	3	1	5	1
SPRIM/IOPOR	4	6	3	1	4	1

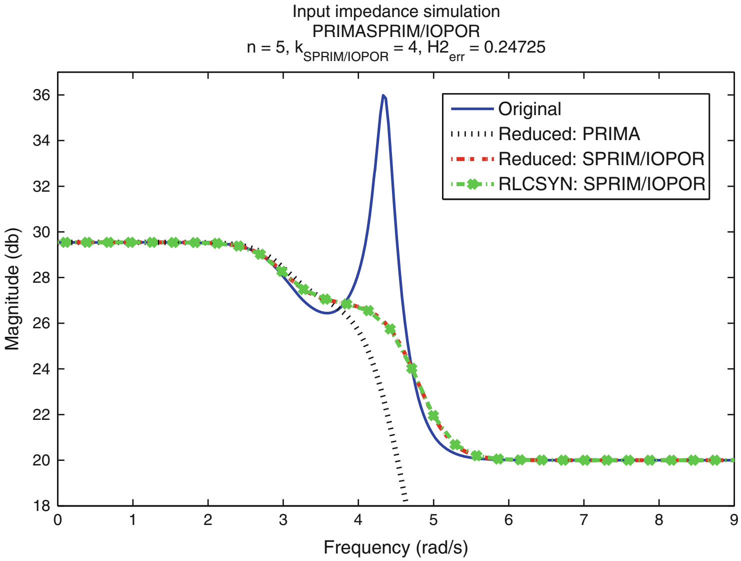


Fig. 4.26 Original, reduced and synthesized systems: PRIMA, SPRIM/IOPOR. The reduced and synthesized systems match but miss the peak around 4.5 rad/s

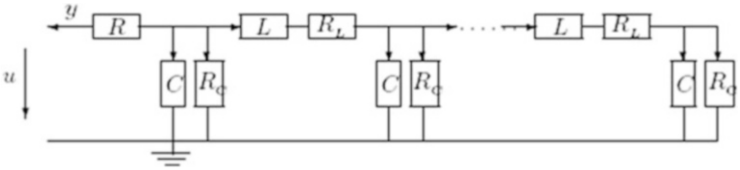


Fig. 4.27 Transmission line from Sect. 4.5.4.2

4.5.4.2 SISO *RLC* Network

We reduce the SISO *RLC* transmission line in Fig. 4.27. Note that the circuit is driven by the voltage **u**, thus it is of admittance type (4.64). The admittance simulation of the model reduced with the *dominant spectral zero method (Dominant SZM)* [157, 161], synthesized with the Foster approach, is shown in Fig. 4.28. The behavior of the original model is well approximated for the entire frequency range, and can also reproduce oscillations at dominant frequency points.

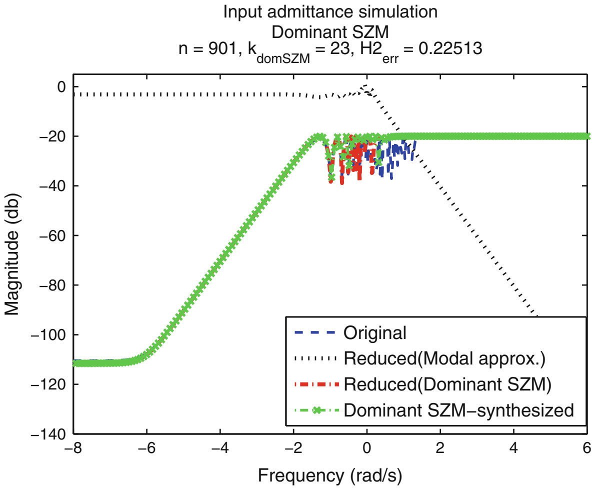


Fig. 4.28 Input admittance transfer function: original, reduced with Dominant SZM in admittance form and synthesized with Foster admittance

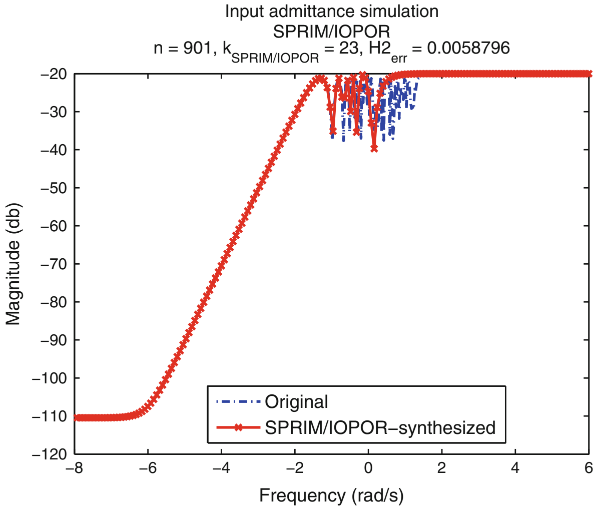


Fig. 4.29 Input admittance transfer function: original and synthesized SPRIM/IOPOR model (via impedance), after reconnecting the voltage source at the input terminal

In Fig. 4.29 the benefit of the admittance-to-impedance transformation, described in Sect. 4.5.3.1, is seen. By reducing the system in impedance form with SPRIM/-IOPOR and synthesizing (4.67) [using the second order form (4.70)] with RLCSYN [176], we are able to recover the reduced admittance (4.68) as well. The approximation is good for the entire frequency range.

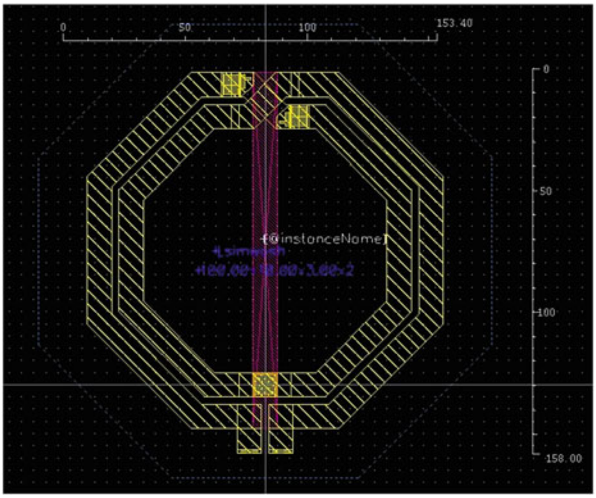


Fig. 4.30 Coil structure from Sect. 4.5.4.3

4.5.4.3 MIMO *RLC* Network

We reduce the MIMO *RLC* netlist resulting from the parasitic extraction [153] of the coil structure in Fig. 4.30. The model has 4 pins (external nodes). Pin 4 is connected to other circuit nodes only via C's, which causes the original model (4.66) to have a pole at 0. The example shows that the SPRIM/IOPOR model preserves the terminals and is synthesizable with RLCSYN without controlled sources.

Figure 4.31, shows the simulation of the transfer function from input 4 to output 4. SPRIM/IOPOR is more accurate around DC than PRIMA. Another alternative is to ground pin 4 prior to reduction. As seen from Fig. 4.32, SPRIM/IOPOR applied on the remaining 3-terminal system gives better approximation than PRIMA for the entire frequency range. With pin 4 grounded however, we loose the ability to (re)connect the synthesized model in simulation via all the terminals.

4.5.5 Conclusions and Outlook

A framework for realizing reduced mathematical models into *RLC* netlists was developed. Model reduction by projection for *RLC* circuits was described and associated with two synthesis approaches: Foster realization (for SISO transfer functions) and RLCSYN [176] synthesis by unstamping (for MIMO systems). An admittance-to-impedance conversion was prosed as a pre-model reduction step and shown to enable synthesis without controlled sources. The approaches were

Fig. 4.31 Input impedance transfer function with "v_4" kept: $\mathbf{H}_{44}$ for PRIMA, SPRIM/IOPOR and RLCSYN realization

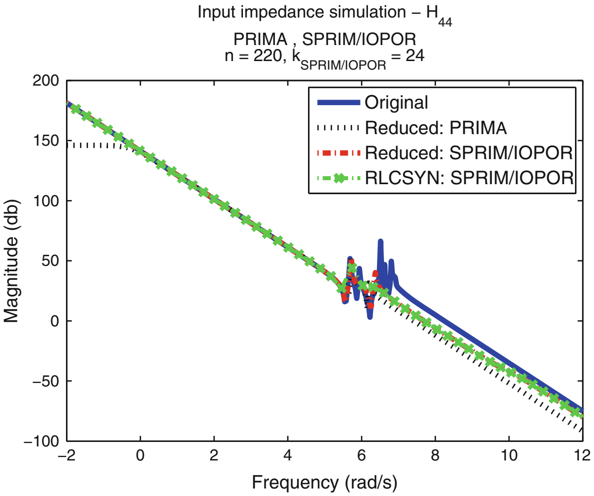
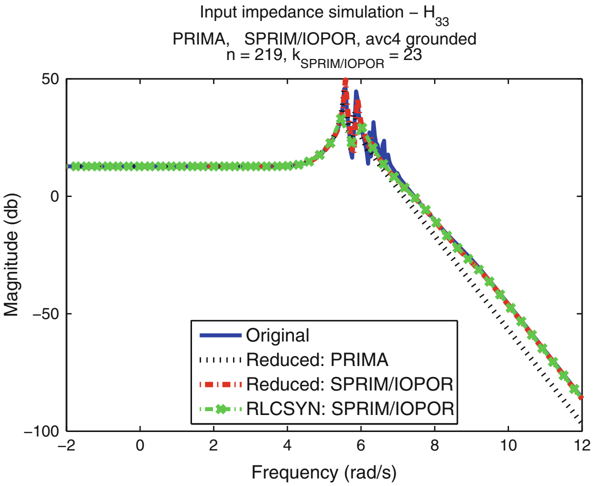


Fig. 4.32 Input impedance transfer function with "v_4" grounded: $\mathbf{H}_{33}$ for PRIMA, SPRIM/IOPOR and RLCSYN realization



tested on several examples. Future research will investigate reduction and synthesis methods for *RCLK* circuits with many terminals, while developments on sparsity-preserving model reduction for multi-terminal *RC* circuits can be found in [160].

References

References for Section 4.1

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
2. Bai, Z.: Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. Appl. Numer. Math. **43**, 9–44 (2002). doi:10.1016/S0168-9274(02)00116-2. http://dl.acm.org/citation.cfm?id=765737.765740
3. Bechtold, T., Verhoeven, A., ter Maten, E.J.W., Voss, T.: Model order reduction: an advanced, efficient and automated computational tool for microsystems. In: Cutello, V., Fotia, G., Puccio, L. (eds.) Applied and Industrial Mathematics in Italy II: Selected Contributions from the 8th SIMAI Conference, Baia Samuele. Series on Advances in Mathematics for Applied Sciences, vol. 75, pp. 113–124. World Scientific, Singapore (2007)
4. Benner, P., Quintana-Ortí, E.: Solving stable generalized Lyapunov equations with the matrix sign function. Numer. Algorithms **20**, 75–100 (1999). http://dx.doi.org/10.1023/A:1019191431273. 10.1023/A:1019191431273
5. Benner, P., Sorensen, D.C., Mehrmann, V. (eds.): Dimension Reduction of Large-Scale Systems: Proceedings Workshop Oberwolfach, Germany, 2003. Lecture Notes in Computational Science and Engineering, vol. 45, pp. 19–25. Springer, Berlin/Heidelberg (2005)
6. Benner, P., Schneider, A.: On stability, passivity and reciprocity preservation of ESVDMOR. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 277–287. Springer, Dordrecht (2011)
7. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
8. Deschrijver, D.: Broadband macromodeling of linear systems by vector fitting. Ph.D. thesis, Universiteit Antwerpen (2007)
9. Feldmann, P., Freund, R.W.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. In: Proceedings of the Conference on European Design Automation (EURO-DAC'94), Grenoble, pp. 170–175. IEEE Computer Society Press, Los Alamitos (1994). http://dl.acm.org/citation.cfm?id=198174.198236
10. Ferranti, F.: Parameterized macromodeling and model order reduction of high-speed interconnects. Ph.D. thesis Ghent University (2011)
11. Ferranti, F., Deschrijver, D., Knockaert, L., Dhaene, T.: Data-driven parameterized model order reduction using z-domain multivariate orthonormal vector fitting technique. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 141–148. Springer, Dordrecht (2011)
12. Freund, R.W.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**(1–2), 395–421 (2000). http://linkinghub.elsevier.com/retrieve/pii/S0377042700003964
13. Freund, R.W.: Model reduction methods based on Krylov subspaces. Acta Numer. **12**, 267–319 (2003)
14. Freund, R.W.: SPRIM: structure-preserving reduced-order interconnect macromodeling. In: Proceedings of the 2004 International Conference on Computer-Aided Design (ICCAD'04), San Jose, 7–11 Nov 2004, pp. 80–87. IEEE Computer Society/ACM (2004). doi:http://doi.acm.org/10.1145/1112239.1112258
15. Freund, R.W.: Structure-preserving moder order reduction of RLC circuit equations. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 49–74. Springer, Berlin (2008)
16. Freund, R.W.: The SPRIM algorithm for structure-preservation order reduction of general RCL circuits. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit

Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 25–52. Springer, Dordrecht (2011)

17. Freund, R.W.: Recent advances in structure-preserving model order reduction. In: Li, P., Silveira, L.M., Feldmann, P. (eds.) Simulation and Verification of Electronic and Biological Systems, pp. 43–70. Springer, Dordrecht/ Heidelberg/London/New York (2011)
18. Freund, R.W., Feldmann, P.: Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm. In: Proceedings of the 1996 IEEE/ACM International Conference on Computer-Aided Design (ICCAD'96), San Jose, pp. 280–287. IEEE Computer Society, Washington, DC (1996). http://dl.acm.org/citation.cfm?id=244522.244571
19. Gallivan, K., Grimme, E., Van Dooren, P.: Asymptotic waveform evaluation via a Lanczos method. Appl. Math. Lett. **7**(5), 75–80 (1994). doi:10.1016/0893-9659(94)90077-9. http://www.sciencedirect.com/science/article/pii/0893965994900779
20. Green, M., Limebeer, D.: Linear Robust Control. Prentice-Hall Information and System Sciences Series. Prentice Hall, Englewood Cliffs (1995). http://books.google.de/books?id=8NdSAAAAMAAJ
21. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois at Urbana-Champaign (1997)
22. Günther, M., Feldmann, U., ter Maten, E.J.W.: Modelling and discretization of circuit problems. In: W.H.A. Schilders, E.J.W. ter Maten (eds.) Handbook of Numerical Analysis. Special Volume Numerical Analysis of Electromagnetism, vol. XIII, pp. 523–659. Elsevier/North Holland, Amsterdam (2005). doi:10.1016/s1570-8659(04)13006-8
23. Gugercin, S., Antoulas, A.C., Beattie, C.: H2 model reduction for large-scale linear dynamical systems. SIAM J. Matrix Anal. Appl. **30**(2), 609–638 (2008)
24. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II – Stiff and Differential-Algebraic Problems, 2nd rev. edn. Springer, Berlin (1996)
25. Honkala, M.: Building blocks for fast circuit simulation. Ph.D. thesis, Aalto University (2013)
26. Ionescu, T.C., Astolfi, A.: Families of moment matching based, structure preserving approximations for linear port Hamiltonian systems. Automatica **49**(8), 2424–2434 (2013)
27. Ionutiu, R., Rommes, J.: "A framework for synthesis of reduced order models". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 3, Section 4.5, this volume (2015)
28. Kailath, T.: Linear Systems. Prentice-Hall Information and System Sciences Series. Prentice Hall International, Singapore (1998). http://books.google.de/books?id=MAfiOgAACAAJ
29. Kunkel, P., Mehrmann, V.: Differential-Algebraic Equations: Analysis and Numerical Solution. EMS Textbooks in Mathematics. European Mathematical Society, Zürich (2006)
30. Li, J., White, J.: Low rank solution of Lyapunov equations. SIAM J. Matrix Anal. Appl. **24**(1), 260–280 (2002). doi:10.1137/S0895479801384937. http://epubs.siam.org/doi/abs/10.1137/S0895479801384937
31. Martins, N., Lima, L., Pinto, H.: Computing dominant poles of power system transfer functions. IEEE Trans. Power Syst. **11**(1), 162–170 (1996)
32. Miettinen, P., Honkala, M., Roos, J.: PARTMOR: partitioning-based RL-in-RL-out MOR method. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 547–594. Springer (2010)
33. Mohaghegh, K.: Linear and nonlinear model order reduction for numerical simulation of electric circuits. Ph.D. thesis, Bergische Universität Wuppertal. Logos Verlag, Berlin (2010)
34. Mohaghegh, K., Striebel, M., ter Maten, J., Pulch, R.: Nonlinear model order reduction based on trajectory piecewise linear approach: comparing different linear cores. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 563–570. Springer (2010)
35. Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **26**(1), 17–32 (1981)
36. Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. In: Proceedings of the 1997 IEEE/ACM international conference

on Computer-Aided Design (ICCAD'97), San Jose, pp. 58–65. IEEE Computer Society, Washington, DC (1997). http://dl.acm.org/citation.cfm?id=266388.266423

37. Phillips, J.R., Silveira, L.M.: Poor Man's TBR: a simple model reduction scheme. IEEE Trans. Comput.-Aided Des. Circuits Syst. **24**(1), 43–55 (2005)
38. Pillage, L., Rohrer, R.: Asymptotic waveform evaluation for timing analysis. IEEE Trans. Comput.-Aided Des. Circuits Syst. **9**(4), 352–366 (1990). doi:10.1109/43.45867
39. Pinnau, R.: Model reduction via proper orthogonal decomposition. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 95–109. Springer, Berlin (2008)
40. Polyuga, R.V.: Model reduction of port-Hamiltonian systems. Ph.D. thesis, Rijksuniversiteit Groningen (2010). http://dissertations.ub.rug.nl/faculties/science/2010/r.v.polyuga/
41. Polyuga, R.V., van der Schaft, A.J.: Structure preserving port-Hamiltonian model reduction of electrical circuits. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 241–260. Springer, Dordrecht (2011)
42. Pulch, R., ter Maten, E.J.W., Augustin, F.: Sensitivity analysis and model order reduction for random linear dynamical systems. CASA-Report 2013–15, TU Eindhoven, http://www.win.tue.nl/analysis/reports/rana13-15.pdf. IMACM-Report 2013–7, Bergische Universität Wuppertal (2013)
43. Rathinam, M., Petzold, L.R.: A new look at proper orthogonal decomposition. SIAM J. Numer. Anal. **41**, 1893–1925 (2003)
44. Reis, T.: Circuit synthesis of passive descriptor systems—a modified nodal approach. Int. J. Circuit Theory Appl. **38**(1), 44–68 (2010). doi:10.1002/cta.532. http://dx.doi.org/10.1002/cta.532
45. Rommes, J.: Modal approximation and computation of dominant poles. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 177–193. Springer, Berlin (2008)
46. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
47. Roos, J., Honkala, M., Miettinen, P.: GABOR: Global-approximation-based order reduction. In: In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 507–514. Springer (2010)
48. Saadvandi, M.: Passivity preserving model reduction and selection of spectral zeros. MSc-thesis, Royal Institute of Technology KTH, Stockholm. Also published as Technical Note NXP-TN-2008/00276, Unclassified Report, NXP Semiconductors, Eindhoven (2008)
49. Saak, J., Benner, P.: Efficient solution of large scale Lyapunov and Riccati equations arising in model order reduction problems. Proc. Appl. Math. Mech. (PAMM) **8**(1), 10085–10088 (2008). doi:10.1002/pamm.200810085. http://dx.doi.org/10.1002/pamm.200810085
50. Schilders, W.H.A.: Introduction to Model Order Reduction. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 3–32. Springer, Berlin (2008)
51. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
52. Schilders, W.H.A.: The need for novel model order reduction techniques in the electronics industry. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 3–24. Springer, Dordrecht (2011)
53. Sorensen, D.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**(4), 347–360 (2005)
54. Stykel, T.: Gramian-based model reduction for descriptor systems. Math. Control Signals Syst. (MCSS) **16**(4), 297–319 (2004)
55. Vollebregt, A.J., Bechtold, T., Verhoeven, A., ter Maten, E.J.W.: Model order reduction of large ODE systems: MOR for Ansys versus ROM Workbench. In: Ciuprina, G., Ioan, D. (eds.)

Scientific Computing in Electrical Engineering. Mathematics in Industry, vol. 11, pp. 175–182. Springer, Berlin/New York (2007)

56. Willcox, K., Peraire, J., White, J.: An Arnoldi approach for generation of reduced-order models for turbomachinery. Comput. Fluids **31**(3), 369–389 (2002)

References for Section 4.2

57. Aguirre, L.A.: Quantitative measure of modal dominance for continuous systems. In: Proceedings of the 32nd Conference on Decision and Control, San Antonio, pp. 2405–2410 (1993)
58. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
59. Bai, Z., Su, Y.: SOAR: a second-order Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal. Appl. **26**(3), 640–659 (2005)
60. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin/New York (2005)
61. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
62. Fokkema, D.R., Sleijpen, G.L.G., van der Vorst, H.A.: Jacobi-Davidson style QR and QZ algorithms for the reduction of matrix pencils. SIAM J. Sci. Comput. **20**(1), 94–125 (1998)
63. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. John Hopkins University Press, Baltimore (1996)
64. Green, M., Limebeer, D.J.N.: Linear Robust Control. Prentice-Hall, Englewood Cliffs (1995)
65. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois (1997)
66. Haley, S.B.: The generalized eigenproblem: pole-zero computation. Proc. IEEE **76**(2), 103–120 (1988)
67. Hochstenbach, M.E., Sleijpen, G.L.G.: Two-sided and alternating Jacobi-Davidson. Linear Algebra Appl. **358**(1–3), 145–172 (2003)
68. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
69. Knyazev, A.V.: Preconditioned eigensolvers. In: Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., van der Vorst, H. (eds.) Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, pp. 337–368. SIAM, Philadelphia (2000)
70. Knyazev, A.V.: Hard and soft locking in iterative methods for symmetric eigenvalue problems. In: Copper Mountain Conference, Copper Mountain (2004). http://math.cudenver.edu/~aknyazev/research/conf/cm04.htm
71. Lampe, J., Voss, H.: Second order Arnoldi reduction: application to some engineering problems. Report 93, Hamburg University of Technology (2006)
72. Martins, N., Lima, L.T.G., Pinto, H.J.C.P.: Computing dominant poles of power system transfer functions. IEEE Trans. Power Syst. **11**(1), 162–170 (1996)
73. Martins, N., Pellanda, P.C., Rommes, J.: Computation of transfer function dominant zeros with applications to oscillation damping control of large power systems. IEEE Trans. Power Syst. **22**(4), 1218–1226 (2007)
74. Martins, N., Quintão, P.E.M.: Computing dominant poles of power system multivariable transfer functions. IEEE Trans. Power Syst. **18**(1), 152–159 (2003)
75. Murthy, D.V., Haftka, R.T.: Derivatives of eigenvalues and eigenvectors of a general complex matrix. Int. J. Numer. Methods Eng. **26**, 293–311 (1988)
76. Ostrowski, A.M.: On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors. I. Arch. Ration. Mech. Anal. **1**, 233–241 (1958)
77. Parlett, B.N.: The Rayleigh quotient iteration and some generalizations for nonnormal matrices. Math. Comput. **28**(127), 679–693 (1974)

78. Parlett, B.N.: The Symmetric Eigenvalue Problem. Classics in Applied Mathematics. SIAM, Philadelphia (1998)
79. Peters, G., Wilkinson, J.H.: Inverse iteration, ill-conditioned equations and Newton's method. SIAM Rev. **21**(3), 339–360 (1979)
80. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D. thesis, Utrecht University (2007). http://igitur-archive.library.uu.nl/dissertations/2007-0626-202553/index.htm
81. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
82. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(3), 1218–1226 (2006)
83. Rommes, J., Martins, N.: Computing large-scale system eigenvalues most sensitive to parameter changes, with applications to power system small-signal stability. IEEE Trans. Power Syst. **23**(4), 434–442 (2008)
84. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles of large second-order dynamical systems. SIAM J. Sci. Comput. **30**(4), 2137–2157 (2008)
85. Rommes, J., Sleijpen, G.L.G.: Convergence of the dominant pole algorithm and Rayleigh quotient iteration. SIAM J. Matrix Anal. Appl. **30**(1), 346–363 (2008)
86. Ruhe, A.: Rational Krylov: a practical algorithm for large sparse nonsymmetric matrix pencils. SIAM J. Sci. Comput. **19**(5), 1535–1551 (1998)
87. Saad, Y.: Numerical Methods for Large Eigenvalue Problems: Theory and Algorithms. Manchester University Press, Manchester (1992)
88. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
89. Sleijpen, G.L.G., van der Vorst, H.A.: A Jacobi-Davidson iteration method for linear eigenvalue problems. SIAM J. Matrix Anal. Appl. **17**(2), 401–425 (1996)
90. Smith, J.R., Hauer, J.F., Trudnowski, D.J., Fatehi, F., Woods, C.S.: Transfer function identification in power system application. IEEE Trans. Power Syst. **8**(3), 1282–1290 (1993)
91. Stathopoulos, A.: A case for a biorthogonal Jacobi-Davidson method: restarting and correction equation. SIAM J. Matrix Anal. Appl. **24**(1), 238–259 (2002)
92. Matlab, available from The MathWorks, Inc., Natick, MA, 01760-2098 USA, http://www.mathworks.com.
93. van der Vorst, H.A.: Computational Methods for Large Eigenvalue Problems. In: Ciarlet, P.G., Lions, J.L. (eds.) Handbook of Numerical Analysis, vol. VIII, pp. 3–179. North-Holland (Elsevier), Amsterdam (2001)
94. Varga, A.: Enhanced modal approach for model reduction. Math. Model. Syst. **1**, 91–105 (1995)
95. Wang, L., Howell, F., Kundur, P., Chung, C.Y., Xu, W.: A tool for small-signal security assessment of power systems. In: Proceedings of the IEEE International Conference on Power Industry Computer Applications PICA 2001, Sydney, Australia, pp. 246–252 (2001)

References for Section 4.3

96. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**(4), 361–374 (2005)
97. Bai, Z., Freund, R.: Eigenvalue-based characterization and test for positive realness of scalar transfer functions. IEEE Trans. Autom. Control **AC-45**, 2396–2402 (2000)
98. Bai, Z., Feldmann, P., Freund, R.: Stable and passive reduced-order model based on partial Padè approximation via the Lanczos process. Bell Laboratories, Lucent Technologies Numerical Analysis Manuscript 97/3-10 (1997)

99. Bai, Z., Feldmann, P., Freund, R.: How to make theoretically passive reduced-order models passive in practice. In: Proceeding of the IEEE 1998 Custom Integrated Circuits Conference, Santa Clatra, pp. 207–210 (1998)
100. Feldman, P., Freund, R.W.: Efficient linear circuit analysis by Padè approximation via a Lanczos method. IEEE Trans. Comput.-Aided Des. **14**, 639–649 (1995)
101. Freund, R.: Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms. In: Proceedings of the ACM DAC'99, New Orleans (1999)
102. Gohberg, I., Lancaster, P., Rodman, L.: Invariant subspace of matrices with applications. SIAM, Philadelphia (1986)
103. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois, Urbana-Champaign (1997)
104. Gugercin, S., Antoulas, A.C.: On balancing related model reduction methods and the corresponding error. Int. J. Control **77**(8), 748–766 (2004)
105. Nong, Hung Dinh: Passivity preserving model reduction via interpolation of spectral zeros: selection criteria and implementation. Master thesis, Rice University (2007)
106. Ober, R.: Balanced parametrization of classes of linear systems. SIAM J. Control Optim. **29**, 1251–1287 (1991)
107. Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **17**, 645–654 (1998)
108. Saadvandi, M.: Passivity preserving model reduction and selection of spectral zeros. MSc-thesis, Royal Institute of Technology KTH, Stockholm. Also published as Technical Note NXP-TN-2008/00276, Unclassified Report, NXP Semiconductors, Eindhoven (2008)
109. Saadvandi, M.: Nonlinear and parametric model order reduction for second order dynamical systems by the dominant pole algorithm. Ph.D. thesis, KU Leuven (2013)
110. Sorensen, D.C.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**(4), 361–374 (2005)
111. De Villemagne, C., Skelton, R.: Model reduction using a projection formulation. Int. J. Control **40**, 2142–2169 (1987)
112. Zhou, K., Doyle, J.C., Glover, K.: Robust and Optimal Control. Prentice Hall, Upper Saddle River (1996)

References for Section 4.4

113. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
114. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**, 361–374 (2005)
115. Bai, Z., Li, R., Su, Y.: A unified Krylov projection framework for structure-preserving model reduction. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 75–93. Springer, Berlin (2008)
116. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
117. Freund, R.: SPRIM: Structure-preserving reduced-order interconnect macromodeling. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'04), San Jose, pp. 80–87. Los Alamitos (2004)
118. Freund, R.: Structure preserving model order reduction of RCL circuit equations. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 49–73. Springer, Berlin (2008)
119. Guillemin, E.A.: Synthesis of Passive Networks, 2nd edn. Wiley, New York/London (1959)

120. Ionutiu, R.: Passivity preserving model reduction in the context of spectral zero interpolation. Master's thesis, William Marsh Rice University, Houston (2008)
121. Ionutiu, R.: Model order reduction for multi-terminal systems – with applications to circuit simulation. Ph.D. thesis, TU Eindhoven (2011). http://alexandria.tue.nl/extra2/716352.pdf
122. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. NXP-TN 2008/00316, NXP Semiconductors (2008)
123. Ionutiu, R., Rommes, J.: "A framework for synthesis of reduced order models". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 3, Section 4.5, this volume (2013)
124. Ionutiu, R., Rommes, J.: "Model order reduction for multi-terminal circuits". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 5, Section 6.2, this volume (2013)
125. Ionutiu, R., Rommes, J., Antoulas, A.: Passivity preserving model reduction using dominant spectral zero interpolation. IEEE Trans. Comput.-Aided Des. Circuits Syst. **27**(12), 2250–2263 (2008)
126. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
127. Kressner, D.: Numerical methods for general and structured eigenvalue problems. Lecture Notes in Computational Science and Engineering, vol. 46. Springer, Berlin (2005)
128. Odabasioglu, A., Celik, M., Pillegi, L.: Prima: Passive reduced-order interconnect macromodelling algorithm. IEEE Trans. Comput.-Aided Des. Circuits Syst. **17**, 645–654 (1998)
129. Phillips, J., Daniel, L., Silveira, L.: Guaranteed passive balancing transformations for model order reduction. IEEE Trans. Comput.-Aided Des. Circuits Syst. **22**(8), 1027–1041 (2003)
130. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D. thesis, Utrecht University, Utrecht (2007). http://sites.google.com/site/rommes
131. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(3), 1218–1226 (2006)
132. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
133. Sheldon, X.D.T., He, L.: Advanced Model Order Reduction Techniques in VLSI design. Cambridge University Press, Cambridge (2007)
134. Sorensen, D.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**, 347–360 (2005)
135. Varga, A.: Enhanced modal approach for model reduction. Math. Model. Syst. **1**, 91–105 (1995)
136. Verbeek, M.E.: Partial element equivalent circuit (PEEC) models for on-chip passives and interconnects. Int. J. Numer. Model. Electron. Netw. Devices Fields **17**(1), 61–84 (2004)
137. Watkins, D.S.: The Matrix Eigenvalue Problem: GR and Krylov Subspace Methods. SIAM, Philadelphia (2007)
138. Yang, F., Zeng, X., Su, Y., Zhou, D.: RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI. Commun. Comput. Phys. **3**(2), 376–396 (2008)
139. You, H., He, L.: A sparsified vector potential equivalent circuit model for massively coupled interconnects. In: IEEE International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, vol. 1, pp. 105–108 (2005)
140. Zheng, H., Pileggi, L.: Robust and passive model order reduction for circuits containing susceptance elements. In: Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD'02), San Jose, pp. 761–766 (2002)

References for Section 4.5

141. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
142. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**, 361–374 (2005)
143. Bai, Z., Li, R., Su, Y.: A unified Krylov projection framework for structure-preserving model reduction. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 75–93. Springer, Berlin (2008)
144. Benner, P.: Advances in balancing-related model reduction for circuit simulation. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry, vol. 14, pp. 469–482. Springer (2010)
145. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin (2005)
146. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Berlin (2011)
147. Bott, R., Duffin, R.: Impedance synthesis without the use of transformers. J. Appl. Phys. **20**, 816 (1949)
148. Brune, O.: Synthesis of a finite two-terminal network whose driving point impedance is a prescribed function of frequency. J. Math. Phys. **10**, 191–236 (1931)
149. Jivaro, available at EdXact SA, Voiron, France, http://www.edxact.com
150. The Spice Home Page, http://bwrcs.eecs.berkeley.edu/Classes/IcBook/SPICE/, EECS Dept., Univ. of California at Berkeley, CA 94720-1770, USA.
151. Freund, R.: SPRIM: structure-preserving reduced-order interconnect macromodeling. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD'04), San Jose, pp. 80–87. Los Alamitos (2004)
152. Grimme, E.J.: Krylov projection methods for model reduction. Ph.D. thesis, University of Illinois (1997)
153. Guille, A., Hanssen, M., Niehof, J.: Comparison between RLCk extraction and EM simulation on RF circuits. Tech. rep., NXP Semiconductors (2008)
154. Guillemin, E.A.: Synthesis of Passive Networks, 2nd edn. Wiley, New York/London (1959)
155. Heres, P.J.: Robust and efficient Krylov subspace methods for model order reduction. Ph.D. thesis, Eindhoven University of Technology (2005)
156. Ionutiu, R.: Model order reduction for multi-terminal systems – with applications to circuit simulation. Ph.D. thesis, TU Eindhoven (2011). http://alexandria.tue.nl/extra2/716352.pdf.
157. Ionutiu, R., Rommes, J., Antoulas, A.: Passivity preserving model reduction using dominant spectral zero interpolation. IEEE Trans. Comput.-Aided Des. Circuits Syst. **27**(12), 2250–2263 (2008)
158. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. Technical Note 2008/00316, NXP Semiconductors (2009)
159. Ionutiu, R., Rommes, J.: On synthesis of reduced order models. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 207–224. Springer, Berlin (2011)
160. Ionutiu, R., Rommes, J.: "Model order reduction for multi-terminal circuits". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 5, Section 6.2, this volume (2013)
161. Ionutiu, R., Rommes, J., Antoulas, A.C.: "Passivity Preserving Model Reduction using the Dominant Spectral Zero Method". In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 3, Section 4.4, this volume (2013)
162. Kailath, T.: Linear Systems. Prentice-Hall, Englewood Cliffs (1980)
163. Kerns, K.J., Yang, A.T.: Preservation of passivity during RLC network reduction via split congruence transformations. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **17**(7), 582–591 (1998)

164. Miettinen, P., Honkala, M., Roos, J., Valtonen, M.: PARTMOR: Partitioning-based realizable model-order reduction method for RLC circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **30**(3), 374–387 (2011)
165. Pstar, in-house industrial circuit simulator, NXP Semiconductors, Eindhoven, The Netherlands, http://www.nxp.com
166. Odabasioglu, A., Celik, M., Pileggi, T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **17**(8), 645–654 (1998)
167. Palenius, T., Roos, J.: Comparison of reduced-order interconnect macromodels for time-domain simulation. IEEE Trans. Microw. Theory Tech. **52**(9), 191–236 (2004)
168. Phillips, J.R., Silveira, L.M.: Poor Man's TBR: a simple model reduction scheme. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **24**(1), 283–288 (2005)
169. Phillips, J.R., Daniel, L., Silveira, L.: Guaranteed passive balancing transformations for model order reduction. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **22**(8), 1027–1041 (2003). doi:10.1109/TCAD.2003.814949
170. Reis, T.: Circuit synthesis of passive descriptor systems – a modified nodal approach. Int. J. Circuit Theory Appl. **38**(1), 44–68 (2010)
171. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D. thesis, Utrecht University (2007)
172. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 11. Springer, Berlin (2008)
173. Sorensen, D.: Passivity preserving model reduction via interpolation of spectral zeros. Syst. Control Lett. **54**, 347–360 (2005)
174. Stykel, T., Reis, T.: Passivity-preserving model reduction of differential-algebraic equations in circuit simulation. In: Proceedings in Applied Mathematics and Mechanics (ICIAM'07), Zurich, pp. 1021601–1021602 (2007)
175. Tan, S.X.D., He, L.: Advanced model order reduction techniques in VLSI design. Cambridge University Press, Cambridge (2007)
176. Yang, F., Zeng, X., Su, Y., Zhou, D.: RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI. Commun. Comput. Phys. **3**(2), 376–396 (2008)

Chapter 5
Parameterized Model Order Reduction

Gabriela Ciuprina, Jorge Fernández Villena, Daniel Ioan, Zoran Ilievski, Sebastian Kula, E. Jan W. ter Maten, Kasra Mohaghegh, Roland Pulch, Wil H.A. Schilders, L. Miguel Silveira, Alexandra Ştefănescu, and Michael Striebel

Abstract This Chapter introduces parameterized, or parametric, Model Order Reduction (pMOR). The Sections are offered in a prefered order for reading, but can be read independently. Section 5.1, written by Jorge Fernández Villena, L. Miguel Silveira, Wil H.A. Schilders, Gabriela Ciuprina, Daniel Ioan and

G. Ciuprina (✉) • D. Ioan
Politehnica University of Bucharest, Spl.Independentei 313, 060042 Bucharest, Romania
e-mail: Gabriela@lmn.pub.ro; Daniel@lmn.pub.ro

J. Fernández Villena • L.M. Silveira
INESC ID/IST - TU Lisbon, Rua Alves Redol 9, 1000-029 Lisbon, Portugal
e-mail: Jorge.Fernandez@inesc-id.pt; LMS@inesc-id.pt

Z. Ilievski
European Space & Technology Centre, Keplerlaan 1, P.O. Box 299, 2200 AG Noordwijk, The Netherlands
e-mail: ZoranI@gmail.com

S. Kula
Institute of Mechanics and Applied Computer Science, Kazimierz Wielki University, ul. Kopernika 1, 85-074 Bydgoszcz, Poland
e-mail: SKula@ukw.edu.pl

E. Jan W. ter Maten
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Gaußstraße 20, D-42119 Wuppertal, Germany

Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O. Box 513, 5600 Eindhoven, The Netherlands
e-mail: Jan.ter.Maten@math.uni-wuppertal.de; E.J.W.ter.Maten@tue.nl

K. Mohaghegh
Multiscale in Mechanical and Biological Engineering (M2BE), Aragón Institute of Engineering Research (I3A), University of Zaragoza, María de Luna, 3, E-50018 Zaragoza, Spain
e-mail: Kasra@unizar.es

R. Pulch
Institut für Mathematik und Informatik, Ernst Moritz Arndt Universität Greifswald, Walther-Rathenau-Straße 47, D-17487 Greifswald, Germany
e-mail: PulchR@uni-greifswald.de

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_5

Sebastian Kula, overviews the basic principles for pMOR. Due to higher integration and increasing frequency-based effects, large, full Electromagnetic Models (EM) are needed for accurate prediction of the real behavior of integrated passives and interconnects. Furthermore, these structures are subject to parametric effects due to small variations of the geometric and physical properties of the inherent materials and manufacturing process. Accuracy requirements lead to huge models, which are expensive to simulate and this cost is increased when parameters and their effects are taken into account. This Section introduces the framework of pMOR, which aims at generating reduced models for systems depending on a set of parameters.

Section 5.2, written by Gabriela Ciuprina, Alexandra Ştefănescu, Sebastian Kula and Daniel Ioan, provides robust procedures for pMOR. This Section proposes a robust specialized technique to extract reduced parametric compact models, described as parametric SPICE-like netlists, for long interconnects modeled as transmission lines with several field effects such as skin effect and substrate losses. The technique uses an EM formulation based on partial differential equations (PDE), which is discretized to obtain a finite state space model. Next, a variability analysis of the geometrical data is carried out. Finally, a method to extract an equivalent parametric circuit is proposed.

Section 5.3, written by Michael Striebel, Roland Pulch, E. Jan W. ter Maten, Zoran Ilievski, and Wil H.A. Schilders, covers ways to efficiently determine sensitivity of output with respect to parameters. First direct and adjoint techniques are considered with forward and backward time integration, respectively. Here also the use of MOR via POD (Proper Orthogonal Decomposition) is discussed. Next, techniques in Uncertainty Quantification are described. Here pMOR techniques can be used efficiently.

Section 5.4, written by Kasra Mohaghegh, Roland Pulch and E. Jan W. ter Maten, provides a novel way in extending MOR to Differential-Algebraic Systems. Here new MOR techniques for reducing semi-explicit system of DAEs are introduced. These techniques are extendable to all linear DAEs. Especially pMOR techniques are exploited for singularly perturbed systems.

A. Ştefănescu
Research Centre of Excellence “Micro- and nanosystems for radiofrequency and photonics”, IMT Bucharest – National Institute for Research and Development in Microtechnologies, 126A (32B), Erou Iancu Nicolae str., 077190 Bucharest, Romania
e-mail: Alexandra.Stefanescu@imt.ro

M. Striebel
ZF Lenksysteme GmbH, Richard-Bullinger-Straße 77, D-73527 Schwäbisch Gmünd, Germany
e-mail: Michael.Striebel@zf-lenksysteme.com

W.H.A. Schilders
Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O. Box 513, 5600 Eindhoven, The Netherlands
e-mail: W.H.A.Schilders@tue.nl

5.1 Parametric Model Order Reduction

Model Order Reduction (MOR) techniques are a set of procedures which aim at replacing a large-scale model of a physical system by a lower dimensional model which exhibits similar behavior, typically measured in terms of its input-output response.[1] Reducing the order or dimension of these models, while guaranteeing that the input-output response is accurately captured, is crucial to enable the simulation and verification of large systems [1–3, 33]. Since the first attempts in this area [31], the methods for linear model reduction have greatly evolved and can be broadly characterized into two types: those that are based on subspace generation and projection methods [13, 27], and those based on balancing techniques [26, 30] (sometimes also referred to as Singular Value Decomposition (SVD)-based [2]). Hybrid techniques that try to combine some of the features of each family have also been presented [18, 19, 21, 29].

Although previously ignored when analyzing or simulating systems, parameter variability can no longer be disregarded as it directly impacts system behavior and performance. Accounting for the effects of manufacturing or operating variability, such as geometric parameters, temperature, etc., leads to parametric models whose complexity must be tackled both during the design and verification phases. For this purpose, *Parametric MOR* (pMOR, also known as *Parameterized MOR*) techniques that can handle parameterized descriptions are being considered as essential in the determination of correct system behavior. The systems generated by pMOR procedures must retain the ability to model the effects of both geometric and operating variability, in order to accurately predict behavior and optimize designs.

Several pMOR techniques have been developed for modeling large-scale parameterized systems. Although the first approaches were based on perturbation based techniques, such as [17, 25], the most common and effective ones appear to be extensions of the basic projection-based MOR algorithms [27, 29] to handle parameterized descriptions. An example of these are multiparameter moment-matching pMOR methods [8] which can generate accurate reduced models that capture both frequency and parameter dependence. The idea is to match, via different approaches, generalized moments of the parametric transfer function, and build an overall projector. Sample-based techniques have been proposed in order to contain the large growth in model order for multiparameter, high accuracy systems [28, 37]. They rely on sampling the joint multi-dimensional frequency and parameters space. This approach allows the inclusion of a priori knowledge of the parameter variation, and provides some error estimation. However, the issue of sample selection becomes particularly relevant when done in a potentially high-dimensional space.

[1] Section 5.1 has been writen by: Jorge Fernández Villena, L. Miguel Silveira, Wil H.A. Schilders, Gabriela Ciuprina, Daniel Ioan and Sebastian Kula. For additional topics and applications see also the Ph.D.-Thesis of the last author [20].

5.1.1 Representation of Parametric Systems

In order to include parametric systems inside an efficient simulation flow, the parametric dependence should be explicit. This means that it must be possible to access the parameter values and modify them inside the same representation, while avoiding, if possible, re-computing the parametric systems, i.e. to perform another extraction.

Parameters usually affect the geometrical or electrical properties of the layout, and thus, most of these variations can be represented as modifications of the values of the system matrices inside a state-space descriptor. For this reason, in most cases, the input and output ports are not affected by these variations (this of course depends on how the system is built), and in the case when they are in fact affected, these variations can be shifted to the inner states. The variability leads to a dependence of the extracted circuit elements on several parameters, of electrical or geometrical origin. This dependence results in a parametric state-space system representation, which in descriptor form can be written as

$$\begin{aligned}&C(\lambda_1,\dots,\lambda_P)\,\dot{x}(\lambda_1,\dots,\lambda_P)+G(\lambda_1,\dots,\lambda_P)\,x(\lambda_1,\dots,\lambda_P)=B\,u,\\&y(\lambda_1,\dots,\lambda_P)=L\,x(\lambda_1,\dots,\lambda_P),\end{aligned}\tag{5.1}$$

where $C, G \in \mathbb{R}^{n\times n}$ are again, respectively, the dynamic and static matrices, $B \in \mathbb{R}^{n\times p}$ is the matrix that relates the input vector $u \in \mathbb{R}^p$ to the inner states $x \in \mathbb{R}^n$ and $L \in \mathbb{R}^{q\times n}$ is the matrix that links those inner states to the outputs $y \in \mathbb{R}^q$. The elements of the matrices C and G, as well as the states of the system x, depend on a set of P parameters $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_P]$ which model the effects of the mentioned uncertainty. This time-domain descriptor yields a parametric dependent frequency response modeled via the transfer function

$$H(s,\lambda_1,\dots,\lambda_P)=L\,\left(s\,C(\lambda_1,\dots,\lambda_P)+G(\lambda_1,\dots,\lambda_P)\right)^{-1}\,B\tag{5.2}$$

for which we seek to generate a reduced order approximation, able to accurately capture the input-output behavior of the system for any point in the parameter space

$$\hat{H}(s,\lambda_1,\dots,\lambda_P)=\hat{L}\,\left(s\,\hat{C}(\lambda_1,\dots,\lambda_P)+\hat{G}(\lambda_1,\dots,\lambda_P)\right)^{-1}\,\hat{B}.\tag{5.3}$$

In general, one attempts to generate a reduced order model whose structure is, as much as possible, similar to the original, i.e. exhibiting a similar parametric dependence. The "de facto" standard used in most of the literature for representing a parametric system is based on a Taylor series expansion with respect to the parameters (shown here for first order in the frequency domain):

$$\begin{aligned}&((C_0+C_1\lambda_1+\ldots+C_P\lambda_P)\,s+(G_0+G_1\lambda_1+\ldots+G_P\lambda_P))\;x(s,\lambda)=B\,u(s),\\&y(s,\lambda)=L\,x(s,\lambda),\end{aligned}\tag{5.4}$$

where G_0 and C_0 are the nominal values of the matrices, whereas G_i and C_i are the sensitivities with respect to the parameters. Novel extraction methodologies can efficiently generate such sensitivity information [5, 12].

A nice feature of this representation is that this explicit parameter dependence allows to obtain a reduced, yet similar representation when a projection scheme is applied

$$\begin{aligned}&((\hat{C}_0 + \hat{C}_1\lambda_1 + \ldots + \hat{C}_P\lambda_P)\,s + (\hat{G}_0 + \hat{G}_1\lambda_1 + \ldots + \hat{G}_P\lambda_P))\,x(s,\lambda) = \hat{B}\,u(s),\\ &y(s,\lambda) = \hat{L}\,x(s,\lambda),\end{aligned} \tag{5.5}$$

where $\hat{C}_i = V^T C_i V$, $\hat{G}_i = V^T G_i V$, $\hat{B} = V^T B$ and $\hat{L} = LV$.

Some questions may be raised about the order neccessary for an accurate representation of the parametric model. This depends on the range of variation and the effect of each parameter, and therefore is not trivial to ascertain.

However, some literature presents interesting results in this area [4, 6], with the conclusion that low order (first order in most cases) Taylor series are a good and useful approximation to the real parametric system. As it will be shown later, this statement has important consequences from the point of view of some parametric algorithms, especially those which rely on moment matching techniques.

5.1.2 *Reduction of Parametric Systems*

The most straight-forward approach for the reduction of such a parametric system is to apply nominal techniques. A first possibility is to apply nominal reduction methodologies on the perturbed system. This means that the model in (5.4) is evaluated for a set of parameter values. This model is no longer parametric, and thus standard reduction methodologies can be applied on it. However, once a "perturbed" system is evaluated and reduced, the parameter dependence is lost, and thus the result is a system which is no longer parametric, and therefore only accurate for a set of parameters.

A slightly different approach that overcomes this issue is to apply the projection on the Taylor series approximation. In this case, depending on the framework applied, we can distinguish two cases:

- First, in a projection methodology, the projector is computed from the nominal system, and later applied on the nominal and on the sensitivity matrices, obtaining a model as in (5.5).
- Second, in the case of Balanced Truncation realizations, the computation of the Gramians is done via the nominal system, but the balancing and the truncation is done both on the nominal matrices and on the sensitivities.

These methods, although not oriented to accurately capture the behavior of the system under variation of the parameters, can yield good approximations in cases of small variations or mild effect of the parameters. However, they are not reliable, and their performance heavily depends on the system.

5.1.2.1 Pertubation Based Parametric Reduction

The first attemps to handle and reduce systems under variations were focused on perturbation techniques.

One of the earliest attempts to address this variational issue was to combine perturbation theory with moment matching MOR algorithms [25] into a **Perturbation-based Projector Fitting scheme**. To model the variational effects of the interconnects, an affine model was built for the capacitance and conductance matrices,

$$\begin{aligned} G(\lambda_1, \ldots, \lambda_P) &= G_0 + \lambda_1 G_1 + \ldots + \lambda_P G_P, \\ C(\lambda_1, \ldots, \lambda_P) &= C_0 + \lambda_1 C_1 + \ldots + \lambda_P C_P, \end{aligned} \tag{5.6}$$

where now C_0 and G_0 are the nominal matrix values, i.e., the value of the matrices under no parameter variation, and C_i and G_i, $i = 1, \ldots, P$, are their sensitivities with respect to those parameters. For small parameter variations, the projection matrix obtained via a moment-matching type algorithm such as PRIMA also may show small perturbations. To capture such effect, several samples in the parameter space were drawn $G(\lambda_1, \ldots, \lambda_P)$ and $C(\lambda_1, \ldots, \lambda_P)$, and for each sample PRIMA was applied resulting a projector. A fitting methodology was later applied in order to determine the coefficients of a parameter dependent projection matrix

$$V(\lambda_1, \ldots, \lambda_P) = V_0 + \lambda_1 V_1 + \ldots + \lambda_P V_P. \tag{5.7}$$

To obtain a reduced model, both the parametric system and the projector are evaluated with the parameter set. Projection is applied and the reduced model obtained. However, this reduced model is only valid for the used parameter set. If a reduced model for a different parameter set is needed, the evaluation and projection must be applied again, what makes hard to include this method in a simulation environment.

Another method combined perturbation theory with the Truncated Balanced Realization (TBR) [26, 30] framework. A perturbation matrix was theoretically obtained starting from the affine models shown in (5.6) [17]. This matrix was applied via a congruence transformation over the Gramians to address the variability, obtaining a set of perturbed Gramians. These in turn were used inside a Balancing Truncation procedure. As with most TBR-inspired methods, this one is also expensive to compute and hard to implement. The above methods have obvious drawbacks, perhaps the most glaring of which is the heavy computation cost required for obtaining the reduced models and the limitation that comes from perturbation based approximations, possibly leading to inaccuracy in certain cases.

5.1.2.2 Multi-dimensional Moment Matching

Most of the techniques in the literature extend the moment matching paradigm [13, 27, 34] to the multi-dimensional case. They usually rely on the implicit or explicit matching of the moments of the parametric transfer function (5.2). These moments depend not only on the frequency, but on the set of parameters affecting the system, and thus are denoted as multi-dimensional or multi-parameter moments.

This family of algorithms assumes that a model based on the Taylor Series expansion can be used for approximating the behavior of the conductance and capacitance, $G(\lambda)$ and $C(\lambda)$, expressed as a function of the parameters

$$\begin{aligned} G(\lambda_1, \ldots, \lambda_P) &= \textstyle\sum_{i_1=0}^{\infty} \cdots \sum_{i_P=0}^{\infty} \; G_{i_1,\ldots,i_P}\, \lambda_1^{i_1} \ldots \lambda_P^{i_P}, \\ C(\lambda_1, \ldots, \lambda_P) &= \textstyle\sum_{i_1=0}^{\infty} \cdots \sum_{i_P=0}^{\infty} \; C_{i_1,\ldots,i_P}\, \lambda_1^{i_1} \ldots \lambda_P^{i_P}, \end{aligned} \tag{5.8}$$

where $G_{i_1,\ldots,i_P}$ and $C_{i_1,\ldots,i_P}$ are the multidimensional Taylor series coefficients. This Taylor series can be extended up to the desired (or required) order, including cross derivatives, for the sake of accuracy. If this formulation is used, the structure for parameter dependence may be maintained if the projection is not only applied to the nominal matrices, but to the sensitivities as well.

Multiple methodologies follow these basic premises, but they differ in how and which such moments are generated and used in the projection stage.

The **Multi-Parameter Moment Matching** method [8] relies on a single-point expansion of the transfer function (5.2) in the joint space of the frequency s and the parameters $\lambda_1, \ldots, \lambda_P$, in order to obtain a power series in several variables,

$$x(s, \lambda_1, \ldots, \lambda_P) = \sum_{k=0}^{\infty} \sum_{k_s=0}^{k} \sum_{k_1=0}^{k-k_s} \cdots \sum_{k_P=0}^{k-k_s-k_1 \ldots -k_{P-1}} M_{k,k_s,k_1,\ldots,k_P}\; s^{k_s}\, \lambda_1^{k_1} \ldots \lambda_P^{k_P}, \tag{5.9}$$

where $M_{k,k_s,k_1,\ldots,k_P}$ is a k-th ($k = k_s + k_1 + \ldots + k_P$) order multi-parameter moment corresponding to the coefficient term $s^{k_s}\, \lambda_1^{k_1} \ldots \lambda_P^{k_P}$.

A basis for the subspace spanned from these moments can be built and the resulting orthonormal basis V can be used as a projection matrix for reducing the original system

$$\text{colspan} V = \text{colspan}\{M_{00\ldots0}, \ldots, M_{k,k_s,k_1,\ldots,k_P}\}. \tag{5.10}$$

This parametrized reduced model matches up to the k-th order multi-parameter moment of the original system.

However, the main inefficiencies of this method are twofold:

- On the one hand, this method generates pure multi-dimensional moments (see Eq. (5.9)), which means that the number of moments grows dramatically (all the possible combinations for a given order must be done) when the number of parameters is increased, even for a modest number of moments for each parameter. For this reason, the reduced model size grows exponentially with the number of parameters and the moments to match.

- On the other hand, the process parameters fluctuate in a small range around their nominal value, whereas the frequency range is much larger, and a higher number of moments are necessary in order to capture the global response for the whole frequency range. This algorithm treates the frequency as one parameter more, which turns to be highly innefficient.

An improvement of the previous approach is to perform a **Low-Rank Approximation** of the multi-dimensional moments [22]. An SVD-based low-rank approximation of the generalized moments, $G^{-1}G_i$ and $G^{-1}C_i$ (which are related to the multidimensional moments), is applied. Then, separate subspaces are built from these low-rank approximations for every parameter. The global projector is obtained from the orthonormalization of the nominal moments (computed via Arnoldi for example), and the moments of the subspaces related to the parameters. The projector is applied on the Taylor Series approximation to obtain a reduced parametric model. This approach, although providing more flexibility and improving the matching, requires the low-rank SVD of the generalized moments, which comes at a cost of $O(n^3)$, i.e., limiting its applicability to small-medium size problems.

A different multi-dimensional moment matching approach was also presented in [16], called **Passive Parameterized Time-Domain Macro Models**. It relies on the computation of several subspaces, built separately for each dimension, i.e. the frequency s (to which respect k_s block moments are obtained in a basis denoted as Q_s) and the parameter set λ (generating the basis Q_i which match k_{λ_i} block moments with respect to parameter λ_i). These independent subspaces can be efficiently computed using standard nominal approaches, e.g. PRIMA. Once all the subspaces have been computed, an orthonormal basis can be obtained so that its columns span the joint of all subspaces. For example, in the affine Taylor Series representation, using Krylov spaces $Kr(A, B, k)$ (matrix A, multi-columns vector B, moments k):

$$\begin{aligned} &\text{colsp}\,\{Q_s\} \equiv Kr\,\{A, R, k_s\} \quad \text{with} \begin{cases} A = -G^{-1}C, \\ R = G^{-1}B \end{cases} \\ &\text{colsp}\,\{Q_i\} \equiv Kr\,\{A_i, R_i, k_i\} \text{ with} \begin{cases} A_i = -(G+sC)^{-1}(G_i + sC_i), \\ R_i = -(G+sC)^{-1}B \end{cases} \\ &\qquad V = QR\ [\,Q_s\ \ Q_1\ \ \ldots\ \ Q_i\ \ \ldots Q_P], \end{aligned} \tag{5.11}$$

where subscript i refers to the i-th parameter λ_i, and the parameter related moments have been generalized to any shifted frequency s. *QR* stands for the *QR*-factorization based orthonormalization. Applying the resulting matrix V in a projection scheme ensures that the parametric Reduced Order Model matches k_s moments of the original system with respect to the frequency, and k_i moments with respect to the parameter λ_i. If the cross-term moments are needed for accuracy reasons, the subspace that spans these moments can also be included by following the same scheme.

A different approach is explored in **CORE** [23]. Here an explicit moment matching with respect to the parameters is first done, via Taylor-series expansion, followed by an implicit moment matching in frequency (via projection). The first step in done by expanding the state space vector x and the matrices G and C in its Taylor Series only with respect to the parameters,

$$x(s,\lambda) = \sum_{i_1=0}^{\infty} \cdots \sum_{i_P=0}^{\infty} x_{i_1,\ldots,i_P}(s)\, \lambda_1^{i_1} \ldots \lambda_P^{i_P}, \tag{5.12}$$

$$\begin{aligned} G(\lambda) &= \textstyle\sum_{i_1=0}^{\infty} \cdots \sum_{i_P=0}^{\infty} G_{i_1,\ldots,i_P}\ \lambda_1^{i_1} \ldots \lambda_P^{i_P}, \\ C(\lambda) &= \textstyle\sum_{i_1=0}^{\infty} \cdots \sum_{i_P=0}^{\infty} C_{i_1,\ldots,i_P}\ \lambda_1^{i_1} \ldots \lambda_P^{i_P}, \end{aligned} \tag{5.13}$$

where $G_{0,\ldots,0}, C_{0,\ldots,0}$ and $x_{0,\ldots,0}(s)$ are the nominal values for the matrices and the states vector, respectively. The remaining $G_{i_1,\ldots,i_P}$, $C_{i_1,\ldots,i_P}$ and $x_{i_1,\ldots,i_P}$ are the sensitivities with respect to the parameters. Explicitly matching the coefficients of the same powers leads to an augmented system, in which the parametric dependence is shifted to the output related matrix L_A:

$$C_A = \begin{bmatrix} C_0 & & & & \\ C_1 & C_0 & & & \\ \vdots & 0 & C_0 & & \\ C_i & 0 & 0 & C_0 & \\ \vdots & & & & \ddots \end{bmatrix}, \qquad B_A = \begin{bmatrix} B \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix}, \qquad x_A = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_i \\ \vdots \end{bmatrix},$$

$$G_A = \begin{bmatrix} G_0 & & & & \\ G_1 & G_0 & & & \\ \vdots & 0 & G_0 & & \\ G_i & 0 & 0 & G_0 & \\ \vdots & & & & \ddots \end{bmatrix}, \; L_A = [L \;\; \lambda_1 L \;\; \cdots \;\; \lambda_i L \;\; \cdots]. \tag{5.14}$$

The second step applies a typical nominal moment matching procedure (e.g. PRIMA [27]) to reduce this augmented system. This is possible because the matrices G_A, C_A and B_A used to build the projector do not depend on the parameters. The projector is latter applied on all the matrices of the augmented system in (5.14). Furthermore, the Block Lower Triangular structure of the system matrices G_A and C_A can be exploited in recursive algorithms to speed-up the reduction stage. This two-step approach allows to increase the number of the matched multi-parameter moments with respect to other techniques, for a similar reduced order. In principle, in spite of the larger size of the augmented model, the order of the reduced system can be much smaller than in the previous cases. On the other hand, the structure of the dependence with respect to the parameters is lost, since the parametric dependence is shifted to the later projected output related L matrix. The projection mixes all the parameters, losing the possibility of modifying them without need of recomputation. This method also has the disadvantage that the explicit computation of the moments with respect to the parameters can lead to numerical instabilities. The method, although stability-preserving, is unable to guarantee passivity preservation.

Some algorithms [24, 37] try to match the same moments as CORE, but in a more numerical stable and efficient fashion, using **Recursive and Stochastic Approaches**. They generalize the CORE paradigm up to an arbitrary expansion order with respect to the parameters, and apply an iterative procedure in order to compute the frequency moments related to the nominal matrices, and the ones obtained from the parametric part (this means, to obtain a basis for each block of states x_i in (5.14), but without building such system).

$$\begin{aligned}
&\text{colspan}\left\{V_0\right\} \equiv Kr\left\{A, R, q_0\right\} = \left[V_0^0\ V_0^1\ \ldots\ V_0^{q-1}\right],\\
&\text{with}\quad A = -\left(G_0 + s_k C_0\right)^{-1} C_0, \quad R = \left(G_0 + s_k C_0\right)^{-1} B,\\
&\text{colspan}\left\{V_i\right\} = \left[V_i^0\ V_i^1\ \ldots V_i^j \ldots\right],\\
&\text{with}\quad V_i^j = -\left(G_0 + s_k C_0\right)^{-1}\left(G_i V_0^j + s_k C_i V_0^{j-1} + C_0 V_i^{j-1}\right),\\
&G_i = G_{0\ldots010\ldots0},\\
&C_i = C_{0\ldots010\ldots0},
\end{aligned} \tag{5.15}$$

where s_k is the expansion point for the Krylov subspace generation, and V_i^j is the j-th moment with respect to the frequency for the i-th parameter. This general recursive scheme, here presented for first order with respect to the parameters, can be extended to any (independent) order with respect to each parameter.

The technique in [37] uses a tree graph scheme, in which each node is associated to a moment, and the branches represent recursive dependences among moments. Each tree level contains all the moments of the same multi-parameter order. On this tree, a random sampling approach is used to select and generate some representative moments, preventing the exponential growth.

On the other hand, the technique in [24] advocates for an exhaustive computation at each parameter order. This means that all the moments for zero-parameter order (i.e. nominal), are computed until no rank is added. The same procedure is repeated for first order with respect to all parameters. If the model is not accurate, more order with respect to the parameters can be added.

Notice that both schemes provide a large degree of flexibility, as different orders with respect to each parameter and with respect to the frequency can be applied. In both approaches, the set of all the moments generated is orthonormalized, so an overall projector is obtained. This is used inside a congruence transformation on the Taylor Series approximation (5.4), to generate a reduced model in the same representation. Another advantage of these methodologies is that the passivity is PRIMA-like preserved, and the basis is built in a numerical stable fashion.

5.1.2.3 Multi-dimensional Sampling

Another option present in the literature relies on sampling schemes for capturing the variational nature of the parametric model. They are applied for the building of a projector to later apply congruence tranformation on the original model.

A simple generalization of the multi-point moment matching framework [11] to a multi-dimensional space can be done via **Variational Multi-Point Moment Matching**. Small research has been devoted to this family of approaches, but one algorithm can be found in [22]. The flexibility it provides is also one of its main drawbacks, as the methodology can be applied in a variety of schemes, from a single-frequency multi-parameter sampling to a pure multi-dimensional sampling. From these expansion points, several moments are computed following a typical moment matching scheme. The orthonormalization of the set of moments provides the overall projector which is applied in a congruence reduction scheme. However, it is hard to determine the number and placement of samples, and the number of moments to match with respect to the frequency and to the parameters.

Another scheme, which overcomes some of the issues of the previous approach is the **Variational Poor Man's TBR** [28]. This approach is based on the statistical interpretation of the algorithm (see [29] for details) and enhances its applicability to multiple dimensions. In this interpretation, the Gramian X_λ is seen as a covariance matrix for a Gaussian variable x_λ, obtained by exciting the (presumed stable) system with u involving white noise. Rewriting the Gramian as

$$X_\lambda = \int_{S_\lambda} \int_{-\infty}^{\infty} (j\omega C_\lambda + G_\lambda)^{-1} \; BB^T (j\omega C_\lambda + G_\lambda)^{-H} \; p(\lambda)\, d\omega d\lambda, \tag{5.16}$$

where $p(\lambda)$ is the probability density of λ in the parameter space, S_λ. Just as in PMTBR, a quadrature rule can be applied in the overall parameter plus frequency space to approximate the Gramian via numerical computation. But in this case the weights are chosen taking into account the *Probability Density Function* (PDF) of λ_i and the frequency constraints. This can be generalized to a set of parameters, where a joint PDF of all the parameters can be applied on the overall parameter space, or the individual PDF of each parameter can be used. This possibility represents an interesting advantage, since a-priori knowledge of the parameters and the frequency can be included in order to constrain the sampling and yield a more accurate reduced model. The result of this approach is an algorithm which generates Reduced Order Models whose size is less dependent on the number of parameters. In the deterministic case, an error analysis and control can be included, via the eigenvalues of the SVD. However, in the variational case only an expected error bound can be given:

$$E\{\|\hat{x}_\lambda(0) - x_\lambda(0)\|_2^2\} \leq \sum_{i=r+1}^{n} \sigma_i^2, \tag{5.17}$$

where r is the reduced order and n the original number of states. On the other hand, in this method the issue of sample selection, already an important one in the deterministic version, becomes even more relevant, since the sampling must now be done in a potentially much higher-dimensional space.

5.1.3 Practical Consideration and Structure Preservation

Inside the pMOR realm, the moment matching algorithms based on single point expansion may not be able to capture the complete behavior along the large frequency range required for common RF systems, and may lead to excessively large models if many parameters are taken into account. Therefore the most suitable techniques for the reduction seem to be the multipoint ones. Among those techniques, Variational PMTBR [28] offers a reliable framework with some interesting features that can be exploited, such as the inclusion of probabilistic information and the trade off between size and error, which allows for some control of the error via analysis of the singular values related to the dropped vectors. On the other hand, it requires a higher computational effort than the multi-dimensional moment matching approaches, as it is based on multidimensional sampling schemes and Singular Value Decomposition (SVD), but the compression ratio and reliability that it offers compensates this drawback. The effort spent in the generation of such models can be amortized when the reduced order model generated is going to be used multiple times. This is usually the case for parametric models, as the designer may require several evaluations for different parameter sets (e.g. in the case of Monte Carlo simulations, or optimization steps). Furthermore, this technique offers some extra advantages when combined with block structured systems [14], such as the block-wise error control with respect to the global input-output behaviour, which can be applied to improve the efficiency of the reduction. This means that each block can be reduced to a different order depending on its relevance in the global response.

An important point to recall here is that the block division may not reflect different sub-domains. Different sub-divisions can be done to address different hierarchical levels. For instance, it may be interesting to divide the complete set in sub-domains connected by hooks, which generates a block structured matricial representation. But inside each block corresponding to a sub-domain, another block division may be done, corresponding either to smaller sub-domains or to a division related to the different kind of variables used to model each domain (for example, in a simple case, currents and voltages). This variable related block structure preservation has already been advocated in the literature [15] and may help the synthesis of and equivalent SPICE-like circuit [35] in the case that is required. Figure 5.1 presents a more intuitive depiction of the previous statements, in which a two domain example is shown with its hierarchy, and each domain has also some inner hierarchy related to the different kind of variables (in this case, voltages and currents, but it can also be related to the electric and magnetic variables, depending on the formulation and method used for the generation of the system matrices).

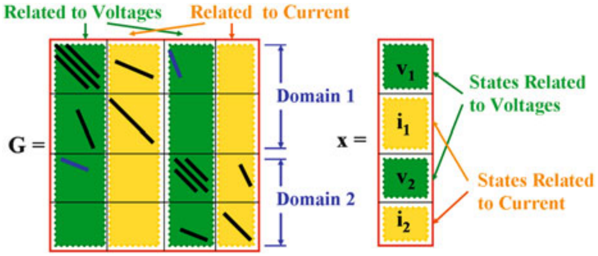


Fig. 5.1 Two-level hierarchy: domain level (given by the numbers, 1 and 2) and variable level (voltages v_k and currents i_k)

The proposed flow starts from a parametric state-space descriptor, such as (5.1), which exhibits a multi-level hierarchy, and a block parametric dependence (as different parameters may affect different sub-domains). The matrices of size n have K domains, each with size n_i, $n = \sum_i n_i$. For instance, for the static part,

$$G = \begin{bmatrix} G_{11}(\lambda_{\{11\}}) & \dots & G_{1K}(\lambda_{\{1K\}}) \\ \vdots & \ddots & \vdots \\ G_{K1}(\lambda_{\{K1\}}) & \dots & G_{KK}(\lambda_{\{KK\}}) \end{bmatrix}, \tag{5.18}$$

where $\lambda_{\{ij\}}$ is the set of parameters affecting $G_{ij} \in \mathbb{R}^{n_i \times n_j}$. Then we perform the multi-dimensional sampling, both in the frequency and the parameter space. For each point we generate a matrix or vector z_j (a matrix in case B includes multiple inputs)

$$z_j = \left(C(\lambda_j)s_j + G(\lambda_j)\right)^{-1} B, \tag{5.19}$$

where $C(\lambda)$ and $G(\lambda)$ are the global matrices of the complete domain, with n degrees of freedom (dofs). To generate the matrix $z_j \in \mathbb{R}^{n \times m}$, with m the number of global ports, we can apply a direct procedure, meaning a factorization (at cost $O(n^\beta)$, with $1.1 \leq \beta \leq 1.5$ for sparse matrices) and a solve (at cost $O(n^\alpha)$, with $1 \leq \alpha \leq 1.2$ for sparse matrices). Novel sparse factorization schemes can be applied to improve the efficiency [9, 10]. In cases when a direct method may be too expensive iterative procedures may be used [32].

The choice of the sampling points may be an issue, as there is no clear scheme or procedure that is known to provide an optimal solution. However, as stated in [28], the accuracy of the method does not depend on the accuracy of the quadrature (and thus in the sampling scheme), but on the subspace generated. For this reason, a good sampling scheme is to perform samples in the frequency for the nominal system, and around these nominal samples, perform some parametric random sampling in order to capture the vectors that the perturbed system generates. The reasoning behind this scheme is that for small variations, such as the ones resulting from process parameters, the subspace generated along the frequency is generally more dominant than the one generated by the parameters. In addition, under small variations, the nominal sampling can be used as a good initial guess for an iterative solver to generate the parametric samples. For the direct solution scheme, to generate P samples (and thus Pm vectors) for the global system has a cost of $O(Pn^{\alpha} + Pn^{\beta})$. Note that since m is the number of global (or external) ports, the number of vectors is smaller than if we take all the hooks into account.

The next step is the orthonormalization, via SVD, of the Pm vectors for generating a basis of the subspace in which to project the matrices. Here an **independent basis** V_i, $i \in \{1, \ldots, K\}$, can be generated for each i-th sub-domain. To this end the columns in z_j are split according to the block structure present in the system matrices (i.e., the n_i rows for each block), and an SVD is performed on each of these set of vectors, at a cost of $O(n_i(Pm)^2)$, where n_i is the size of the corresponding block, and $n = \sum_i n_i$. For each block, the independent SVD allows to drop the vectors less relevant for the global response (estimated by the dropped singular value ratio, as presented in [28]). This step generates a set of projectors, $V_i \in \mathbb{R}^{n_i \times q_i}$, with $q_i \ll n_i$ the reduced size for the i-th block of the global system matrix. These projectors can be placed in the diagonal blocks of an overall projector, that can be used for reducing the initial global matrices to an order $q = \sum_i q_i$. This block diagonal projector allows a block structure (and thus sub-domain) preservation, increasing the sparsity of the ROM with respect to that of the standard projection. This sparsity increase is particularly noticeable in the case of the sensitivities (if a Taylor series is used as base representation), as the block parameter dependence is maintained (e.g. in the static matrix)

$$\hat{G}_{ij}(\lambda_{\{ij\}}) = V_i^T G_{ij}(\lambda_{\{ij\}}) V_j . \tag{5.20}$$

The total cost for the procedure can be approximated by

$$O(Pn^{\alpha} + Pn^{\beta} + (Pm)^2 \textstyle\sum_i n_i). \tag{5.21}$$

5.1.4 Examples

5.1.4.1 L-Shape

As a first example we present a simple L-shape interconnect structure depending on the width of the metal layer. Figure 5.2 shows the frequency response for a fixed parameter value, of the nominal system, the Taylor series approximation (both of order 313), and the reduction models obtained with several parametric approaches:

- Nominal reduction of the Taylor Series, via PRIMA, of order 25,
- Multi-dimensional moment matching, via CORE, of order 25,

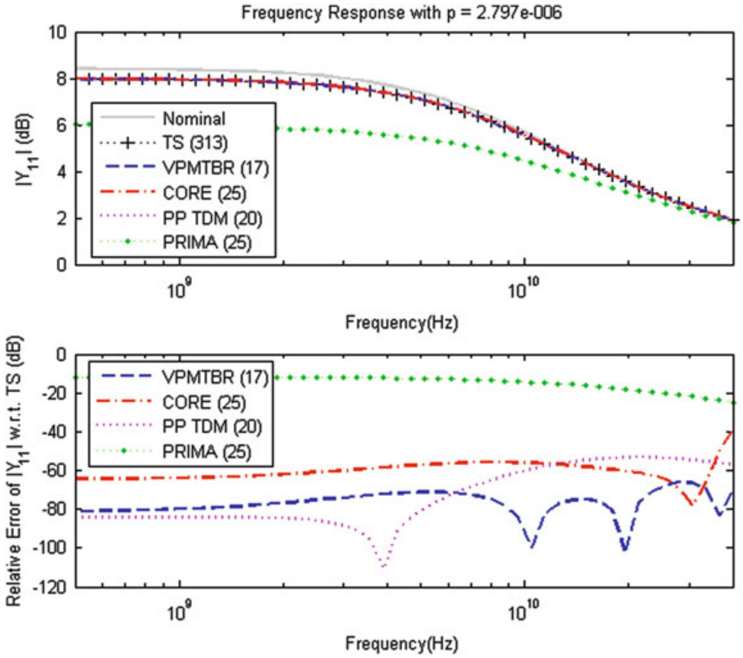


Fig. 5.2 (*Top*): Frequency response of the L-shape example. The original, both the nominal and the Taylor series for a fixed parameter value, of order 313, and the reductions via PRIMA, CORE, Passive Parameterized Time-Domain Macro Models (PP TDM), and variational PMTBR (VPMTBR), of different orders. (*Bottom*): Relative error of the reduction models with respect to the original Taylor series approximation

- Multi-dimensional moment matching, via Passive Parameterized Time-Domain Macro Models technique, of order 20,
- And Multi-dimensional sampling, via Variational PMTBR, of order 16.

Figure 5.3 shows the same example, but in this case the response of the systems with respect of the parameter variation, for a given frequency point. It is clear that the parametric Model Order Reduction techniques are able to capture the parametric behavior, whereas the nominal approach (PRIMA) fails to do so, even for high order.

5.1.4.2 U-Coupled

This is a simple test case, which has two U-shape conductors; each of the conductors ends represent one port, having one terminal voltage excited (intentional terminal, IT) and one terminal connected to ground. A clear illustration of the setting is given by Fig. 5.4. The distance (d) separating the conductors and the thickness (h) of the corresponding metal layer are parameterized. The complete domain is partitioned into three sub-domains, each of them connected to the others via a set of hooks (both

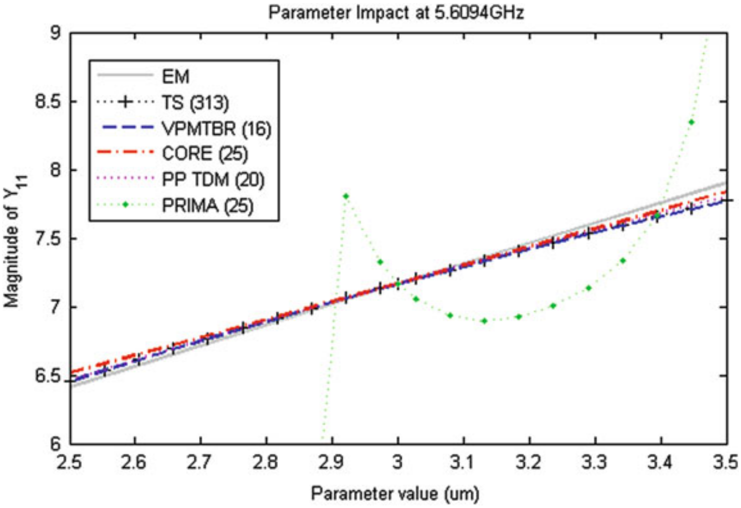


Fig. 5.3 Parameter impact on the response of the L-shape example. The EM model for several parameter values (of order 313), the Taylor series approximation (of order 313 as well), and the reductions via PRIMA, CORE, PP TDM, and VPMTBR, of different orders

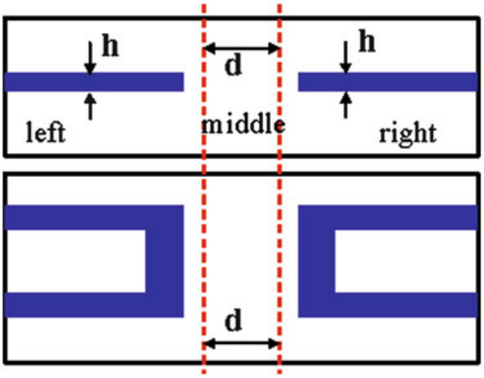


Fig. 5.4 Topology of the U-shape: (Up) cross view, (Down) top view. Parameters: distance between conductors, d, and thickness of the metal, h

electric, EH, and magnetic, MH). The domain hierarchy and parameter dependence are kept after the reduction, via Block Structure Preserving approaches. The Full Wave EM model was obtained via Finite Integration Technique (FIT) [7], and its matrices present a Block Structure that follows the domain partitioning. Table 5.1 shows the characteristics of the original system. Each sub-domain is affected by a parameter. The left and right sub-domains contain the conductors, and thus are affected by the metal thickness h. The middle domain width varies with the distance between the two conductors, and thus is affected by parameter d. For each parameter

Table 5.1 Characteristics of the examples

Ex	Domain	Dofs	Terminals (EH,MH,IT)	ROM Dofs
U-shape	Left	785	77 (42, 34, 1)	85
	Middle	645	152 (84, 68, 0)	90
	Right	785	77 (42, 34, 1)	85
	Complete	2,215	2 (0, 0, 2)	260
Double Spiral	Var_1	49,125	2 (0, 0, 2)	142
	Var_2	54,977	2 (0, 0, 2)	165
	Complete	104,102	2 (0, 0, 2)	307

the first order sensitivity is taken into account, and a first order Taylor Series (TS) formulation is taken as the original system.

For the reduction we apply three techniques. First, a Nominal Block Structure Preserving (BSP) PRIMA [36], with a single expansion point and matching 50 moments, is applied. This leads to a 100-vector generated basis, that after BSP expansion produces a 300-dofs Reduced Order Model (ROM). Second, a BSP procedure coupled with a Multi-Dimensional Moment Matching (MDMM) approach [16], is tried. The basis will match 40 moments with respect to the frequency, and 30 moments with respect to each parameter. The orthonormalized basis has 196 vectors, that span a BSP ROM of size 588. Third, the proposed BSP VPMTBR, with 60 multidimensional samples, and a relative tolerance of 0.001 for each block, is studied. This process generates different reduced sizes for each block: 85, 90 and 85, with a global size of 260.

Figure 5.5 shows the relative error in the frequency transfer function at a parameter set point for the three ROMs w.r.t. the Taylor series. PRIMA and MDMM approaches fail to capture the behavior with the order set, but the proposed approach performs much better even for a lower order. Figure 5.6 shows the response change with the variation of parameter d at a single frequency point (Parameter Impact). PRIMA and MDMM only present accuracy for the nominal point, whereas the proposed method maintains the accuracy for the parameter range.

5.1.4.3 Double Spiral

This is an industrial example, composed by two square integrated spiral inductors in the same configuration as the previous example (See Fig. 5.7). The complete domain has two ports, and 104,102 Dofs. The example also depends on the same two parameters, the distance d between spirals, and the thickness h of the corresponding metal layer. In this case a single domain is used, but the BSP approach is applied on the inner structure provided by the different variables in the FIT method (electric and magnetic grid). For the reduction, the proposed BSP VPMTBR methodology is benchmarked against a nominal BSP PRIMA (400 dofs) methodology, and compared with the original Taylor Series formulation. The ROM size in this case

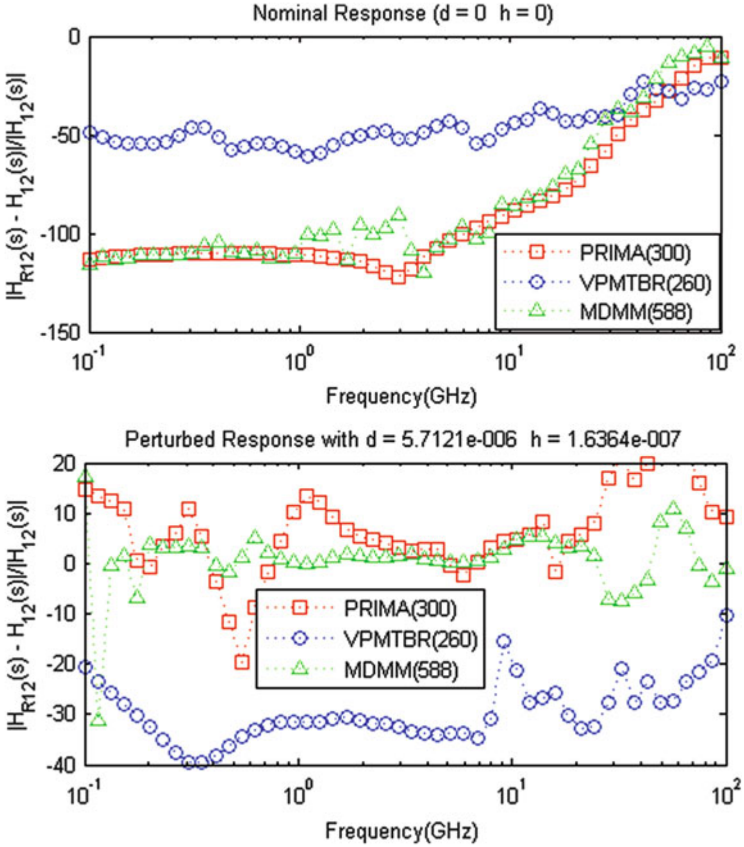


Fig. 5.5 U-coupled: Relative error (dB) in $|H_{12}(s)|$ for (Up) the nominal response, and (Down) the perturbed response at a single parameter set. The curves represent: BSP PRIMA, BSP VPMTBR, and BSP MDMM

is 142 and 165 respectively for the blocks. The results are presented in the Figs. 5.8 and 5.9. Figure 5.8 shows the frequency relative error of the ROMs with respect to the original Taylor Series. PRIMA, although accurate for the nominal response, fails to capture the parametric behavior, whereas the proposed method succeeds in modeling such behavior. This is also the conclusion that can be drawn from the parameter impact in Fig. 5.9.

5.1.5 Conclusions

We conclude that Parametric Model Order Reduction techniques are essential for addressing parameter variability in the simulation of large dynamical systems.

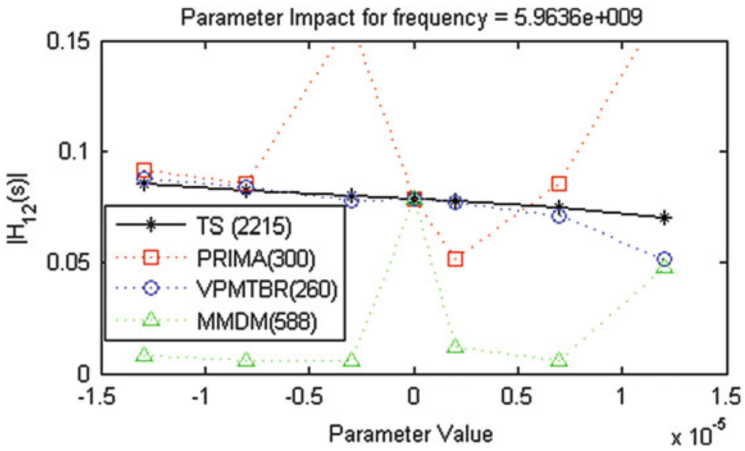


Fig. 5.6 U-coupled: Variation of $|H_{12}|$ vs. the variation of the parameter d at 59.6 GHz for the original TS and the three BSP ROMs

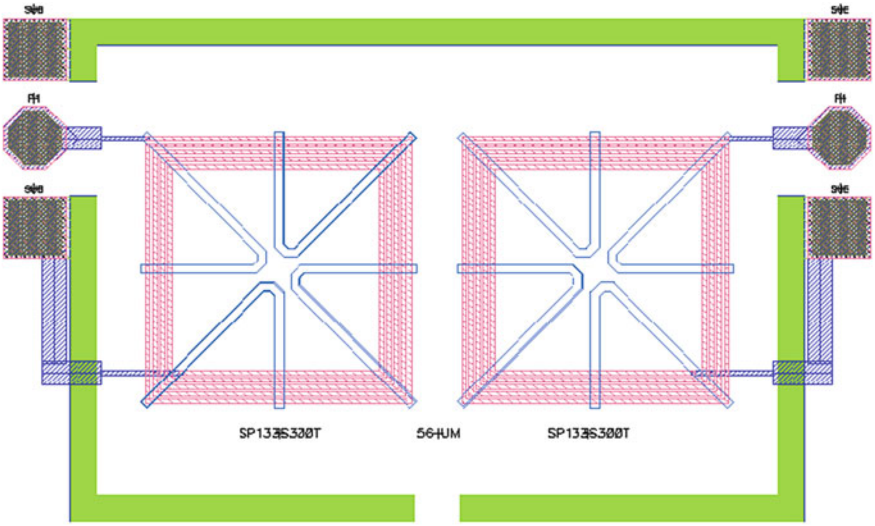


Fig. 5.7 Layout configuration of the Double Spiral example (view from the *top*)

Representation of the state space based on Taylor series expansion with respect to the parameters provide the flexibility and accuracy required by efficient simulation. This reresentation approach can be combined with projection-based methods to generate structural equivalent reduced models.

Single-point based moment-matching approaches are suitable for small variations and local approximations, but usually suffer from several drawbacks when applied to EM based models operating in a wide frequency range. Multi-point

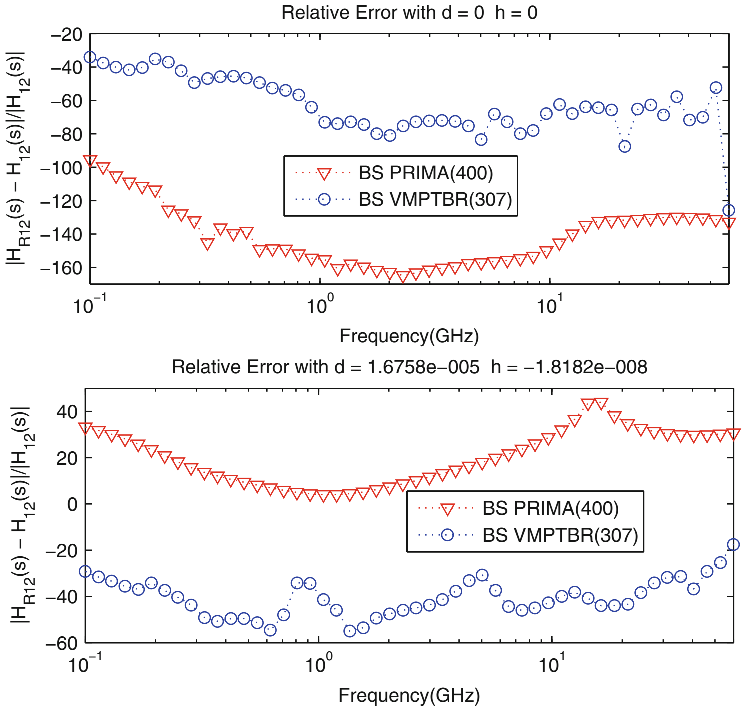


Fig. 5.8 Double Spiral: Relative error (dB) in $|H_{12}(s)|$ for (Up) the nominal response, and (Down) the perturbed response at a single parameter set

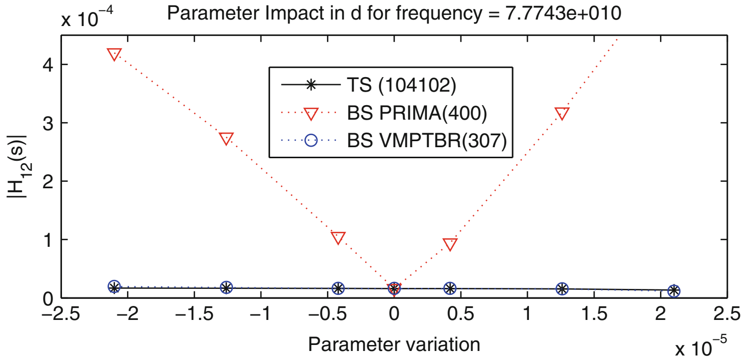


Fig. 5.9 Double Spiral: $|H_{12}|$ vs. the variation of the parameter d at a frequency point for the original TS and the ROMs: PRIMA, and VPMTBR

based approaches, although computationally more expensive, are more reliable and generate more compressed models. Thus, the generation cost can be amortized in the simulation stages.

Combination of the projection methodologies with Block Structure Preserving approaches can be done efficiently in parametric environments. Further advantages can be obtained in this case, such as different compression order for each block based on its relevance in the global behavior, higher degree of sparsification of the nominal matrices, and in particular, of the sensitivities, and the maintenance of the block domain hierarchy and block parameter dependence after reduction.

5.2 Robust Procedures for Parametric Model Order Reduction of High Speed Interconnects

Due to higher integration and increasing of running frequency, full Electromagnetic Models (EM) are needed for an accurate prediction of the real behavior of integrated passives and interconnects in currently designed chips [45].[2] In general, if on-chip interconnects are sorted with respect to their electric length, they may be categorized in three classes: short, medium and long. While the short interconnects have simple circuit models with lumped parameters, the extracted model of the interconnects longer than the wave length has to consider the effect of the distributed parameters, as well. Fortunately, the long interconnects have usually the same cross-sectional geometry along their extension. If not, they may be decomposed in straight parts connected by junction components (Fig. 5.10). The former are represented as transmission lines (TLs) whereas the latter are modeled as common passive 3D components.

Due to the fact that the lithographic technology is pushed today to work at its limit, the variability of geometrical and physical properties cannot be neglected.

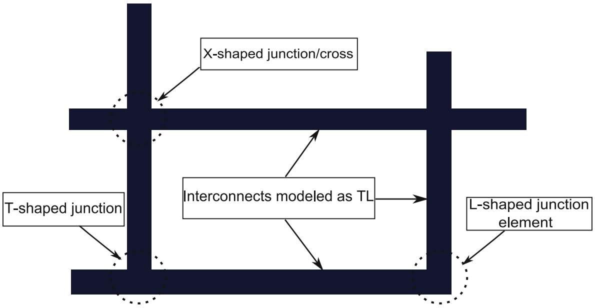


Fig. 5.10 Decomposition of the interconnect net in 2D TLs and 3D junctions

[2] Section 5.2 has been written by: Gabriela Ciuprina, Alexandra Ştefănescu, Sebastian Kula and Daniel Ioan. For additional topics see also the Ph.D.-Theses of the second author [59] and of the third author [56].

That is why, to obtain robust devices, the variability analysis is necessary even during the design stage [44, 55].

This Section proposes a robust specialized technique to extract reduced parametric compact models, described as parametric SPICE like netlists, for long interconnects modeled as transmission lines with several field effects such as skin effect and substrate losses. The technique uses an EM formulation based on partial differential equations (PDE), which is discretized to obtain a finite state space model. Next, a variability analysis of the geometrical data is carried out. Finally, a method to extract an equivalent parametric circuit is proposed. The procedure is validated by applying it on a study case for which experimental data is available.

5.2.1 Field Problem Formulation: 3D – PDE Models

Long interconnects and passive components with significant high frequency field effects, have to be modeled by taken into consideration Full Wave (FW) electromagnetic field equations. Typical examples of such parasitic effects are: skin effect and proximity, substrate losses, propagation retardation and crosstalk. Only Maxwell equations in FW regime

$$\begin{array}{rclc} \operatorname{curl}\boldsymbol{H} &=& \boldsymbol{J}+\frac{\partial \boldsymbol{D}}{\partial t}, & \operatorname{div}\boldsymbol{B}=0, \\ \operatorname{curl}\boldsymbol{E} &=& -\frac{\partial \boldsymbol{B}}{\partial t}, & \operatorname{div}\boldsymbol{D}=\rho, \end{array} \tag{5.22}$$

complemented with the constitutive equations which describe the material behavior:

$$\boldsymbol{B}=\mu\boldsymbol{H}, \quad \boldsymbol{D}=\varepsilon\boldsymbol{E}, \quad \boldsymbol{J}=\sigma\boldsymbol{E}, \tag{5.23}$$

can model these effects. While material constants are known for each subdomain (Si, Al, SiO_2), vectorial fields $\boldsymbol{B}, \boldsymbol{H}, \boldsymbol{E}, \boldsymbol{D} : \Omega \times [0, T) \rightarrow \mathbb{R}^3$ and the scalar field $\rho : \Omega \times [0, T) \rightarrow \mathbb{R}$ are the unknowns of the problem. They can be univocal determined in the simple connected set Ω, which is the computational domain, for zero initial conditions ($\boldsymbol{B} = \boldsymbol{0}, \boldsymbol{D} = \boldsymbol{0}$ for $t = 0$), if appropriate boundary conditions are imposed.

According to authors' knowledge, the best boundary conditions which allow the field-circuit coupling are those given by the electric circuit element (ECE) formulation [54]. Considering $S_1', S_2', \ldots, S_n' \subset \partial\Omega$ a disjoint set of surfaces, called terminals (Fig. 5.11), the following boundary conditions are assumed:

$$\boldsymbol{n}\cdot\operatorname{curl}\boldsymbol{E}=0 \quad \text{on} \quad \partial\Omega, \tag{5.24}$$

$$\boldsymbol{n}\cdot\operatorname{curl}\boldsymbol{H}=0 \quad \text{on} \quad \partial\Omega\setminus\cup_{k=1}^{n} S_k' \tag{5.25}$$

$$\boldsymbol{n}\times\boldsymbol{E}=\boldsymbol{0} \quad \text{on} \quad \cup_{k=1}^{n} S_k' \tag{5.26}$$

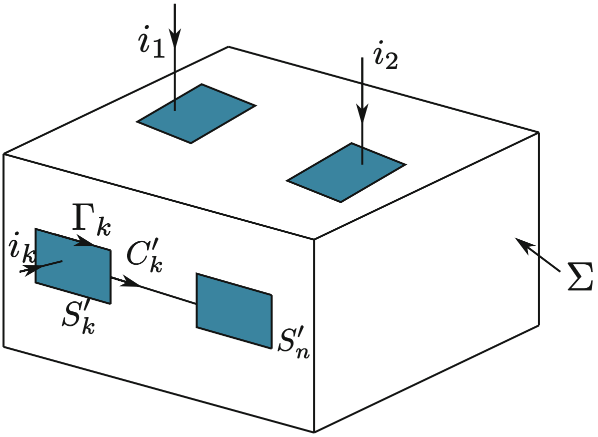


Fig. 5.11 ECE – electric circuit element with multiple terminals

Condition (5.24) interdicts the magnetic coupling between the domain and its environment, (5.25) interdicts the galvanic coupling and the capacitive coupling through the boundary excepting for the terminals and (5.26) interdicts the variation of the electric potential over the terminal, thus allowing the connection of the device to exterior electric circuit nodes. For each terminal, $k = 1, \ldots, n$ the voltage and the current can be univocal defined:

$$u_k = \int_{C_k \subset \partial\Omega} \boldsymbol{E} \cdot \mathrm{d}\boldsymbol{r}, \quad i_k = \int_{\partial S'_k} \boldsymbol{H} \cdot \mathrm{d}\boldsymbol{r}, \tag{5.27}$$

where C'_k is an arbitrary path on the device boundary $\partial\Omega$, that starts on S'_k and ends on S'_n, where, by convention, the n-th terminal is considered as reference, i.e. $u_n = 0$. If we assume that the terminals are excited in voltage, then u_k, $k = 1, 2, \ldots, n-1$ are input signals and i_k, $k = 1, 2, \ldots, n-1$ are output signals. Equations (5.24) $\div$ (5.26) define a multiple input multiple output (MIMO) linear system with $n-1$ inputs and $n-1$ outputs, but with a state space of infinite dimension. In the weak form of Maxwell's equations, state variables, $\boldsymbol{H}$, $\boldsymbol{E}$ belong to the Sobolev space $H(\text{curl}, \Omega)$ [39]. Uniqueness theorem of the ECE field problem [54] generates the correct formulation of the transfer function $\mathbf{Y}(s) : \mathbb{C} \to \mathbb{C}^{(n-1)\times(n-1)}$, which represents the matrix of the terminals admittance for a complex frequency s. The relation

$$\mathbf{i} = \mathbf{Y}\mathbf{u} \tag{5.28}$$

defines a linear transformation in the frequency domain of the terminal voltages vector $\mathbf{u} \in \mathbb{C}^{n-1}$ to the currents vector $\mathbf{i} \in \mathbb{C}^{n-1}$.

5.2.2 Numeric Discretization and State Space Models

PDE models are too complex for designers needs. The approach we propose for the extraction of the electric models is schematically represented in Fig. 5.12. The left block corresponds to the formulation described in the previous section.

The next important step in the EM modeling is the discretization of the PDEs. One of the simplest methods to carry out this, is based on the Finite Integration Technique (FIT), a numerical method able to solve field problems based on spatial discretization "without shape functions". Two staggered orthogonal (Yee type) grids are used as discretization mesh [42]. The centers of the primary cells are the nodes of the secondary cells. The degrees of freedom (dofs) used by FIT are not local field components as in FEM or in FDTD, but global variables, i.e., electric and magnetic voltages $\mathbf{u}_e$, $\mathbf{u}_m$, electric currents $\mathbf{i}$, and magnetic and electric fluxes ϕ, ψ assigned to the grid elements: edges and faces, respectively. They are associated to these grids elements in a coherent manner (Fig. 5.13).

By applying the global form of electromagnetic field equations on the mesh elements (elementary faces and their borders), a system of differential algebraic equations (DAE), called Maxwell Grid Equations (MGE) is obtained:

$$\mathrm{curl}\mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \quad \Rightarrow \int_{\Gamma} \mathbf{E} d\mathbf{r} = -\int\int_{S_{\Gamma}} \frac{\partial \mathbf{B}}{\partial t} d\mathbf{A} \quad \Rightarrow \mathbf{C}\mathbf{u}_e = -\frac{d\varphi}{dt} \tag{5.29}$$

$$\hookrightarrow \mathrm{div}\mathbf{B} = 0 \quad \Rightarrow \int\int_{\Sigma} \mathbf{B} d\mathbf{A} = 0 \quad \Rightarrow \mathbf{D}'\varphi = 0 \tag{5.30}$$

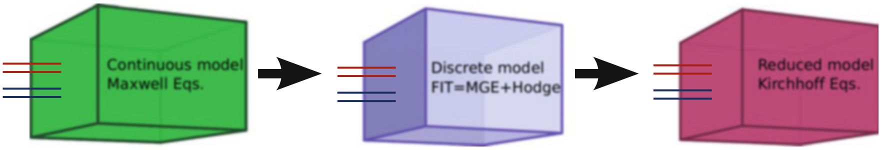


Fig. 5.12 Three levels of abstraction for a component model and its corresponding equations

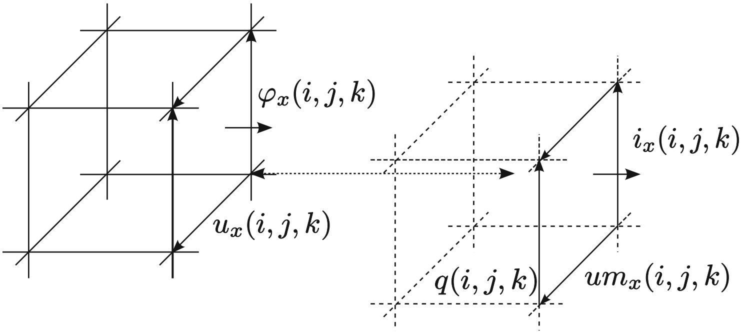


Fig. 5.13 Dofs for FIT numerical method in the two dual grids cells

$$\mathrm{curl}\mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \Rightarrow \int_{\Gamma} \mathbf{H} d\mathbf{r} = \int\int_{S_{\Gamma}} (J + \tfrac{\partial \mathbf{D}}{\partial t}) d\mathbf{A} \Rightarrow \mathbf{C}'\mathbf{u}_m = \mathbf{i} + \frac{d\boldsymbol{\psi}}{dt} \tag{5.31}$$

$$\hookrightarrow \mathrm{div}\mathbf{D} = \boldsymbol{\rho} \quad \Rightarrow \int\int_{\Sigma} \mathbf{D}d\mathbf{A} = \int\int\int_{D_{\Sigma}} \rho d\mathbf{v} \quad \Rightarrow \mathbf{D}\boldsymbol{\psi} = \mathbf{q} \tag{5.32}$$

$$\hookrightarrow \mathrm{div}\mathbf{J} = -\frac{\partial \rho}{\partial t} \Rightarrow \int\int_{\Sigma} \mathbf{J}d\mathbf{A} = -\int\int\int_{D_{\Sigma}} \tfrac{\partial \rho}{\partial t} d\mathrm{v} \Rightarrow \mathbf{Di} = -\frac{d\mathbf{q}}{dt} \tag{5.33}$$

FIT combines MGE with Hodge's linear transform operators, which approximate the material behavior (5.23):

$$\boldsymbol{\varphi} = \mathbf{G_m}\mathbf{u}_m, \quad \boldsymbol{\psi} = \mathbf{C_e}\mathbf{u}_e, \quad \mathbf{i} = \mathbf{G_e}\mathbf{u}_e. \tag{5.34}$$

The main characteristics of the FIT method are:

- There is no discretization error in the MGE fundamental Eqs. (5.29) ÷ (5.33). All numerical errors are hold by the discrete Hodge operators (5.34).
- An equivalent FIT circuit (Fig. 5.14), having MGE + Hodge as equations may be easily build. The graphs of the two constituent mutually coupled sub-circuits are exactly the two dual discretization grids; therefore the complexity of the equivalent circuit has a linear order with respect to the number of grid-cells [49].
- MGE are:
  - **Sparse**: matrices $\boldsymbol{G}_m, \boldsymbol{C}_e$ and $\boldsymbol{G}_e$ are diagonal and matrices $\boldsymbol{C}, \boldsymbol{D}$ have maximum six non-zero entries per row,
  - **Metric-free**: matrices **C** – the discrete-curl and **D** – the discrete-div operators have only 0, $+1$ and -1 as entries,
  - **Mimetic**: in Maxwell equations curl and div operators are replaced by their discrete counterparts **C** and **D**, and
  - **Conservative**: the discrete form of the discrete charge conservation equation is a direct consequence of both Maxwell and as well as of the MGE equations.

Due to these properties the numerical solutions have **no spurious modes**.

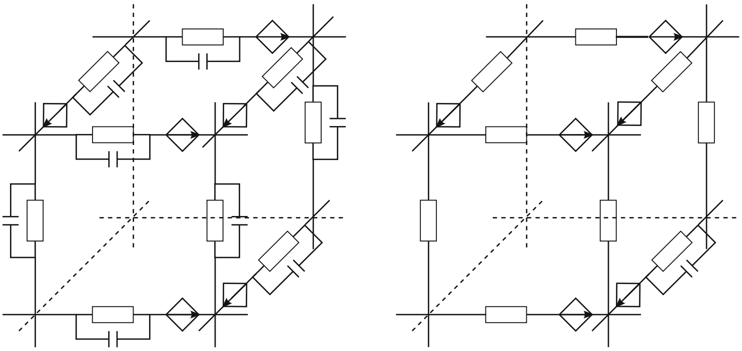


Fig. 5.14 Electric (*left*) and magnetic (*right*) equivalent FIT circuits

Considering FIT Eqs. (5.29), (5.31), and (5.34) with the discrete forms of boundary conditions (5.24) ÷ (5.27) a linear time-invariant system is defined having the same input-output quantities as (5.28), but the state equations:

$$\boldsymbol{C}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} + \boldsymbol{G}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{u}, \qquad \boldsymbol{i} = \boldsymbol{L}\boldsymbol{x}, \tag{5.35}$$

where $\boldsymbol{x} = [\boldsymbol{u}_m^T, \boldsymbol{u}_e^T, \boldsymbol{i}^T]^T$ is the state space vector, consisting of electric voltages $\boldsymbol{u}_e$ defined on the electric grid used by FIT, magnetic voltages $\boldsymbol{u}_m$ defined on the magnetic grid and output quantities $\boldsymbol{i}$. Equations can be written such that only two semi-state space matrices ($\boldsymbol{C}$ and $\boldsymbol{G}$) are affected by geometric parameters (denoted by α in what follows). Considering all terminals voltage-excited, the number of inputs is always equal to the number of outputs. Since output currents are components of the state vector, the matrix $\boldsymbol{L} = \boldsymbol{B}^{\boldsymbol{T}}$ is merely a selection matrix.

For instance, the structure of the matrices in the case of voltage excitation is the following:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{G}_m(\alpha) & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{C}_i(\alpha)\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{SI}(\alpha) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{C}_{TE}(\alpha) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{G} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{B}_1\ \boldsymbol{B}_2 & \boldsymbol{0} \\ \boldsymbol{B}_1^T & -\boldsymbol{G}_i(\alpha)\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}\ \boldsymbol{B}_{SI} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_{SI}(\alpha) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_{TE}(\alpha) & -\boldsymbol{S}_E \\ \boldsymbol{0} & \boldsymbol{P}_E & \boldsymbol{0} \end{bmatrix} \tag{5.36}$$

There are six sets of rows, corresponding to the six sets of equations. The first group of equations is obtained by writing Faraday's law for inner elementary electric loops. $\boldsymbol{G}_m$ is a diagonal matrix holding the magnetic conductances that pass through the electric loops. The block $\left[\boldsymbol{B}_1\ \boldsymbol{B}_2\right]$ has only 0, 1, -1 entries, describing the incidence of inner branches and branches on the boundary to electric faces. The second group corresponds to Ampere's law for elementary magnetic loops. $\boldsymbol{C}_i$ and $\boldsymbol{G}_i$ are diagonal matrices, holding the capacitances and electric conductances of the inner branches. The third group represents Faraday's law for electric loops on the boundary. $\boldsymbol{B}_{SI}$ has only 0, 1, -1 entries, describing the incidence of electric branches included in the boundary to the electric boundary faces. The forth row is obtained from the current conservation law for all nodes on the boundary excepting for the nodes on the electric terminals. $\boldsymbol{G}_{SI}$ and $\boldsymbol{C}_{SI}$ hold electric conductances and capacitances directly connected to boundary. The fifth set of equations represents current conservation for electric terminals. $\boldsymbol{G}_{TE}$ and $\boldsymbol{C}_{TE}$ hold electric conductances and capacitances that are directly connected to electric terminals. $\boldsymbol{S}_E$ is the connexion matrix between electric branches and terminals path. The last row is the discrete form of (5.27), obtained by expressing the voltages of electric terminals as sums of voltages along open paths from terminals to ground, $\boldsymbol{P}_E$ being a topological matrix that holds the paths that connect electric terminals to ground.

Thus, the top left square block of $\boldsymbol{C}$ is diagonal and the top left square bloc of $\boldsymbol{G}$ is symmetric. The size of this symmetric bloc corresponds to the useful magnetic branches and to the useful inner electric branches. Its size is dominant over the size of the matrix, therefore, solving or reduction strategies that take into consideration this particular structure are useful.

The discretized state-space system given by (5.35) describes the input output relationship in the frequency domain

$$\boldsymbol{i} = \boldsymbol{Y}\boldsymbol{u}, \tag{5.37}$$

similar to (5.28), but having as transfer (circuit) function:

$$\boldsymbol{Y} = \boldsymbol{L}\,(s\boldsymbol{C} + \boldsymbol{G})^{-1}\,\boldsymbol{B} \tag{5.38}$$

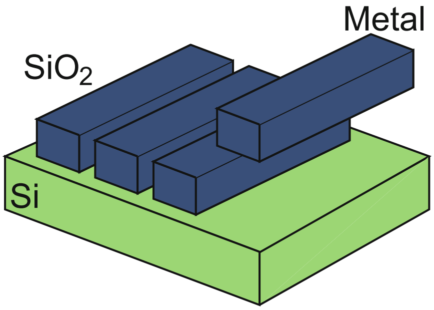
which is a rational function with a finite number of poles.

In conclusion, the discretization of the continuous model leads to a model represented by a MIMO linear time invariant system described by the state equations of finite size. Even if this is an important step ahead in the extraction procedure, the state space dimension is still too large for designer's needs, therefore a further modeling step aiming an order reduction is required.

5.2.3 Transmission Lines: 2D + 1D Models

In this section, aiming to reduce the model extraction effort, we will exploit the particular property of interconnects of having invariant transversal section along their extent. We assume that the field has a similar structure as a transversal electromagnetic wave that propagates along the line. The typical interconnect configuration (Fig. 5.15) considered consists of n parallel conductors having rectangular cross section, permeability $\mu = \mu_0$, permittivity $\varepsilon = \varepsilon_0$ and conductivity $\sigma_k, k = 1, 2, \cdots, n$, placed in a SiO$_2$ layer (σ_d, ε_d, possibly dependent on y) placed above a silicon substrate (σ_s, ε_s).

Fig. 5.15 Typical interconnect configuration



If the field is decomposed in its **longitudinal** (oriented along the line, which is assumed to lie along the Oz axis) and the **transversal components** (oriented in the xOy plane)

$$\mathbf{E} = \mathbf{E}_t + \mathbf{k}E_z, \quad \mathbf{J} = \mathbf{J}_t + \mathbf{k}J_z, \quad \mathbf{H} = \mathbf{H}_t + \mathbf{k}H_z, \tag{5.39}$$

then Maxwell's Equations can be separated into two groups:

$$\begin{array}{ll} \mathrm{curl}_{xy}\mathbf{H}_t = \mathbf{k}\left(J_z + \epsilon\frac{\partial E_z}{\partial t}\right), & \mathrm{div}_{xy}(\mu\mathbf{H_t}) = -\frac{\partial(\mu H_z)}{\partial z}, \\ \mathrm{curl}_{xy}\mathbf{E}_t = -\mathbf{k}\mu\frac{\partial H_z}{\partial t}, & \mathrm{div}_{xy}(\epsilon\mathbf{E_t}) = \rho - \frac{\partial(\epsilon E_z)}{\partial z}, \end{array} \tag{5.40}$$

called **transversal equations** and

$$\begin{array}{l} \frac{\partial \mathbf{E}_t}{\partial z} - \mathrm{grad}_{xy} E_t = -\mu\frac{\partial}{\partial t}(\mathbf{H_t} \times \mathbf{k}); \\ \frac{\partial \mathbf{H}_t}{\partial z} - \mathrm{grad}_{xy} H_z = \mathbf{J}_t \times \mathbf{k} + \epsilon\frac{\partial}{\partial t}(\mathbf{E_t} \times \mathbf{k}); \end{array}$$

called **propagation equations**.

The following hypotheses are adopted:

- The volume charge density ρ and the displacement current density $\frac{\partial \mathbf{E}}{\partial t}$ are neglected both in conductors and in the substrate.
- The following "longitudinal" terms $E_z = 0$, $H_z = 0$ are canceled in the transversal equations, neglecting the field generated by eddy currents.
- The longitudinal conduction current is neglected in dielectric $J_z = 0$, but not in the conductors.
- Since the conductances σ_k of the conductors are much bigger than the dielectric conductance σ_d, the transversal component of the electric field is neglected in the line conductors and in the substrate:

$$\mathbf{E}_t = \frac{1}{\sigma_k}\mathbf{J}_t = \mathbf{0}. \tag{5.41}$$

Under these hypotheses the transversal equations have the following form (where (k) = conductor, (s) = substrate, (d) = dielectric):

$$\begin{array}{ll} \mathrm{curl}_{xy}\mathbf{H}_t = \begin{cases} \mathbf{k}J_z, & \text{in (k) and (s)} \\ 0, & \text{in (d)} \end{cases} & \mathrm{div}_{xy}(\mu\mathbf{H}_t) = 0 \\ \mathrm{curl}_{xy}\mathbf{E}_t = 0, & \mathrm{div}_{xy}(\epsilon\mathbf{E}_t) = 0, \end{array} \tag{5.42}$$

identical with the **steady state electromagnetic field equations**. For this reason, the electric field admits a scalar electric potential $V(x, y, z, t)$, whereas the magnetic field admits a vector magnetic potential $\mathbf{A}(x, y, z, t) = \mathbf{k}A(x, y, z, t)$ with longitudinal orientation, so that:

$$\mathbf{E}_t = -\text{grad}_{xy} V, \tag{5.43}$$

$$\mathbf{H}_t = \frac{1}{\mu}[\mathbf{k} \times (\text{curl}\mathbf{A} \times \mathbf{k})] = -\mathbf{k} \times \frac{1}{\mu}\text{grad}_{xy}(A \times \mathbf{k}). \tag{5.44}$$

Thus, **the propagation equations** become:

$$\begin{aligned} &\text{grad}_{xy}\left[\tfrac{\partial A}{\partial t} + \tfrac{\partial V}{\partial z} + E_z\right] = 0, \\ &-\text{grad}_{xy} H_z = \mathbf{k} \times \left[\tfrac{1}{\mu}\text{grad}_{xy}\left(\tfrac{\partial A}{\partial z}\right) + \sigma \text{grad}_{xy} V + \epsilon \text{grad}_{xy}\left(\tfrac{\partial V}{\partial t}\right)\right]. \end{aligned} \tag{5.45}$$

By assuming an asymptotic behavior of potentials, the integration of the propagation equations yields to:

$$\begin{aligned} E_z &= \tfrac{1}{\sigma} J_z = -\tfrac{\partial V}{\partial z} - \tfrac{\partial A}{\partial t}, \\ H_z &= -\textstyle\int_C \left[\tfrac{1}{\mu}\tfrac{\partial}{\partial \mathbf{n}}\left(\tfrac{\partial A}{\partial z}\right) + \sigma \tfrac{\partial V}{\partial \mathbf{n}} + \epsilon \tfrac{\partial}{\partial \mathbf{n}}\left(\tfrac{\partial V}{\partial t}\right)\right] \mathrm{d}s, \end{aligned} \tag{5.46}$$

where C is a curve in the plane $z =$ constant, which starts from the infinity and stops in the computation point of the field H_z, $\mathbf{n}$ is the normal to the curve, oriented so that the line element is

$$\mathrm{d}\mathbf{s} = \mathrm{d}s\mathbf{k} \times \mathbf{n}. \tag{5.47}$$

From (5.41) it follows that the potential V is constant on every transversal cross-section of the conductors and zero in the substrate:

$$V|_{S_k} = V_k(z,t), \quad V_s = 0. \tag{5.48}$$

From relations (5.42) and (5.43) it follows that, in the transversal plane, the electric field has the same distribution as an electrostatic field. By using the uniqueness theorem of the electrostatic field it results that the function $V(x, y, z, t)$ is uniquely determined by the potentials of the conductors V_k. Consequently, due to the linearity, the per unit length (p.u.l.) charge of conductors and the current loss through the dielectric are:

$$q_k(z,t) = -\int_{\partial S_k} \epsilon_d \frac{\partial V}{\partial \mathbf{n}} \mathrm{d}s = \sum_{m=1}^{n} c_{km} V_m(z,t); \tag{5.49}$$

$$i_{gk}(z,t) = -\int_{\partial S_k} \sigma_d \frac{\partial V}{\partial \mathbf{n}} \mathrm{d}s = \sum_{m=1}^{n} g_{km} V_m(z,t), \tag{5.50}$$

where c_{km} is the p.u.l. capacitance, and g_{km} is the p.u.l. conductance between the conductor k and the conductor m.

By integrating E_z equation from (5.46) over the surface S_k and H_z equation from (5.46) along the path ∂S_k which bounds this surface, the following propagation equations for potentials are obtained:

$$-\frac{\partial \tilde{v}_k}{\partial z} = r_k^0 i_k + \frac{\partial \tilde{a}_k}{\partial t}; \qquad -\frac{\partial i_k}{\partial z} = i_{gk} + \frac{\partial q_k}{\partial t}, \tag{5.51}$$

where $r_k^0 = 1/(\sigma_k A_{S_k})$ is the p.u.l. d.c. resistance of the conductor k, and

$$\begin{aligned} \tilde{v}_k(z,t) &= \tfrac{1}{A_{S_k}} \textstyle\int_{S_k} V(x,y,z,t)\mathrm{d}x\mathrm{d}y = v_k(z,t), \\ \tilde{a}_k &= \tfrac{1}{A_{S_k}} \textstyle\int_{S_k} A(x,y,z,t)\mathrm{d}x\mathrm{d}y \end{aligned} \tag{5.52}$$

are the average values of the two potentials on the cross-section of the conductor k.

By computing the average values of the magnetic potential as in [58] and by substituting (5.49), (5.50) in (5.51) the following expressions are obtained in zero initial conditions:

$$-\frac{\partial v_k}{\partial z} = r_k^0 i_k + \sum_{m=1}^{n} l_{km}^0 \frac{\partial i_m}{\partial t} + \sum_{m=1}^{n} \frac{\partial}{\partial t} \int_0^t \left(\frac{\mathrm{d}l_{km}}{\mathrm{d}t} \right)_{t-\tau} i_m(\tau,t)\,\mathrm{d}\tau, \tag{5.53}$$

$$-\frac{\partial i_k}{\partial z} = \sum_{m=1}^{n} \left(g_{km} v_m + c_{km} \frac{\partial v_m}{\partial t} \right), \tag{5.54}$$

where l_{km}^0 are the p.u.l. external inductances (self inductances for $k = m$ and mutual inductances for $k \neq m$) of the conductors (k) and (m) where the return current is distributed on the surface of the substrate, and $l_{km}(t)$ are "transient p.u.l. inductances", defined as the average values on S_k of the vector potential A obtained in zero initial conditions by a unity step current injected in conductor (m).

For zero initial conditions for the currents $i_m(z,0) = 0$, for the potential $v_m(z,0) = 0$ and for the field $\boldsymbol{B}_k^0(s) = \boldsymbol{0}$, the Laplace transform of (5.53) and (5.54) can be written as:

$$-\frac{\mathrm{d}v_k(z,s)}{\mathrm{d}z} = \sum_{m=1}^{n} Z_{km}(s) i_m(z,s), \quad -\frac{\mathrm{d}i_k(z,s)}{\mathrm{d}z} = \sum_{m=1}^{n} Y_{km}(s) v_m(z,s), \tag{5.55}$$

which is identical to the operational form of the classical Transmission Lines (TLs) Telegrapher's equations, but where the p.u.l. inductances depend on s (implicitly on the frequency in a time-harmonic regime). In order to extract these dependencies, a magneto-quasi-static (MQS) field problem has to be solved.

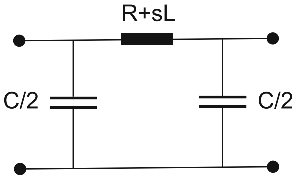


Fig. 5.16 The coarsest model for a single transmission line: a pi equivalent circuit

5.2.4 Numeric Extraction of Line Parameters

Models with various degrees of fineness can be established for TLs. The coarsest ones are circuit models with lumped parameters, such as the pi equivalent circuit for a single TL shown in Fig. 5.16. As expected, the characteristic of such a circuit is appropriate only at low frequencies, over a limited range, and for short lines. Even chaining similar cells, the result is not appropriate.

At high frequencies, the distributed effects have to be considered as an important component of the model. Proper values for the line parameters can be obtained only by simulating the electromagnetic field. The extraction of line parameters is the main step in TLs modeling since the behavior of a line with a given length can be computed from them. For instance, for a multiconductor transmission line, from the per unit length parameters matrices **R**, **L**, **C** and **G** the transfer matrix for a line of length l can be computed as

$$\mathbf{T} = \exp[(\mathbf{D} + j\omega\mathbf{E})l], \quad \text{where} \quad \mathbf{D} = \begin{bmatrix} \mathbf{0} & -\mathbf{R} \\ -\mathbf{G} & \mathbf{0} \end{bmatrix}, \quad \mathbf{E} = \begin{bmatrix} \mathbf{0} & -\mathbf{L} \\ -\mathbf{C} & \mathbf{0} \end{bmatrix}. \tag{5.56}$$

From them, other parameters (impedance, admittance or scattering) can be computed. The simplest method to extract constant matrices of the line resistance **R**, capacitance **C** and inductance **L**, respectively, is to solve the field equations numerically in steady-state electric conduction (EC), electrostatics (ES) and magnetostatics (MS) regimes. Empirical formulas may also be found in the literature, such as the ones given in [62] for the line capacitance. None of them take the frequency dependence of p.u.l. parameters into account.

A first attempt to take into consideration the frequency effect, which becomes important at high frequencies, is to compute the skin depth in the conductor and to use a better approximation for the resistance. In [52] we proposed a much more accurate estimation of frequency dependent line parameters based on the numerical modeling of the EM field including the semiconductor substrate. The previous section is the theoretical argument of this approach in which two complementary problems are solved, the first one describing the transversal behavior of the line from which $\mathbf{Y}_l(\omega) = \mathbf{G}(\omega) + \mathrm{j}\omega\mathbf{C}(\omega)$ is consequently extracted, and the second one describing the longitudinal behavior of the line from which $\mathbf{Z}_l(\omega) = \mathbf{R}(\omega) + \mathrm{j}\omega\mathbf{L}(\omega)$ is extracted.

Since the first field problem is dedicated to the computation of the transversal capacitances between wires and their loss conductances, according to the previous section, the natural choice is to solve a 2D problem of the transversal electro-quasi-static (EQS) field in dielectrics, considering the line wires as perfect conductors with given voltage. The boundary conditions are of Dirichlet type $V = 0$ on the lower electrode, and open boundary conditions (e.g. Robin, SDI or appropriate ELOB [50]) on the other three sides. A dual approach, such as dFIT [51] allows a robust and accurate parameter extraction.

The second field problem focuses on the longitudinal electric and the generated transversal magnetic field. Consequently, a short line-segment (with only one cell layer) is considered. The magneto-quasi-static (MQS) regime of the EM field is appropriate for the extraction of $\mathbf{Z}_l(\omega)$. However, for our simulations we used a our FIT solver for Full Wave (FW) ECE problems. The magnetic grid is 2D, thus ensuring the TM mode of propagation.

In order to eliminate the transversal distribution of the electric field, the lower electrode is prolongated over the entire far-end cross-section of the rectangular computational domain, which thus has perfect electric conductor (PEC) boundary conditions $\mathbf{E}_t = \mathbf{0}$ on two of their faces. On the three lateral faces, open-absorbing boundary conditions are the natural choice, whereas on the near-end cross-section the natural boundary conditions are those of the Electric Circuit Element (ECE): $B_n = 0$, $\mathbf{n} \times \text{curl}\mathbf{H} = \mathbf{0}$ excepting for the wire traces, where $\mathbf{E}_t = \mathbf{0}$. These conditions ensure the correct definition of the terminals voltages, and consequently of the impedance/admittance matrix (Fig. 5.17).

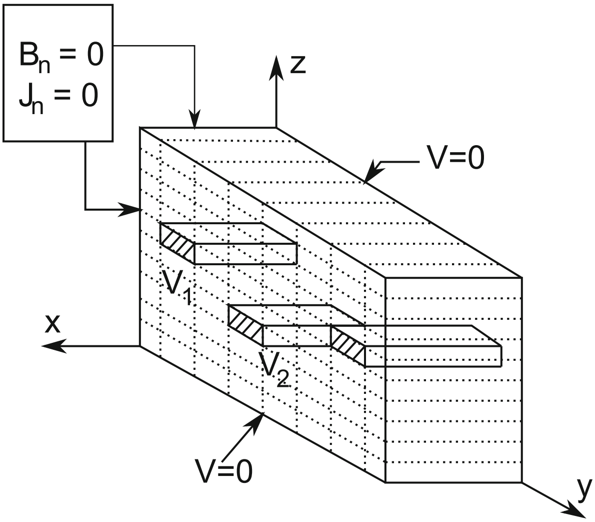


Fig. 5.17 Boundary conditions for the full wave – transversal magnetic problem

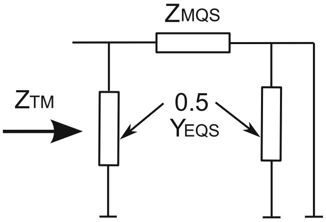


Fig. 5.18 The pi equivalent circuit for a simulated short line segment. Parameters are evaluated from field simulations

These boundary conditions are the field representation of the line segment with short-circuit at the far-end, whereas the 2D EQS problem is the field representation of the segment-line with open far-end.

The transversal component is finally subtracted from the FW-TM simulation to obtain an accurate approximation of the line impedance, as given by

$$\mathbf{Z}_{MQS} = \left(\mathbf{Z}_{TM}^{-1} - \frac{1}{2}\mathbf{Y}_{EQS}\right)^{-1}. \tag{5.57}$$

This subtraction is carried out according to a pi-like equivalent net for the simulated short segment (Fig. 5.18). Finally, the line parameters are:

$$\mathbf{G}(\omega) = \mathrm{Re}(\mathbf{Y}_l), \quad \mathbf{C}(\omega) = \mathrm{Im}(\mathbf{Y}_l)/\omega, \quad \mathbf{R}(\omega) = \mathrm{Re}(\mathbf{Z}_l), \quad \mathbf{L}(\omega) = \mathrm{Im}(\mathbf{Z}_l)/\omega, \tag{5.58}$$

where

$$\mathbf{Y}_l = \mathbf{Y}_{EQS}/\Delta l, \quad \mathbf{Z}_l = \mathbf{Z}_{MQS}/\Delta l, \tag{5.59}$$

where Δl is the length of the considered line-segment and $\mathbf{Z}_{TM}$ is the impedance matrix extracted from the TM field solution.

This numerical approach to extract the line parameters, named the *two fields method*, is more robust and may be applied without difficulties to multi-wire lines. The obtained values of the line parameters are frequency dependent, taking into consideration proximity and skin effects as well as losses induced in the conducting substrate.

5.2.5 Variability Analysis of Line Parameters

The simplest way to analyze the parameter variability is to compute first order sensitivities. These are derivatives of the device characteristics with respect to the design parameters. The sensitivities of the line parameters are essential to estimate the impact of small variations on the device behavior. Moreover, the sensitivity of the terminal behavior of interconnects can also be estimated.

For instance, in the case of a single TL, having the global admittance given by

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} \frac{\cosh\gamma l}{Z_c \sinh\gamma l} & -\frac{1}{Z_c \sinh\gamma l} \\ -\frac{1}{Z_c \sinh\gamma l} & \frac{\cosh\gamma l}{Z_c \sinh\gamma l} \end{bmatrix} \tag{5.60}$$

the sensitivities of the terminal admittance with respect to a parameter can be computed as:

$$\frac{\partial Y_{11}}{\partial \alpha} = \frac{l}{Z_c}\frac{\partial \gamma}{\partial \alpha} - \frac{\cosh\gamma l}{Z_c^2 \sinh\gamma l}\frac{\partial Z_c}{\partial \alpha} - \frac{l}{Z_c}\frac{\cosh^2\gamma l}{\sinh^2\gamma l}\frac{\partial \gamma}{\partial \alpha} \tag{5.61}$$

$$\frac{\partial Y_{12}}{\partial \alpha} = \frac{1}{Z_c^2 \sinh\gamma l}\frac{\partial Z_c}{\partial \alpha} + \frac{l}{Z_c}\frac{\cosh}{\sinh^2\gamma l}\frac{\partial \gamma}{\partial \alpha} \tag{5.62}$$

where the sensitivities of

$$\gamma = \sqrt{(R + j\omega L)(G + j\omega C)} \quad \text{and} \quad Z_c = \sqrt{(R + j\omega L)/(G + j\omega C)}$$

can be computed if the sensitivities of the p.u.l. parameters $\partial R/\partial \alpha$, etc. are known.

In the case of a multiconductor TL with n conductors the sensitivity of the admittance matrix $\mathbf{Y}$ of dimension $(2n \times 2n)$ is computed by means of the sensitivity of the transfer matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} \\ \mathbf{T}_{21} & \mathbf{T}_{22}, \end{bmatrix} \tag{5.63}$$

also of dimension $(2n \times 2n)$, knowing that

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \end{bmatrix} = \begin{bmatrix} -\mathbf{T}_{12}^{-1}\mathbf{T}_{11} & \mathbf{T}_{12}^{-1} \\ \mathbf{T}_{22}\mathbf{T}_{12}^{-1}\mathbf{T}_{11} - \mathbf{T}_{21} & -\mathbf{T}_{22}\mathbf{T}_{12}^{-1} \end{bmatrix}. \tag{5.64}$$

In the formulas above, all the sub-blocks are of dimensions $(n \times n)$. For instance

$$\frac{\partial \mathbf{Y}_{11}}{\partial \alpha} = -\mathbf{T}_{12}^{-1}\frac{\partial \mathbf{T}_{12}}{\partial \alpha}\mathbf{T}_{12}^{-1}\mathbf{T}_{11} - \mathbf{T}_{12}^{-1}\frac{\partial \mathbf{T}_{11}}{\partial \alpha}, \tag{5.65}$$

$$\frac{\partial \mathbf{Y}_{12}}{\partial \alpha} = -\mathbf{T}_{12}^{-1}\frac{\partial \mathbf{T}_{12}}{\partial \alpha}\mathbf{T}_{12}^{-1}. \tag{5.66}$$

The transfer matrix $\mathbf{T}$ is computed with (5.56) and its sensitivity is

$$\frac{\partial \mathbf{T}}{\partial \alpha} = \exp[(\mathbf{D} + j\omega\mathbf{E})l]\left(\frac{\partial \mathbf{D}}{\partial \alpha} + j\omega\frac{\partial \mathbf{E}}{\partial \alpha}\right), \tag{5.67}$$

where

$$\frac{\partial \mathbf{D}}{\partial \alpha} = \begin{bmatrix} \mathbf{0} & -\partial \mathbf{R}/\partial \alpha \\ -\partial \mathbf{G}/\partial \alpha & \mathbf{0} \end{bmatrix}, \quad \frac{\partial \mathbf{E}}{\partial \alpha} = \begin{bmatrix} \mathbf{0} & -\partial \mathbf{L}/\partial \alpha \\ -\partial \mathbf{C}/\partial \alpha & \mathbf{0} \end{bmatrix}. \tag{5.68}$$

Thus, the basic quantities needed to estimate the sensitivity of the admittance are the sensitivities of the p.u.l. parameters. By using a direct differentiation technique, as explained in [41] the sensitivities of the EQS and TM problems with respect to the parameters that vary, i.e. $\partial \mathbf{Y}_{EQS}/\partial \alpha$ and $\partial \mathbf{Z}_{TM}/\partial \alpha$ are computed. Then, the sensitivity of the MQS mode is computed by taking the derivative of (5.57):

$$\frac{\partial \mathbf{Z}_{MQS}}{\partial \alpha} = -\left(\mathbf{Z}_{TM}^{-1} - \frac{1}{2}\mathbf{Y}_{EQS}\right)^{-1}\left(-\mathbf{Z}_{TM}^{-1}\frac{\partial \mathbf{Z}_{TM}}{\partial \alpha}\mathbf{Z}_{TM}^{-1} - \frac{1}{2}\frac{\partial \mathbf{Y}_{EQS}}{\partial \alpha}\right)\left(\mathbf{Z}_{TM}^{-1} - \frac{1}{2}\mathbf{Y}_{EQS}\right)^{-1} \tag{5.69}$$

Finally, the sensitivities of the p.u.l. parameters are:

$$\frac{\partial \mathbf{R}}{\partial \alpha} = \frac{1}{l}\mathrm{Re}\left\{\frac{\partial \mathbf{Z}_{MQS}}{\partial \alpha}\right\}, \quad \frac{\partial \mathbf{L}}{\partial \alpha} = \frac{1}{l\omega}\mathrm{Im}\left\{\frac{\partial \mathbf{Z}_{MQS}}{\partial \alpha}\right\}, \tag{5.70}$$

$$\frac{\partial \mathbf{G}}{\partial \alpha} = \frac{1}{l}\mathrm{Re}\left\{\frac{\partial \mathbf{Y}_{EQS}}{\partial \alpha}\right\}, \quad \frac{\partial \mathbf{C}}{\partial \alpha} = \frac{1}{l\omega}\mathrm{Im}\left\{\frac{\partial \mathbf{Y}_{EQS}}{\partial \alpha}\right\}. \tag{5.71}$$

The values of the sensitivities thus obtained depend on the frequency as well.

5.2.6 Parametric Models Based on Taylor Series

Continuous improvements in today's fabrication processes determine smaller chip sizes and smaller device geometries. Process variations induce changes in the properties of metallic interconnect between devices.

Simple parametric models are often obtained by truncating the Taylor series expansion for the quantity of interest. This requires the computation of the derivatives of the device characteristics with respect to the design parameters [55]. Let us assume that $y(\alpha_1, \alpha_2, \cdots, \alpha_n) = y(\boldsymbol{\alpha})$ is the device characteristic which depends on the design parameters $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_n]$. The quantity y may be, for instance, the real or the imaginary part of the device admittance at a given frequency or any of the p.u.l. parameters. The parameter variability is thus completely described by the real function, y, defined over the design space S, a subset of $\mathbb{R}^n$. The nominal design parameters correspond to the particular choice $\boldsymbol{\alpha}_0 = [\alpha_{01}\ \alpha_{02}\ \cdots\ \alpha_{0n}]$.

5.2.6.1 Additive Model (A)

If y is smooth enough then its truncated Taylor Series expansion is the best polynomial approximation in the vicinity of the expansion point $\boldsymbol{\alpha}_0$. For one parameter ($n = 1$), the additive model is the first order truncation of the Taylor series:

$$\hat{y}(\alpha) = y(\alpha_0) + \frac{\partial y}{\partial \alpha}(\alpha_0)(\alpha - \alpha_0). \tag{5.72}$$

If we denote by $y(\alpha_0) = y_0$ the nominal value of the output function, by $\frac{\partial y}{\partial \alpha}(y_0)\frac{\alpha_0}{y_0} = S_\alpha^y$ the relative first order sensitivity and by $(\alpha - \alpha_0)/\alpha_0 = \delta\alpha$ the relative variation of the parameter α, then the variability model based on (5.72) defines an *affine* [60] or *additive* model (A):

$$\hat{y}(\alpha) = y_0(1 + S_\alpha^y \delta\alpha). \tag{5.73}$$

To ensure a relative validity range of the first order approximation of the output quantity less a given threshold t_1, the absolute variation of the parameter must be less than

$$V_d = \sqrt{\frac{2y_0t_1}{D_2}}, \tag{5.74}$$

where D_2 is an upper limit of the second order derivative of the output quantity y with respect to parameter α [41].

For the multiparametric case, one gets:

$$y(\boldsymbol{\alpha}) = y(\boldsymbol{\alpha}_0) + \nabla y(\boldsymbol{\alpha}_0) \cdot (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) = y_0 + \sum_{k=1}^{n} \frac{\partial y}{\partial \alpha_k}(\boldsymbol{\alpha}_0)(\alpha_k - \alpha_{0k}). \tag{5.75}$$

Similar with one parameter case, the relative sensitivities w.r.t. each parameter are denoted by $\frac{\partial y}{\partial \alpha_k}(\boldsymbol{\alpha}_0)\frac{\alpha_{0k}}{y_0} = S_{\alpha_k}^y$ and the relative variations of the parameters by $\delta\alpha_k = (\alpha_k - \alpha_{0k})/\alpha_{0k}$, the additive model (A) for n parameters being given by:

$$\hat{y}(\boldsymbol{\alpha}) = y_0\left(1 + \sum_{k=1}^{n} S_{\alpha_k}^y \delta\alpha_k\right). \tag{5.76}$$

Thus, each new independent parameter taken into account adds a new term to the sum [52]. The additive model is simply a normalized standard version of a linearly truncated Taylor expansion.

Instead of using this truncated expansion may be numerically favorable to expand some transformation $F(y)$ of y instead. Two particular choices for F have practical importance: identity and inversion as it will be indicated below.

5.2.6.2 Rational Model (R)

The rational model is the additive model for the reverse quantity $1/y$. It is obtained from the first order truncation of the Taylor Series expansion for the function $1/y$. For $n = 1$, if we denote by $r(\alpha) = \frac{1}{y(\alpha)}$, it follows that:

$$\hat{r}(\alpha) = r(\alpha_0) + \frac{\partial r}{\partial \alpha}(\alpha_0)(\alpha - \alpha_0). \tag{5.77}$$

We define the relative first order sensitivity of the reverse circuit function:

$$\frac{\partial r}{\partial \alpha}(\alpha_0)\frac{\alpha_0}{r(\alpha_0)} = S_\alpha^r = S_\alpha^{1/y}. \tag{5.78}$$

Consequently, we obtain the rational model for $n = 1$:

$$y(\alpha) = \frac{y_0}{1 + S_\alpha^{1/y}\delta\alpha}. \tag{5.79}$$

It can be easily shown that the reverse relative sensitivity is $S_\alpha^{\frac{1}{y}} = -S_\alpha^y$. For the multiple parameter case, the rational model is:

$$\hat{y}(\boldsymbol{\alpha}) = \frac{y_0}{1 + \sum_{k=1}^{n} S_{\alpha_k}^{1/y}\delta\alpha_k}. \tag{5.80}$$

If the circuit function y is for instance the admittance, its inverse $1/y$ is the impedance. In the time domain, these two transfer functions correspond to a device excited in voltage or in current, respectively. Consequently, the choice between additive and rational models for the variability analysis of the circuit functions in frequency domain can be interpreted as a change in the terminal excitation mode in the time domain state representation. Choosing the appropriate terminal excitation, the validity range of the parametric model based on first order Taylor series approximation can be dramatically extended.

5.2.7 Parametric Circuit Synthesis

We have shown in [48] that one of the most efficient order reduction method for the class of problems we address is the Vector Fitting (VFIT) method proposed in [47], improved in [43, 46] and available at [61]. It finds the transfer function matching a given frequency characteristic. Thus, in the frequency domain, for the output quantity $\boldsymbol{y}(s)$, this procedure finds the poles p_m (real or complex conjugate pairs), the residuals $\boldsymbol{k}_m$ and the constant terms $\boldsymbol{k}_0$ and $\boldsymbol{k}_\infty$ of a rational approximation of the output quantity (e.g an admittance):

$$\boldsymbol{y}(s) \approx \boldsymbol{y}_{VFIT}(s) = \boldsymbol{k}_\infty + s\boldsymbol{k}_0 + \sum_{m=1}^{q} \frac{\boldsymbol{k}_m}{s - p_m}. \tag{5.81}$$

The resulting approximation has guaranteed stable poles and the passivity can be enforced in a post-processing step [43]. The transfer function (5.81) can be synthesized by using the Differential Equation Macromodel (DEM) [57]. Our aim is to extend DEM to take into consideration the parameterization.

To simplify the explanations, we assume a single input single output system, excited in voltage. It follows that the output current is given by (5.82), where $x_m(s)$ is a new variable defined by (5.83).

$$i(s) = y(s)u(s) = k_\infty u(s) + sk_0 u(s) + \sum_{m=1}^{q} k_m x_m(s), \tag{5.82}$$

$$x_m(s) = \frac{u(s)}{s - p_m}. \tag{5.83}$$

By applying the inverse Laplace transformation to (5.82) and (5.83), relationships (5.84) and (5.85) are obtained:

$$i(t) = k_\infty u(t) + k_0 \frac{\mathrm{d}\,u(t)}{\mathrm{d}\,t} + \sum_{m=1}^{q} k_m x_m(t), \tag{5.84}$$

$$\frac{\mathrm{d}\,x_m(t)}{\mathrm{d}\,t} = p_m x_m(t) + u(t). \tag{5.85}$$

If we use the following matrix notations

$$\boldsymbol{A} = \mathrm{diag}(p_1, p_2, \ldots, p_q), \qquad \boldsymbol{b} = \left[1\ 1\ \cdots\ 1\right]^T, \tag{5.86}$$

$$\boldsymbol{c} = \left[k_1\ k_2\ \cdots\ k_q\right]^T \qquad \boldsymbol{x} = \left[x_1\ x_2\ \cdots\ x_q\right]^T, \tag{5.87}$$

then equations of the system (5.84), (5.85) can be written in a compact form as

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t), \tag{5.88}$$

$$i(t) = k_\infty u(t) + k_0 \frac{\mathrm{d}\,u(t)}{\mathrm{d}\,t} + \boldsymbol{c}\boldsymbol{x}(t). \tag{5.89}$$

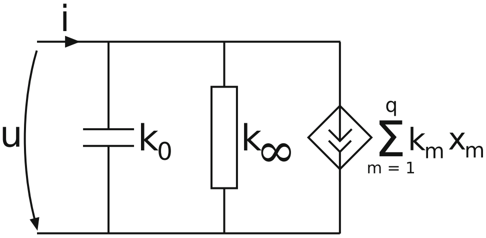


Fig. 5.19 Equivalent circuit for the output equation if all poles are real

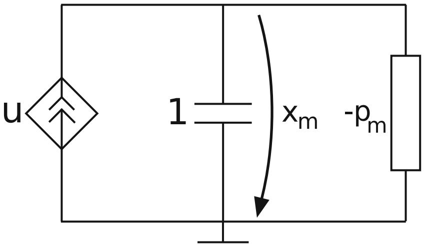


Fig. 5.20 Sub-circuit corresponding to a real pole

5.2.7.1 Case of Real Poles

In the case in which all poles (and consequently, all the residuals) are real, Eq. (5.84) can be synthesized by the circuit shown in Fig. 5.19 which consists of a capacitor having the capacitance k_0, in parallel with a resistor having the conductance k_∞, in parallel with q voltage controlled current sources, their parameters being the residuals k_m.

Equation (5.85) can be synthesized by the circuit in Fig. 5.20, where x_m is the voltage across a unity capacitor, connected in parallel with a resistor having the conductance $-p_m$ and a voltage controlled current source, controlled by the input voltage u.

We would like to include the parametric dependence into the VFIT model and in the synthesized circuit. To keep the explanations simple, we assume that there is only one parameter that varies, i.e. the quantity α is a scalar. Assuming that keeping the order q is satisfactory for the whole range of the variation of this parameter, this means that (5.81) can be parameterized as:

$$\boldsymbol{y}(s,\alpha) \approx \boldsymbol{y}_{VFIT}(s,\alpha) = \boldsymbol{k}_\infty(\alpha) + s\boldsymbol{k}_0(\alpha) + \sum_{m=1}^{q} \frac{\boldsymbol{k}_m(\alpha)}{s - p_m(\alpha)}. \tag{5.90}$$

Without loss of generality, we can assume that the additive model is more accurate than the rational one. If not, the reverse quantity is used, which is equivalent, for our class of problems, to change the excitation of terminals from voltage excited to current excited, and use an additive model for the impedance $\boldsymbol{z} = \boldsymbol{y}^{-1}$. The additive model (5.73) can be written as

$$\boldsymbol{y}(s,\alpha) \approx \boldsymbol{y}_A(s,\alpha) = \boldsymbol{y}(s,\alpha_0) + \frac{\partial \boldsymbol{y}}{\partial \alpha}(s,\alpha_0)(\alpha - \alpha_0), \tag{5.91}$$

where here $\boldsymbol{y}$ is a matrix function. By combining (5.90) and (5.91) we obtain an approximate additive model based on VFIT:

$$\boldsymbol{y}(s,\alpha) \approx \boldsymbol{y}_{A-VFIT}(s,\alpha) = \boldsymbol{y}_{VFIT}(s,\alpha_0) + \frac{\partial \boldsymbol{y}_{VFIT}}{\partial \alpha}(s,\alpha_0)(\alpha - \alpha_0). \tag{5.92}$$

From (5.90) it follows that the sensitivity of the VFIT approximation needed in (5.92) is

$$\frac{\partial \boldsymbol{y}_{VFIT}}{\partial \alpha} = \frac{\partial \boldsymbol{k}_\infty}{\partial \alpha} + s\frac{\partial \boldsymbol{k}_0}{\partial \alpha} + \sum_{m=1}^{q}\left[\frac{\partial \boldsymbol{k}_m/\partial \alpha}{s - p_m} + \frac{\boldsymbol{k}_m}{(s-p_m)^2}\frac{\partial p_m}{\partial \alpha}\right]. \tag{5.93}$$

The sensitivity $\partial \boldsymbol{y}/\partial \alpha$ can be evaluated with (5.61) for as many frequencies as required and thus the sensitivities of poles and residues in (5.93) can be computed by solving the linear system (5.93) by least square approximation. Finally, by substituting (5.93) and (5.90) in (5.92), the final parameterized and frequency dependent model is obtained:

$$\begin{aligned}\boldsymbol{y}_{A-VFIT}(s,\alpha) &= \left[\boldsymbol{k}_\infty + (\alpha-\alpha_0)\frac{\partial \boldsymbol{k}_\infty}{\partial \alpha}\right] + s\left[\boldsymbol{k}_0 + (\alpha-\alpha_0)\frac{\partial \boldsymbol{k}_0}{\partial \alpha}\right] + \\ &+ \sum_{m=1}^{q}\left[\frac{\boldsymbol{k}_m + (\alpha-\alpha_0)\partial \boldsymbol{k}_m/\partial \alpha}{s-p_m}\right] + (\alpha-\alpha_0)\sum_{m=1}^{q}\left[\frac{\boldsymbol{k}_m}{(s-p_m)^2}\frac{\partial p_m}{\partial \alpha}\right].\end{aligned} \tag{5.94}$$

Expression (5.94) has the advantage that it has an explicit dependence with respect both to the frequency $s = j\omega$ and parameter α, is easy to implement and feasible to be synthesized as a net-list having components with dependent parameters, as explained below.

If we denote by

$$k_*(\alpha) = k_* + (\alpha - \alpha_0)\frac{\partial k_*}{\partial \alpha}, \tag{5.95}$$

where $k_* = k_*(\alpha_0)$ then Eq. (5.94) can be written as

$$y_{A-VFIT}(s,\alpha) = y_1(s,\alpha) + y_2(s,\alpha), \tag{5.96}$$

where

$$y_1(s,\alpha) = k_\infty(\alpha) + sk_0(\alpha) + \sum_{m=1}^{q} \frac{k_m(\alpha)}{s - p_m}, \tag{5.97}$$

$$y_2(s,\alpha) = (\alpha - \alpha_0) \sum_{m=1}^{q} \frac{k_m}{(s-p_m)^2} \frac{\partial p_m}{\partial \alpha}. \tag{5.98}$$

The output current is thus

$$i(s,\alpha) = y_1(s,\alpha)u(s) + y_2(s,\alpha)u(s), \tag{5.99}$$

where the first term can be synthesized with a circuit similar to the one in Fig. 5.19 but where the k_* parameters depend on α, and the second term

$$i_2(s,\alpha) = (\alpha - \alpha_0) \sum_{m=1}^{q} \frac{k_m}{(s-p_m)^2} \frac{\partial p_m}{\partial \alpha} u(s) \tag{5.100}$$

adds q new parallel branches to the circuit (Fig. 5.21). It is useful to write (5.100) as

$$i_2(s,\alpha) = \sum_{m=1}^{q} E_m(\alpha) \frac{u(s)}{(\frac{s}{p_m} - 1)^2}, \quad \text{where} \quad E_m(\alpha) = \frac{(\alpha - \alpha_0)k_m}{p_m^2} \frac{\partial p_m}{\partial \alpha}. \tag{5.101}$$

The part that depends on s in (5.101) can be synthesized by a second order circuit, such as the one in Fig. 5.22.

The current through the coil is

$$j(s) = \frac{u(s)}{s^2LC + sLG + 1}. \tag{5.102}$$

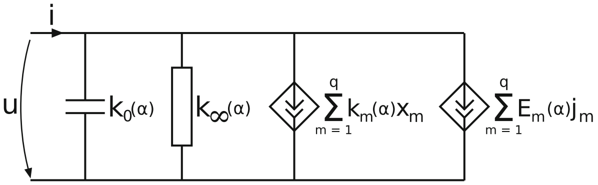

i
u
$k_0(\alpha)$
$k_\infty(\alpha)$
$\sum_{m=1}^{q} k_m(\alpha)x_m$
$\sum_{m=1}^{q} E_m(\alpha)j_m$

Fig. 5.21 Parameterized circuit corresponding to the output equation

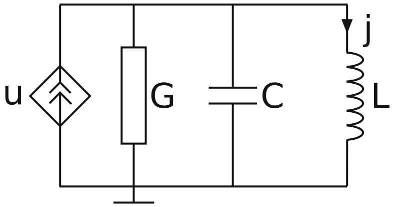


Fig. 5.22 Second order subcircuit, with a voltage controlled current source

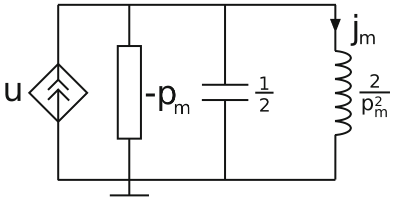


Fig. 5.23 Second order subcircuit corresponding to a real pole

To obtain the expression in (5.101) it is necessary that $LC = 1/p_m^2$, $LG = -2/p_m$, for instance, we can chose $G = -p_m$, $L = 2/p_m^2$, $C = 1/2$. Thus, the parameterized circuit is given by the sub-circuits in Figs. 5.21, 5.20 and 5.23. The circuit that corresponds to the output equations has new branches with current controlled current sources. Only this sub-circuit contained parameterized components.

Another possibility to derive a parameterized circuit is to do as follows. In (5.100) we denote by

$$\frac{1}{(s-p_m)^2}\frac{\partial p_m}{\partial \alpha}u(s) = f_m(s), \tag{5.103}$$

and by

$$(s-p_m)f_m(s) = g_m(s). \tag{5.104}$$

Relationships (5.103) and (5.104) are equivalent to

$$sg_m(s) = p_m g_m(s) + \frac{\partial p_m}{\partial \alpha}u(s), \tag{5.105}$$

$$sf_m(s) = p_m f_m(s) + g_m(s), \tag{5.106}$$

which correspond in the time domain to

$$\frac{\mathrm{d}g_m(t)}{\mathrm{d}t} = p_m g_m(t) + \frac{\partial p_m}{\partial \alpha}u(t), \tag{5.107}$$

$$\frac{\mathrm{d}f_m(t)}{\mathrm{d}t} = p_m f_m(t) + g_m(t). \tag{5.108}$$

Equations (5.107) and (5.108) can be synthesized with the subcircuit shown in Fig. 5.24. In this case the circuit that corresponds to the output equation is the one in Fig. 5.25. In brief, the parameterized reduced order circuit can be either the one in Figs. 5.21, 5.20 and 5.23 or in Figs. 5.25, 5.20 and 5.24. In both approaches only the circuit that corresponds to the output equation is parameterized. The second approach has the advantage that can be generalized for a transfer function having complex poles as well.

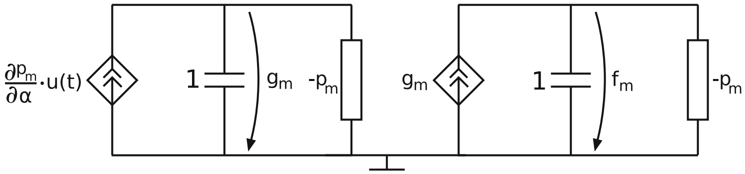


Fig. 5.24 Subcircuit corresponding to the second order term (second approach)

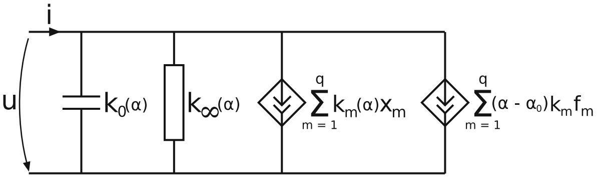


Fig. 5.25 Parameterized circuit corresponding to the output equation (second approach)

5.2.7.2 Case of Complex Poles

Nominal Differential Equation Macromodel

If some of the q poles are complex, then they appear in conjugated pairs since they are the roots of the characteristic equation corresponding to a real matrix. We assume for the beginning that the transfer function has only one pair of complex conjugate poles: $p = a + jb$ and $p^* = a - jb$. In this case the transfer function is

$$y(s) = \frac{k_1}{s-p} + \frac{k_2}{s-p^*} = \frac{(s-a)(k_1+k_2) + jb(k_1-k_2)}{(s-a)^2+b^2}. \tag{5.109}$$

The numerator can be a real polynomial in s only if k_1 and k_2 are complex conjugated residues: $k_1 = c + jd$, $k_2 = c - jd$. In this case, the matrices in (5.86) are

$$\boldsymbol{A} = \begin{bmatrix} a+jb & 0 \\ 0 & a-jb \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 1 \; 1 \end{bmatrix}^T \quad \boldsymbol{c} = \begin{bmatrix} c+jd \; c-jd \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} x_1 \; x_2 \end{bmatrix}^T. \tag{5.110}$$

In order to obtain a real coefficient equation, a matrix transformation is introduced. The system (5.88) becomes

$$\boldsymbol{V}\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^{-1}\boldsymbol{V}\boldsymbol{x}(t) + \boldsymbol{V}\boldsymbol{b}u(t), \tag{5.111}$$

$$i(t) = k_\infty u(t) + k_0\frac{\mathrm{d}\,u(t)}{\mathrm{d}\,t} + \boldsymbol{c}\boldsymbol{V}^{-1}\boldsymbol{V}\boldsymbol{x}(t), \tag{5.112}$$

where

$$\boldsymbol{V} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{j}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \end{bmatrix}, \quad \boldsymbol{V}^{-1} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{j}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{j}{\sqrt{2}} \end{bmatrix}. \tag{5.113}$$

Let

$$\hat{\boldsymbol{x}} = \boldsymbol{V}\boldsymbol{x} = \begin{bmatrix} \hat{x}_1 \ \hat{x}_2 \end{bmatrix}^T, \quad \hat{\boldsymbol{A}} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^{-1} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}, \tag{5.114}$$

$$\hat{\boldsymbol{b}} = \boldsymbol{V}\boldsymbol{b} = \begin{bmatrix} -\sqrt{2} \ 0 \end{bmatrix}, \quad \hat{\boldsymbol{c}} = \boldsymbol{c}\boldsymbol{V}^{-1} = \begin{bmatrix} -\sqrt{2}c \ \sqrt{2}d \end{bmatrix}. \tag{5.115}$$

The transformation $\hat{\boldsymbol{A}} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^{-1}$ is a *similarity transformation*, preserving the eigenvalues of the matrix and thus the characteristic polynomial of the system.

The two equations corresponding to the complex conjugated pair of poles

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p & 0 \\ 0 & p^* \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t) \tag{5.116}$$

become after applying the similarity transformation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}\begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} + \begin{bmatrix} -\sqrt{2} \\ 0 \end{bmatrix} u(t). \tag{5.117}$$

If there are several pairs of complex conjugated poles, Eq. (5.117) will be true for any of these pairs and, by renaming $p \to p_m$, $\hat{x}_1 \to \hat{x}'_m$, $\hat{x}_2 \to \hat{x}''_m$, $a \to a_m$, $b \to b_m$, the synthesized circuit is shown in Fig. 5.26.

In general, if the system has q poles out of which q_r are real and $q_c = (q - q_r)/2$ are pairs of complex conjugate poles, then the synthesis will be done as follows: for each real pole $m = 1, \ldots, q_r$, let k_m be the residue corresponding to the pole; for each pair of complex conjugate poles $m = 1, \ldots, q_c$ let the pole be $p'_m = a_m + jb_m$, with the corresponding residue $k'_m = c_m + jd_m$. An equivalent circuit for the output equation is shown in Fig. 5.27. It consists of the following elements connected in parallel:

- A capacitance k_0;
- A conductance k_∞,

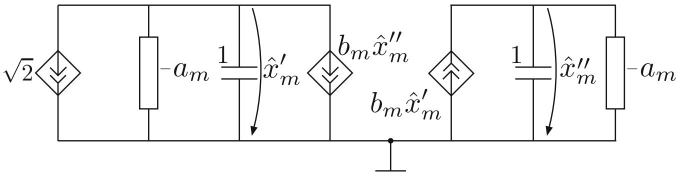


Fig. 5.26 Sub-circuit corresponding to a pair of complex conjugate poles

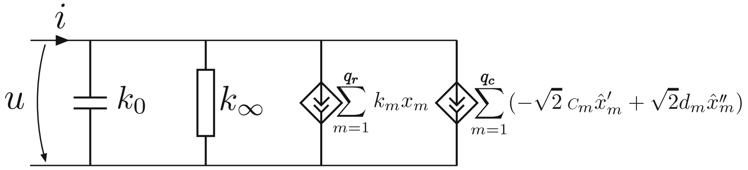

i
u
k_0
k_∞
$\sum_{m=1}^{q_r} k_m x_m$
$\sum_{m=1}^{q_c} (-\sqrt{2}\, c_m \hat{x}'_m + \sqrt{2} d_m \hat{x}''_m)$

Fig. 5.27 Sub-circuit corresponding to a pair of complex conjugate poles

- q_r voltage controlled current sources (having the parameter k_m, controlled by the voltages x_m),
- q_c voltage controlled current sources (having the parameter $-\sqrt{2}c_m$, controlled by the voltages $\hat{x}'_m$)
- q_c voltage controlled current sources (having the parameter $\sqrt{2}d_m$, controlled by the voltages $\hat{x}''_m$).

The voltages x_m are defined on the q_r subcircuits that correspond to real poles (Fig. 5.20) and the voltages $\hat{x}'_m$, $\hat{x}''_m$ are defined on the q_c subcircuits that correspond to the pair of complex conjugate poles (Fig. 5.26).

Parametric DEM

To derive the parametric circuit in the case of complex poles, we could proceed as we did in the first approach for real poles. This would conduce to a transfer function of order 4, which is not obvious how it can be synthesized. The second approach can be extended to the case of complex poles, as follows.

Let's consider Eqs. (5.107) and (5.108) written for a pair of complex conjugate poles $p_1 = a + jb$, $p_2 = a - jb$:

$$\frac{\mathrm{d}g_1(t)}{\mathrm{d}t} = p_1 g_1(t) + \frac{\partial p_1}{\partial \alpha} u(t), \tag{5.118}$$

$$\frac{\mathrm{d}f_1(t)}{\mathrm{d}t} = p_1 f_1(t) + g_1(t), \tag{5.119}$$

$$\frac{\mathrm{d}g_2(t)}{\mathrm{d}t} = p_2 g_2(t) + \frac{\partial p_2}{\partial \alpha} u(t), \tag{5.120}$$

$$\frac{\mathrm{d}f_2(t)}{\mathrm{d}t} = p_2 f_2(t) + g_2(t). \tag{5.121}$$

By using the matrix notations

$$\boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \quad \boldsymbol{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad \frac{\partial \boldsymbol{p}}{\partial \alpha} = \begin{bmatrix} \partial p_1/\partial \alpha \\ \partial p_2/\partial \alpha \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix}, \tag{5.122}$$

it follows that (5.118) ÷ (5.121) can be written in a compact form as

$$\frac{\mathrm{d}\boldsymbol{g}(t)}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{g}(t) + \frac{\partial \boldsymbol{p}}{\partial \alpha}u(t), \tag{5.123}$$

$$\frac{\mathrm{d}\boldsymbol{f}(t)}{\mathrm{d}t} = \boldsymbol{A}\boldsymbol{f}(t) + \boldsymbol{g}(t), \tag{5.124}$$

and by applying the similarity transformation described in the previous section it follows that

$$\frac{\mathrm{d}\hat{\boldsymbol{g}}(t)}{\mathrm{d}t} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^{-1}\hat{\boldsymbol{g}}(t) + \boldsymbol{V}\frac{\partial \boldsymbol{p}}{\partial \alpha}u(t), \tag{5.125}$$

$$\frac{\mathrm{d}\hat{\boldsymbol{f}}(t)}{\mathrm{d}t} = \boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^{-1}\hat{\boldsymbol{f}}(t) + \hat{\boldsymbol{g}}(t), \tag{5.126}$$

where $\boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^{-1}$ is given by (5.114). It is straightforward to derive that

$$\boldsymbol{V}\frac{\partial \boldsymbol{p}}{\partial \alpha} = \left[-\sqrt{2}\tfrac{\partial a}{\partial \alpha}\ \ -\sqrt{2}\tfrac{\partial b}{\partial \alpha}\right]. \tag{5.127}$$

Thus, the Eqs. (5.123) and (5.124) corresponding to the two complex-conjugated poles become after applying the similarity transformation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\hat{g}_1\\ \hat{g}_2\end{bmatrix} = \begin{bmatrix}a & -b\\ b & a\end{bmatrix}\begin{bmatrix}\hat{g}_1\\ \hat{g}_2\end{bmatrix} + \begin{bmatrix}-\sqrt{2}\partial a/\partial\alpha\\ -\sqrt{2}\partial b/\partial\alpha\end{bmatrix}u(t), \tag{5.128}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\hat{f}_1\\ \hat{f}_2\end{bmatrix} = \begin{bmatrix}a & -b\\ b & a\end{bmatrix}\begin{bmatrix}\hat{f}_1\\ \hat{f}_2\end{bmatrix} + \begin{bmatrix}\hat{g}_1\\ \hat{g}_2\end{bmatrix}. \tag{5.129}$$

If there are several pairs of complex conjugated poles, equations above will be true for any of these pairs and, by renaming $p \rightarrow p_m$, $\hat{g}_1 \rightarrow \hat{g}'_m$, $\hat{g}_2 \rightarrow \hat{g}''_m$, $\hat{f}_1 \rightarrow \hat{f}'_m$, $\hat{f}_2 \rightarrow \hat{f}''_m$, $a \rightarrow a_m$, $b \rightarrow b_m$, the synthesized circuit is shown in Fig. 5.28.

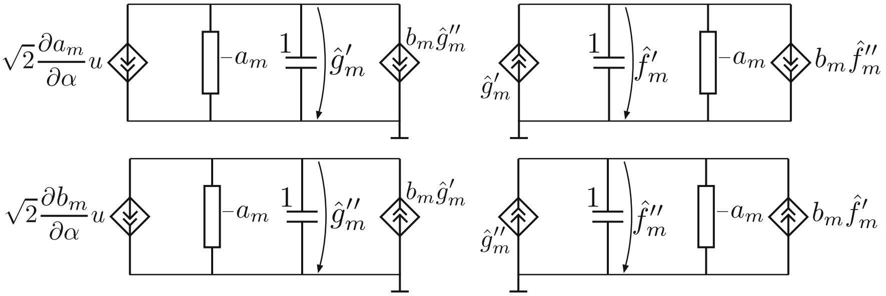


Fig. 5.28 Sub-circuit corresponding to a pair of complex conjugate poles

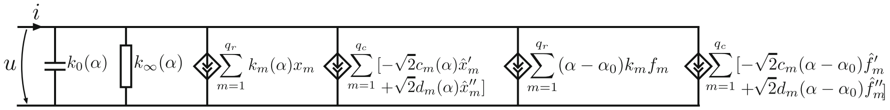

i
u
$k_0(\alpha)$
$k_\infty(\alpha)$
$\sum_{m=1}^{q_r} k_m(\alpha)x_m$
$\sum_{m=1}^{q_c} [-\sqrt{2}c_m(\alpha)\hat{x}'_m + \sqrt{2}d_m(\alpha)\hat{x}''_m]$
$\sum_{m=1}^{q_r} (\alpha-\alpha_0)k_m f_m$
$\sum_{m=1}^{q_c} [-\sqrt{2}c_m(\alpha-\alpha_0)\hat{f}'_m + \sqrt{2}d_m(\alpha-\alpha_0)\hat{f}''_m]$

Fig. 5.29 Parametric sub-circuit corresponding to the output equation

The new terms added in the output equations are

$$i_2(s,\alpha) = (\alpha-\alpha_0)\begin{bmatrix} k & k^* \end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = (\alpha-\alpha_0)\begin{bmatrix} k & k^* \end{bmatrix} \boldsymbol{V}^{-1}\hat{\boldsymbol{f}} = \tag{5.130}$$

$$= (\alpha-\alpha_0)\begin{bmatrix} -\sqrt{2}c & \sqrt{2}d \end{bmatrix}\hat{\boldsymbol{f}} = -\sqrt{2}c(\alpha-\alpha_0)\hat{f}_1 + \sqrt{2}d(\alpha-\alpha_0)\hat{f}_2.$$

In general, if the system has q poles out of which q_r are real and $q_c = (q-q_r)/2$ are pairs of complex conjugate poles, then the parametric synthesis will be done as follows: for each real pole $m = 1,\ldots,q_r$, let k_m be the residue corresponding to the pole; for each pair of complex conjugate poles $m = 1,\ldots,q_c$ let the pole be $p'_m = a_m + jb_m$, with the corresponding residue $k'_m = c_m + jd_m$. The equivalent circuit for the parametric output equation is shown in Fig. 5.29. It consists of the following elements connected in parallel:

- A parameterized capacitance $k_0(\alpha) = k_0 + (\alpha-\alpha_0)\partial k_0/\partial\alpha$,
- A parameterized conductance $k_\infty(\alpha) = k_\infty + (\alpha-\alpha_0)\partial k_\infty/\partial\alpha$,
- q_r voltage controlled current sources (having as parameter the parameterized value $k_m(\alpha) = k_m + (\alpha-\alpha_0)\partial k_m/\partial\alpha$, controlled by the voltages x_m),
- q_c voltage controlled current sources (having as parameter the parameterized value $-\sqrt{2}c_m(\alpha)$, controlled by the voltages $\hat{x}'_m$),
- q_c voltage controlled current sources (having the parameter $\sqrt{2}d_m(\alpha)$, controlled by the voltages $\hat{x}''_m$),
- q_r voltage controlled voltage sources (having the parameter $(\alpha-\alpha_0)k_m$, controlled by the voltages f_m,
- q_c voltage controlled current sources (having the parameter $-\sqrt{2}c_m(\alpha-\alpha_0)$, controlled by the voltages $\hat{f}'_m$),
- q_c voltage controlled current sources (having the parameter $\sqrt{2}d_m(\alpha-\alpha_0)$, controlled by the voltages $\hat{f}''_m$).

The voltages x_m are defined on the q_r subcircuits that correspond to real poles (Fig. 5.20), the voltages $\hat{x}'_m$, $\hat{x}''_m$ are defined on the q_c subcircuits that correspond to the pair of complex conjugate poles (Fig. 5.26), the voltages f_m are defined on the q_r subcircuits that correspond to real poles (Fig. 5.24), the voltages $\hat{f}'_m$ and $\hat{f}''_m$ are defined on the q_c subcircuits that correspond to the complex poles (Fig. 5.28).

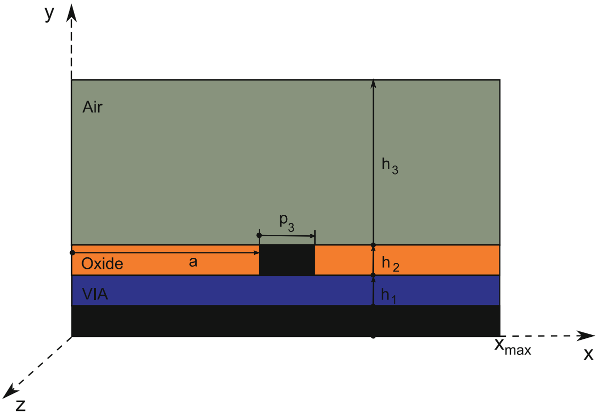


Fig. 5.30 Stripline parameterized structure

5.2.8 Case Study

In order to validate our approach and to evaluate different parametric models which can be extracted by the proposed procedure, several experiments have been performed on a test structure that consists of a microstrip (MS) transmission line having one Aluminum conductor embedded in a SIO_2 layer. The line has a rectangular cross-section, parameterized by several parameters (Fig. 5.30). The return path is the grounded surface placed at $y = 0$. The nominal values used are: $h_1 = 1\,\mu$m, $h_2 = 0.69\,\mu$m, $h_3 = 10\,\mu$m, $a = 130.5\,\mu$m, $p_1 = h_1$, $p_2 = h_2$, $p_3 = 3\,\mu$m, $x_{max} = 264\,\mu$m. In order to comply with designer's requirements, the model should include the field propagation along the line, taking into consideration the distributed parameters and the high frequency effects.

5.2.8.1 Validation of the Nominal Model

The first step of the validation refers to the simulation of the nominal case for which measurements (S parameters) are available from the European project FP5/Codestar (http://www.magwel.com/codestar/). By using dFIT + dELOB [52], at low frequencies, the following values are obtained:

$$R = 18.11\text{k}\Omega/\text{m}, \quad L = 322\text{nH/m}, \quad C = 213\text{pF/m}, \tag{5.131}$$

which are coherent with the values obtained from the measurements at low frequencies, and validates the grid used and the extension of the boundary used in the numerical model. Then, by using the method described in Sect. 5.2.4 the dependence of p.u.l. parameters with respect to the frequency was computed. The comparison between the resulting S parameters and the measurements is shown in

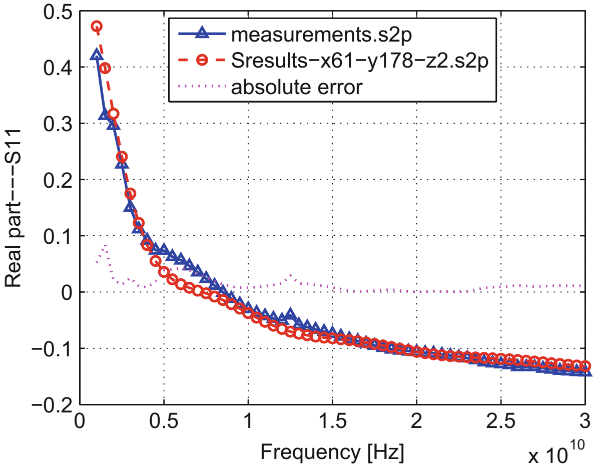


Fig. 5.31 Frequency characteristic Re(S_{11}): numerical model vs. measurements

Fig. 5.31 and it validates the nominal model. The sensitivities of the p.u.l. parameters are computed using the CHAMY software [40], by direct differentiation method applied to the state space equations [41]. They could also be computed by Adjoint Field Technique (AFT) [38, 53].

5.2.8.2 Parametric Models

In this section, the accuracy of several parametric models for the line capacitance is investigated.

The first sets of tests considered only one parameter that varies, namely the width of the line, p_3. The nominal value chosen was $p_3 = 3\,\mu\text{m}$ and samples in the interval $[1, 5]\,\mu\text{m}$ were considered. The reference result was obtained by simulated the samples separately (each sample was discretized and solved). These were compared with the approximate values obtained from models A and R (Fig. 5.32). As expected intuitively, the dependence w.r.t. p_3 is almost linear and the A model is better than the R model. Considering the relative variation of the parameters less than 15 % (which is the typical limit for the technological variations nowadays) the relative variation of the output parameter is obtained (Fig. 5.32, right). The errors of both affine and rational first order models for p.u.l. parameters are given in Table 5.2. Model A based on the first order Taylor series approximation has a maximal error for technologic variations 1.78 % for p.u.l. resistance when p_3 is variable, whereas model R has an approximation error of only 0.6 % for the same range of the technological variations for p.u.l. capacitance when p_3 is variable. Using (5.74) one can be easily identify which is the best model for any case.

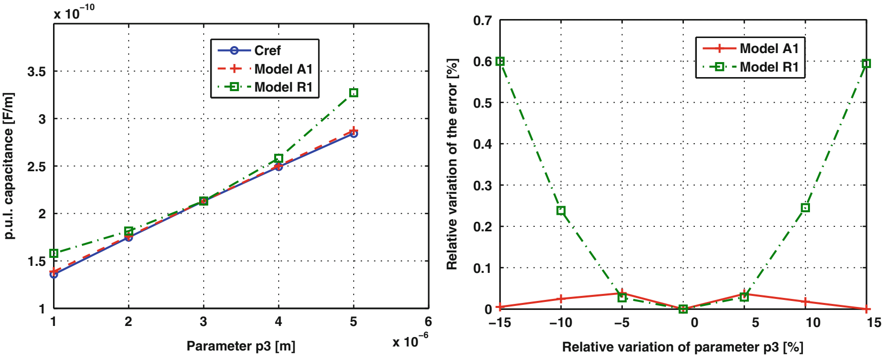


Fig. 5.32 *Left*: Reconstruction of the p.u.l. C from Taylor Series first order expansion; *Right*: Relative error w.r.t. the relative variation of parameter p_3

Table 5.2 Maximal errors [%] of p.u.l. parameters for technology variation of ± 15 %

Parameter	Quantity	Affine (A1)	Rational (R1)
p_1	L	0.11	0.15
	C	0.65	0.25
p_3	R	1.78	0.22
	L	0.34	0.04
	C	0.035	0.6

The second set of tests considered two parameters that vary simultaneously: p_1 and p_3. For reference, a set of samples in $[0.8, 1.2] \times [2, 4]$ μm were considered. The p.u.l. capacitance was approximated using the additive, rational and multiplicative models described above. In this case, a new model M is computed using an additive model for p_3 and a rational one for p_1, which is the best choice. Fig. 5.33, left compares the relative variation of the errors w.r.t. a relative variation of parameter p_1 for a variation of p_3 of 5 %. Model M provides lower errors (maximum error is 2 %) than models A (3.7 %) and R (2.2 %). Figure 5.33, right illustrates that in the range from 20 to 40 % model M is the best one if we look at the variation w.r.t. p_3 for a variation of p_1 of 10 %.

Thus, by using the appropriate multiplicative models in the modeling of the technological variability, the necessity of higher order approximations can be eliminated.

5.2.8.3 Frequency Dependent Parametric Models

In this case, the parameter considered variable is h_2. The sensitivity of the admittance with respect to this parameter has been calculated according to (5.61), using EM field solution. By applying Vector Fitting, a transfer function with 8 poles has been obtained. This conduced to an overdetermined system of size (236, 26) which has been solved with an accuracy (relative residual) of 3.7 % (Fig. 5.34-left).

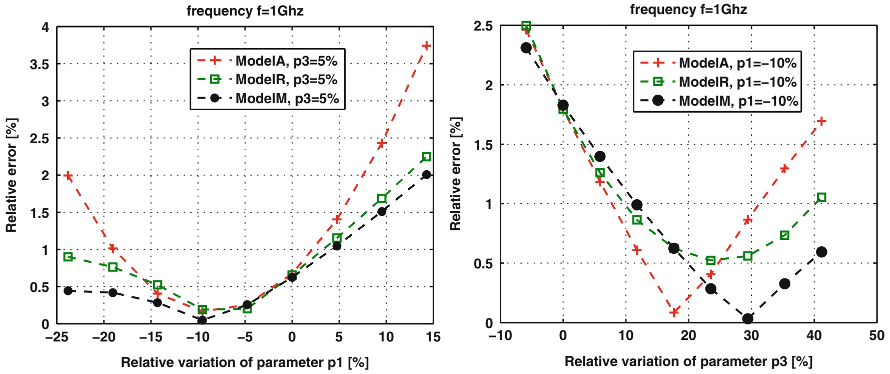


Fig. 5.33 *Left*: Relative error w.r.t. the relative variation of parameter p_1, for a variation of p_3 of 5 %; *Right*: Relative error w.r.t. the relative variation of parameter p_3, for a variation of p_1 of 10 %

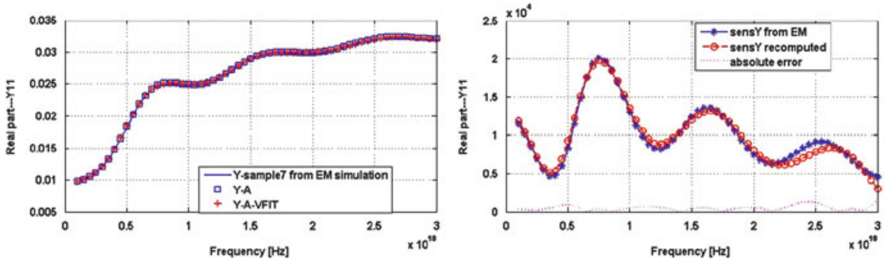


Fig. 5.34 *Left*: variation of the admittance sensitivity with respect to the frequency; *right*: reference simulation vs. answer obtained from the frequency dependent parametric model (5.94)

Finally, the relative error of the A-VFIT model is 1.09 % compared to the relative error of the A model which is 0.95 % for a relative variation of the parameter of 10 % (in Fig. 5.34-right the three curves are on top of each other).

5.2.9 Conclusions

The paper describes an effective procedure to extract reduced order parametric models of on-chip interconnects allowing model order reduction in coupled field (PDE) – circuit (DAE) problems. These models consider all EM field effects at high frequency, described by 3D-FW Maxwell equations. The proposed procedure is summarized by the following steps:

- Step 1 – Solve two field problems (2D EQS and FW-TM) and compute frequency dependent p.u.l. parameters and their sensitivities with respect to the geometric parameters that vary;
- Step 2 – Compute admittance for the real length of the line and its sensitivities with respect to the variable parameters;

- Step 3 – Choose the A/R variation model, i.e. the appropriate terminal excitation (admittance or impedance);
- Step 4 – Apply Vector Fitting for the nominal case in order to extract a rational model for the circuit function with respect to the frequency;
- Step 5 – Compute sensitivities of poles and residues of the circuit function by solving a least square problem;
- Step 6 – Assemble the frequency dependent parametric model by using the compact expression (5.94) or by synthesis of a SPICE like parametric netlist having frequency constant parameters.

Step 1 is dedicated to the extraction of the frequency dependent p.u.l. line parameters in a more robust and flexible way than the inversion of the equation of the short line segment. It is based on the solving of two field problems: 2D-EQS field which describes the transversal effects such as capacitive coupling whereas EMQS-TM field describes the longitudinal effects such as inductive, skin effect and eddy currents. The longitudinal propagation is described by the classic TL equations, but with frequency dependent p.u.l. line parameters.

Then (step 2), variability models for TL structures considering the dependency of p.u.l. parameters w.r.t. geometric parameters, at a given frequency were analyzed. A detailed study of the line sensitivity was made by using numeric techniques. For one parameter case, the proposed methods avoid the evaluation of higher order sensitivities, but keeping a high level of accuracy by introducing models based on a rational approximation in the frequency domain. The multi-parametric case has been analyzed, in addition, a multiplicative parametric model (M) has been proposed. This is based on the assumption that the quantity of interest can be expressed with separated variables, for which A and/or R models are used. Model M is sometimes better than A and R models obtained from Taylor Series expansion. Its specific terms (products of first order sensitivities) can thus approximate higher order, cross-terms of Taylor Series. In order to automatically select the best first order model for a multiparametric problem, the validity ranges of direct and reversed quantities have to be evaluated (step 3). Once we establish the best model (A or R) for each parameter, the M model will be easily computed by multiplication of individual submodels. Our numerical experiments with the proposed algorithm in all particular structures we investigated prove that the technological variability (e.g. ±20 % variation of geometric parameters, which is typical for the technology node of 65 nm) can be modeled with acceptable accuracy (relative errors under 5 %) using only first order parametric models for line parameters. This is one of the most important results of our research.

Next, a rational approximation in the frequency domain, obtained with Vector Fitting (step 4) is combined with a first order Taylor Series approximation. The sensitivities of poles, residues and constant terms are computed by solving an over-determined system of linear equations (step 5). The main advantage of this approach is that the final result is amenable to be synthesized with a small parameterized circuit (step 6). This method relies on the differential equation macromodel which is extended in order to take into account the variability. It also assumes that a first order Taylor Series expansion for the parameter that varies is accurate enough for the frequency range of interest. As shown in our previous work, there is a specific excitation type of terminals for which this assumption is acceptable for a certain frequency range. The passivity of the obtained circuit is guaranteed by the fact that the transfer function used as input for the synthesis procedure is passive as it is obtained by a fitting procedure with passivity enforcement.

Thus, the proposed method allows one to obtain parameterized reduced order circuits, having equivalent behavior as on-chip interconnects. These equivalent circuits described in SPICE language are extracted by considering all electromagnetic field effects in interconnects at very high frequency. This method applied to extract the reduced order model of the system described by PDE is a robust and efficient one, being experimentally validated.

The advantages of the proposed approach are:

- Its high accuracy, due to the consideration of all field effects at high frequency;
- Fast model extraction due to the reduced order of degrees of freedom in the numerical approach;
- High efficiency of the model order reduction step due to the use of Vector Fitting; in all interconnect studied cases, extracted models with an order less than 10 had an acceptable accuracy for designers.
- Simple geometric variability models based only on first order sensitivities, with extended valability domain due to the appropriate excitation;
- Appropriate variation model for frequency and length of interconnects, due to the transmission lines approach;
- The reduced SPICE models are simple and compact, containing ideal linear elements with lumped frequency independent constant parameters: capacitances, resistances and voltage controlled current sources; these element parameters have very simple affine variation in the case of the geometric variability.

The proposed method was successfully applied to model technological variability, without being necessary the use of higher order sensitivities.

5.3 Model Order Reduction and Sensitivity Analysis

Several types of parameters $\mathbf{p} = (\mathbf{d}, \mathbf{s}, \theta)$ influence the behaviour of electronic circuits and have to be taken into account when optimizing appropriate performance functions $f(\mathbf{p})$: design parameters $\mathbf{d}$, manufacturing process parameters $\mathbf{s}$, and operating parameters θ.[3] The impact of changes of design parameters, e.g., the width and length of transistors or the values of resistors, plays a key role in the design of

[3] Section 5.3 has been written by: Michael Striebel, Roland Pulch, E. Jan W. ter Maten, Zoran Ilievski, and Wil H.A. Schilders. Of parts of this Section extensions can be found in the Ph.D.-Thesis of the fourth author [95].

integrated circuits. Deviations from the nominal values defined in the design phase arise in the manufacturing process. Hence, to guarantee that the physical circuit shows the performances that were specified, the design has to be robust with respect to variations in the manufacturing phase. It has to be analysed how sensitive to parameter changes an integrated circuit and its performance is.

The manufacturing process parameters have a statistical impact, f.i., for the oxide thickness threshold. Examples of operating parameters are temperature and supply voltage.

Sensitivity can ease calculations on statistics (for instance by including the sensitivity in calculating the standard deviation of quantities that nearly linearly depend on independent normal distributed parameters [91]: if $\boldsymbol{F}(\boldsymbol{p}) \approx \boldsymbol{F}_0 + \boldsymbol{A}\,\boldsymbol{p}$ with $p_i \sim \mathrm{N}(0, \sigma_i)$ then $\sigma^2(\boldsymbol{F}_i) \approx \sum_j a_{ij}^2 \sigma_j^2$ ($\sigma(F_i)$, and σ_j being the standard deviation of F_i and p_j, resp.).

For optimizing one wants to minimize a performance function $f(\mathbf{p})$ while also several constraints have to be satisfied. The performance function $f(\mathbf{p})$ and the constraint functions $c(\mathbf{p})$ can be costly to evaluate and are subject to noise (for instance due to numerical integration effects). For both, the dependency on $\mathbf{p}$ can be highly nonlinear. Here there is interest in derivative free optimization [118], or to response surface model techniques [79, 80, 92, 117]. Partly these approaches started because in circuit simulation, sensitivities of $f(\mathbf{p})$ and $c(\mathbf{p})$ with respect to $\mathbf{p}$ are not always provided (several model libraries do not yet support the calculation of sensitivities). However, when the number of parameters increases adjoint sensitivity methods become of interest [74, 75]. For transient integration of linear circuits this is described in [76, 77]. In [96] a more general procedure is described that also applies to nonlinear circuits and retains efficiency by exploiting (nonlinear) techniques from Model Order Reduction.

A special sensitivity problem arises in verification of a design after layouting. During the verification the original circuit is extended by a huge number of ‘parasitics’, linear elements that generate additional couplings to the system. To reduce their effect the dominant parasitics should be detected in order to modify the layout.

Adjoint equations are also used for goal achievement. One example is in global error estimation in numerical integration [73, 99].

In this Section we describe adjoint techniques for sensitivity analysis in the time domain and indicate how MOR techniques like POD (Proper Orthogonal Decomposition) may fit here. Next we give a short introduction into Uncertainity Quantification which techniques provide an alternative way to perform sensitivity analysis. Here pMOR (parameterized MOR) techniques can be exploited.

5.3.1 Recap MNA and Time Integration of Circuit Equations

Modified Nodal Analysis (MNA) is commonly applied to model electrical circuits [86, 90]. Including the parameterization the dynamical behaviour of a circuit is then described by network equations of the general form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) + \mathbf{j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) = \mathbf{s}(t,\mathbf{p}). \tag{5.132a}$$

The state variables $\mathbf{x}(t,\mathbf{p}) \in \mathbb{R}^n$, i.e., the potentials at the network's nodes and the currents through inductors and current sources, are the unknowns in this system. They depend implicitly on the parameters gathered in the vector $\mathbf{p} \in \mathbb{R}^{n_p}$, because the voltage-charge and current-flux relations of capacitors and inductors, subsumed in $\mathbf{q}(\cdot,\cdot) \in \mathbb{R}^n$, the voltage-current relations of resistive elements, appearing in $\mathbf{j}(\cdot,\cdot) \in \mathbb{R}^n$ and the source terms $\mathbf{s}(\cdot,\cdot) \in \mathbb{R}^n$, i.e., the excitation of the circuit, may depend on the parameterization. The elements' characteristics $\mathbf{q},\mathbf{j}$ are usually nonlinear in the state variables $\mathbf{x}$, e.g., when transistors or diodes are present in the design at hand.

If, however, all elements behave linear with respect to $\mathbf{x}$, the MNA equations are of the form

$$\mathbf{C}(\mathbf{p})\dot{\mathbf{x}}(t,\mathbf{p}) + \mathbf{G}(\mathbf{p})\mathbf{x}(t,\mathbf{p}) = \mathbf{s}(t,\mathbf{p}), \tag{5.132b}$$

where $\dot{\mathbf{x}}$ denotes total differentiation, $(d/dt)\,\mathbf{x}(t,\mathbf{p})$ with respect to time. $\mathbf{C}(\mathbf{p})$ and $\mathbf{G}(\mathbf{p})$ are real $n \times n$-matrices that might depend nonlinearly on the parameters $\mathbf{p}$.

Usually the network equations (5.132) state a system of Differential-Algebraic Equations (DAEs), i.e., $(\partial/\partial\mathbf{x})\,\mathbf{q}(\mathbf{x},\mathbf{p})$ (or $\mathbf{C}(\mathbf{p})$) does not have full rank along the solution trajectory $\mathbf{x}(t,\mathbf{p})$.

In transient analysis the network equations (5.132) are solved on a time-interval $[t_0, t_{\mathrm{end}}] \subset \mathbb{R}$, where the parameter vector $\mathbf{p}$ is fixed and a (consistent) initial value $\mathbf{x}_{0,\mathbf{p}} := \mathbf{x}(t_0,\mathbf{p}) \in \mathbb{R}^n$ is chosen. As the system can usually not be solved exactly, numerical integration, e.g., BDF (backward differentiation formulas) or RK (Runge-Kutta) methods are used to compute approximations $\mathbf{x}_{i,\mathbf{p}} \approx \mathbf{x}(t_i,\mathbf{p})$ to the state variables on a discrete timegrid $\{t_0,\dots,t_l,\dots,t_K = t_{\mathrm{end}}\}$. For a timestep h form t_{l-1} to $t_l = t_{l-1}+h$, multistep methods, like the BDF schemes, approximate the time derivative $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(\mathbf{x}(t_l,\mathbf{p}),\mathbf{p})$ by some k-stage operator $\rho\mathbf{q}(\mathbf{x}_{l,\mathbf{p}},\mathbf{p}) := \alpha\mathbf{q}(\mathbf{x}_{l,\mathbf{p}},\mathbf{p}) + \boldsymbol{\beta}$, where $\alpha = \alpha(h) \in \mathbb{R}$ is the integration coefficient and $\boldsymbol{\beta} \in \mathbb{R}^n$ is made up of history terms $\mathbf{q}(x_{\mu,\mathbf{p}},\mathbf{p})$ at the timepoints t_μ for $\mu = l-k,\dots,l-1$. For the backward Euler method, e.g., we have

$$\rho\mathbf{q}(\mathbf{x}_{l,\mathbf{p}},\mathbf{p}) := \underbrace{\frac{1}{h}}_{=\alpha}\,\mathbf{q}(\mathbf{x}_{l,\mathbf{p}},\mathbf{p})\underbrace{-\frac{1}{h}\mathbf{q}(\mathbf{x}_{l-1,\mathbf{p}},\mathbf{p})}_{=\boldsymbol{\beta}}.$$

This results at each discretisation point $t_l \in \{t_0,t_1,\dots,t_K\}$ in a nonlinear equation for the state variable $\mathbf{x}_{l,\mathbf{p}}$ of the form

$$\alpha\mathbf{q}(\mathbf{x}_{l,\mathbf{p}},\mathbf{p}) + \boldsymbol{\beta} + \mathbf{j}(\mathbf{x}_{l,\mathbf{p}},\mathbf{p}) = \mathbf{s}(t_l,\mathbf{p}). \tag{5.133}$$

This nonlinear problem is usually solved with some Newton-Raphson technique, where in each underlying iteration $\nu = 1, 2, \ldots$ a linear system with a system matrix of the form

$$\mathbf{J}(\mathbf{x}_{l,\mathbf{p}}^{(\nu)}) = \left(\alpha \frac{\partial \mathbf{q}(\cdot, \mathbf{p})}{\partial \mathbf{x}} + \frac{\partial \mathbf{j}(\cdot, \mathbf{p})}{\partial \mathbf{x}}\right)(\mathbf{x}_{l,\mathbf{p}}^{(\nu)}) \tag{5.134}$$

appears. Typically, simplified Newton-Raphson iterqations may be applied. That is, only the evaluation at $\mathbf{x}_{l,\mathbf{p}}^{(1)}$ is involved. Note, also when applying a onestep method, like an RK-scheme, linear systems, made up of the Jacobian (5.134) arise.

5.3.2 *Sensitivity Analysis*

We encounter that the state variables $\mathbf{x}(t, \mathbf{p})$ implicitly depend on the the parameters $\mathbf{p} \in \mathbb{R}^{n_p}$. Hence, one is interested in how sensitive the behavior of the circuit with respect to variations in the parameters is. Thinking about "behavior of the system" we can basically have in mind

(i) How do the state variables vary with varying parameters?
(ii) How do measures derived from the state variables, e.g., the power consumption change with varying parameter?

Furthermore, due to usually nonlinear dependence of the element characteristics $\mathbf{q}$ and $\mathbf{j}$ or $\mathbf{C}$ and $\mathbf{G}$, respectively, on the parameters, we are interested in variations around a nominal value $\mathbf{p}_0 \in \mathbb{R}^{n_p}$.

In the following we will give a brief overview on the different kinds of sensitivities and how they can be treated numerically. For further reading we refer to the PhD thesis by Ilievski [95] and the papers by Daldoss et al. [78], Hovecar et al. [93], Cao et al. [74, 75] and Ilievski et al. [96].

5.3.2.1 State Sensitivity

In (transient) state sensitivity one is interested, how the trajectories of the state variables $\mathbf{x}$ vary with respect to the parameters $\mathbf{p}$ around the nominal setting $\mathbf{p}_0$. Hence, the goal is to compute

$$\boldsymbol{\chi}_{\mathbf{p}_0}(t) := \frac{d\mathbf{x}(t, \mathbf{p})}{d\mathbf{p}}\Big|_{\mathbf{p}=\mathbf{p}_0} \in \mathbb{R}^{n \times n_p}, \quad \text{for all } t \in [t_0, t_{\text{end}}]. \tag{5.135}$$

As described by Daldoss et al. [78] we linearize the nonlinear network equations (5.132b) around the nominal parameter set $\mathbf{p}_0$, i.e., we differentiate with respect to $\mathbf{p}$. We assume that the element functions $\mathbf{q}$, $\mathbf{j}$ are sufficiently smooth such that we can exchange the order of differentiation (Schwarz's theorem) and get:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\mathbf{C}_x(t)\cdot\boldsymbol{\chi}_{\mathbf{p}_0}(t)\right]+\mathbf{G}_x(t)\cdot\boldsymbol{\chi}_{\mathbf{p}_0}(t)=\mathbf{S}_p(t)-(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{C}_p(t)+\mathbf{G}_p(t)) \tag{5.136}$$

$$\begin{aligned}\text{with}\quad \mathbf{C}_x(t)&:=\frac{\partial\mathbf{q}}{\partial\mathbf{x}}(\mathbf{x}(t,\mathbf{p}_0),\mathbf{p}_0), & \mathbf{C}_p(t)&:=\frac{\partial\mathbf{q}}{\partial\mathbf{p}}(\mathbf{x}(t,\mathbf{p}_0),\mathbf{p}_0),\\ \mathbf{G}_x(t)&:=\frac{\partial\mathbf{j}}{\partial\mathbf{x}}(\mathbf{x}(t,\mathbf{p}_0),\mathbf{p}_0), & \mathbf{G}_p(t)&:=\frac{\partial\mathbf{j}}{\partial\mathbf{p}}(\mathbf{x}(t,\mathbf{p}_0),\mathbf{p}_0),\\ \mathbf{S}_p(t)&:=\frac{\partial\mathbf{s}}{\partial\mathbf{p}}(t,\mathbf{p}_0),\end{aligned}$$

where $\mathbf{C}_x(t),\mathbf{G}_x(t)\in\mathbb{R}^{n\times n}$ and $\mathbf{C}_p(t),\mathbf{G}_p(t),\mathbf{S}_p(t)\in\mathbb{R}^{n\times n_p}$ and $\mathbf{x}(t,\mathbf{p}_0)$ solves the network problem (5.132a).

The initial sensitivity value $\boldsymbol{\chi}_{\mathbf{p}_0}(t_0)=:\boldsymbol{\chi}_{p_0,0}=:\boldsymbol{\chi}_{p_0}^{\mathrm{DC}}$ can easily be calculated as the sensitivity of the DC-solution $\mathbf{x}(0,\mathbf{p}_0):=\mathbf{x}_{\mathbf{p}_0}^{\mathrm{DC}}$ of the network equation (5.132a), satisfying

$$\mathbf{j}(\mathbf{x}_{\mathbf{p}_0}^{\mathrm{DC}},\mathbf{p}_0)=\mathbf{s}(t_0,\mathbf{p}_0). \tag{5.137}$$

Obviously, the DAE (5.136) states a *linear* time varying dynamical system for the state sensitivity $\boldsymbol{\chi}_{\mathbf{p}_0}$, even when (5.132a) was nonlinear. We assume that we have used the backward Euler method to solve the network problem (5.132). Using the same time grid for solving the state sensitivity problem (5.136), we advance from time point t_{l-1} to $t_l=t_{l-1}+h$, i.e., we compute $\boldsymbol{\chi}_{\mathbf{p}_0,l}\approx\boldsymbol{\chi}_{\mathbf{p}_0}(t_l)$ again with the backward Euler by solving

$$\mathbf{M}_l\,\boldsymbol{\chi}_{\mathbf{p}_0,l}=\mathbf{rhs}_l \tag{5.138}$$

with

$$\begin{aligned}\mathbf{M}_l&:=\frac{1}{h}\mathbf{C}_x(t_l)+\mathbf{G}_x(t_l)\approx\frac{\partial\mathbf{q}}{\partial\mathbf{x}}(\mathbf{x}_{l,\mathbf{p}_0},\mathbf{p}_0)+\frac{\partial\mathbf{j}}{\partial\mathbf{x}}(\mathbf{x}_{l,\mathbf{p}_0},\mathbf{p}_0),\\ \mathbf{rhs}_l&:=\mathbf{S}_p(t_l)-(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{C}_p(t_l)+\mathbf{G}_p(t_l))+\frac{1}{h}\mathbf{C}_x(t_{l-1})\cdot\boldsymbol{\chi}_{\mathbf{p}_0,l-1}\\ &\approx\mathbf{S}_p(t_l)-\left(\frac{1}{h}\left(\frac{\partial\mathbf{q}}{\partial\mathbf{p}}(\mathbf{x}_{l,\mathbf{p}_0},\mathbf{p}_0)-\frac{\partial\mathbf{q}}{\partial\mathbf{p}}(\mathbf{x}_{l-1,\mathbf{p}_0},\mathbf{p}_0)\right)\right.\\ &\qquad\left.+\frac{\partial\mathbf{j}}{\partial\mathbf{p}}(\mathbf{x}_{l,\mathbf{p}_0},\mathbf{p}_0)\right)-\frac{1}{h}\frac{\partial\mathbf{q}}{\partial\mathbf{x}}(\mathbf{x}_{l-1,\mathbf{p}_0},\mathbf{p}_0)\cdot\boldsymbol{\chi}_{\mathbf{p}_0,l-1}.\end{aligned} \tag{5.139}$$

We note, that the partial derivatives with respect to $\mathbf{x}$ have already been computed in the transient analysis and are available, if they have been stored. Especially, the system matrix $\mathbf{M}_l$ is the same as we have used in applying the backward Euler method in the underlying simulation: within the Newton iterations these where the system matrices in the steps where convergence was recorded. Hence, also the decomposition of this matrix is available, such that the system could be solved efficiently. For schemes other than the backward Euler, we can also state, that the solution of the transient sensitivity problem (5.136) needs ingredients that are available (if they have been stored) from the solution of the network problem with the same method and the same step size. A reasoning for this and details on step size control and error estimation can be found in the paper by Daldoss et al. [78].

However, the sensitivities of the element functions $\mathbf{q}, \mathbf{j}$ and $\mathbf{s}$ have to be calculated. In total, the evaluation of the right-hand side $\mathbf{rhs}_l$ requires $\mathcal{O}(n_p \cdot n^2) + \mathcal{O}(n_p \cdot n)$ operations [96]. As in addition a lot of data has to be stored, computing the state sensitivities for circuits containing a large number of parameters is not tractable.

5.3.2.2 Observation Function Sensitivity

Often one is not interested in the sensitivity $\boldsymbol{\chi}_{\mathbf{p}_0}(t)$ of the states of the parameter dependent network problem (5.132) but rather in the sensitivity of some performance figures of the system, like e.g., power consumption. These measures can usually be described by some *observation function* $\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{p}) \in \mathbb{R}^{n_o}$ of the form

$$\boldsymbol{\Gamma}(\mathbf{x}, \mathbf{p}) = \int_{t_0}^{t_{\text{end}}} \mathbf{g}(\mathbf{x}, t, \mathbf{p})\, dt, \tag{5.140}$$

where the function $\mathbf{g} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_o}$ is such that the partial derivatives $\partial\mathbf{g} / \partial\mathbf{x}$ and $\partial\mathbf{g} / \partial\mathbf{p}$ exist and are bounded. Note that at the left-hand side of (5.140) $\mathbf{x} = \mathbf{x}(., \mathbf{p})$, which is a whole waveform in time.

The sensitivity of the observation function $\boldsymbol{\Gamma} : \mathbb{R}^n \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_o}$ around some nominal parameter set $\mathbf{p}_0 \in \mathbb{R}^{n_p}$, clearly is

$$\frac{d\boldsymbol{\Gamma}}{d\mathbf{p}}(\mathbf{x}(\mathbf{p}_0), \mathbf{p}_0) = \frac{\partial\boldsymbol{\Gamma}}{\partial\mathbf{x}}(\mathbf{x}(\mathbf{p}_0), \mathbf{p}_0)\, \frac{\partial\mathbf{x}}{\partial\mathbf{p}}(\mathbf{x}(\mathbf{p}_0), \mathbf{p}_0) + \frac{\partial\boldsymbol{\Gamma}}{\partial\mathbf{p}}(\mathbf{x}(\mathbf{p}_0), \mathbf{p}_0) \in \mathbb{R}^{n_o \times n_p}\,. \tag{5.141}$$

For problems where the sensitivity of a few observables, i.e., where n_o is small but the system depends on a large number n_p of parameters, the *adjoint method*, introduced by Cao et al. in [74, 75] is an attractive approach. In the mentioned papers, the observation sensitivity problem is derived for implicit differential equations of the form

$$\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}, t, \mathbf{p}) = \mathbf{0}. \tag{5.142}$$

Here we derive the observation sensitivity problem for problem (5.132a) as we usually encounter in circuit simulation. The idea however, follows the idea presented by Cao et al. in the papers mentioned.

The observation function's sensitivity is not calculated directly. Instead, an intermediate quantity $\boldsymbol{\lambda}$, defined by a dynamical system, the *adjoint model* [82] of the parent problem, is calculated.

5.3.2.3 Adjoint System for Sensitivity Analysis

Instead of considering the definition (5.140) of the observation function $\boldsymbol{\Gamma}$ directly, we define an augmented observation function

$$\boldsymbol{\Upsilon}(\mathbf{x},\mathbf{p}) := \boldsymbol{\Gamma}(\mathbf{x},\mathbf{p}) - \int_{t_0}^{t_{\text{end}}} \boldsymbol{\lambda}^T(t)\left[\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) + \mathbf{j}(\mathbf{x}(t,\mathbf{p}),\mathbf{p}) - \mathbf{s}(t,\mathbf{p})\right] dt, \tag{5.143}$$

which arises from coupling the dynamics and the observation function $\boldsymbol{\Gamma}$ by a Lagrangian multiplier $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n \times n_o}$ that we will define more precisely further on.

If $\mathbf{x}(t,\mathbf{p})$ solves the network equations (5.132a) for $\mathbf{p} = \mathbf{p}_0$ it holds $\boldsymbol{\Upsilon}(\mathbf{x},\mathbf{p}_0) = \boldsymbol{\Gamma}(\mathbf{x},\mathbf{p}_0)$ and also the sensitivities coincide:

$$\frac{d\boldsymbol{\Gamma}}{d\mathbf{p}}(\mathbf{x},\mathbf{p}_0) = \frac{d\boldsymbol{\Upsilon}}{d\mathbf{p}}(\mathbf{x},\mathbf{p}_0).$$

Note, that where it is clear from the context we omit in the following the specifications of the evaluation points, e.g., $(\mathbf{x},\mathbf{p}_0)$.

By the definitions (5.140) and (5.143) of the observation function and the augmented observation function, respectively, we get

$$\frac{d\boldsymbol{\Gamma}}{d\mathbf{p}} = \int_{t_0}^{t_{\text{end}}}\left(\frac{\partial\mathbf{g}}{\partial\mathbf{p}} + \frac{\partial\mathbf{g}}{\partial\mathbf{x}}\frac{\partial\mathbf{x}}{\partial\mathbf{p}}\right) dt - \int_{t_0}^{t_{\text{end}}} \boldsymbol{\lambda}^T(t)\left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{d\mathbf{q}}{d\mathbf{p}} + \frac{d\mathbf{j}}{d\mathbf{p}} - \frac{d\mathbf{s}}{d\mathbf{p}}\right) dt. \tag{5.144}$$

We have a closer look at the second integral and apply integration by parts:

$$\int_{t_0}^{t_{\text{end}}} \boldsymbol{\lambda}^T(t)\left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{d\mathbf{q}}{d\mathbf{p}}\right) dt = \left[\boldsymbol{\lambda}^T\frac{d\mathbf{q}}{d\mathbf{p}}\right]_{t_0}^{t_{\text{end}}} - \int_{t_0}^{t_{\text{end}}} \frac{d\boldsymbol{\lambda}^T}{dt}\frac{d\mathbf{q}}{d\mathbf{p}}\, dt.$$

Recombining this with the observation sensitivity (5.144) and expanding the total derivatives with respect to $\mathbf{p}$ we see

$$\begin{aligned}\frac{d\boldsymbol{\Gamma}}{d\mathbf{p}} =& - \left[\boldsymbol{\lambda}^T(t)\left(\mathbf{C}_x(t)\boldsymbol{\chi}_{\mathbf{p}_0}(t) + \mathbf{C}_p(t)\right)\right]_{t_0}^{t_\text{end}} \\ &+ \int_{t_0}^{t_\text{end}} \left(\frac{d\boldsymbol{\lambda}^T}{dt}\mathbf{C}_x(t) - \boldsymbol{\lambda}^T\mathbf{G}_x(t) + \boldsymbol{\gamma}_x(t)\right)\cdot\boldsymbol{\chi}_{\mathbf{p}_0}(t)\,dt \\ &+ \int_{t_0}^{t_\text{end}} \left(\boldsymbol{\gamma}_p(t) + \frac{d\boldsymbol{\lambda}^T}{dt}\mathbf{C}_p(t) - \boldsymbol{\lambda}^T\left[\mathbf{G}_p - \mathbf{S}_p(t)\right]\right) dt,\end{aligned} \tag{5.145}$$

where $\mathbf{C}_x, \mathbf{G}_x, \mathbf{C}_p, \mathbf{G}_p$ and $\mathbf{S}_p$ are the quantities defined in Eq. (5.136) and

$$\boldsymbol{\gamma}_x(t) := \frac{\partial\mathbf{g}}{\partial\mathbf{x}}(\mathbf{x}(t,\mathbf{p}_0)) \in \mathbb{R}^{n_o\times n}, \quad \boldsymbol{\gamma}_p(t) := \frac{\partial\mathbf{g}}{\partial\mathbf{p}}(\mathbf{x}(t,\mathbf{p}_0)) \in \mathbb{R}^{n_o\times n_p}$$

are the partial derivatives of the kernel of the observable and can thus be computed, if the solution trajectory $\mathbf{x}(t,\mathbf{p}_0)$ is known.

In the present form (5.145) the calculation of the observation function's sensitivity still demands to know the development of the state sensitivities $\boldsymbol{\chi}_{\mathbf{p}_0}(t)$. As the above considerations are valid for any smooth $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n\times n_o}$ we may choose this parameter such that the state sensitivity disappears in the equation. We have already seen that the sensitivity $\boldsymbol{\chi}_{\mathbf{p}_0}^{\text{DC}}$ of the circuit's operating point $\mathbf{x}_{\mathbf{p}_0}^{\text{DC}}$ can easily be calculated. Hence, choosing $\boldsymbol{\lambda}$ such that

$$\mathbf{C}_x^T(t)\frac{d\boldsymbol{\lambda}}{dt} - \mathbf{G}_x(t)^T\boldsymbol{\lambda} = -\boldsymbol{\gamma}_x^T(t) \tag{5.146a}$$

$$\text{and} \quad \boldsymbol{\lambda}^T(T)\mathbf{C}_x(T) = \mathbf{0}, \tag{5.146b}$$

the calculation of the observable function's sensitivity reduces to evaluating

$$\begin{aligned}\frac{d\boldsymbol{\Gamma}}{d\mathbf{p}} =& \boldsymbol{\lambda}^T(t_0)\left(\mathbf{C}_x(t_0)\boldsymbol{\chi}_{\mathbf{p}_0}^{\text{DC}} + \mathbf{C}_p(t_0)\right) - \boldsymbol{\lambda}^T(t_\text{end})\mathbf{C}_p(t_\text{end}) \\ &+ \int_{t_0}^{t_\text{end}} \left(\boldsymbol{\gamma}_p(t) + \frac{d\boldsymbol{\lambda}^T}{dt}\mathbf{C}_p(t) - \boldsymbol{\lambda}^T\left[\mathbf{G}_p - \mathbf{S}_p(t)\right]\right) dt.\end{aligned} \tag{5.147}$$

Equation (5.146a) inherits the basic structure of the underlying network problem (5.132a). Therefore, this equation defining the Lagrangian $\boldsymbol{\lambda}(t)$, is usually a DAE system. This linear system is called the *adjoint system* to the underlying network equation. For DAEs of index up to 1, the choice (5.146b) defines a consistent initial value [75]. For systems of index larger and equal to 2, the consistent initialisation is more difficult. As an initial value for $\boldsymbol{\lambda}$ is specified for the end of the interval $[t_0, t_{\text{end}}]$ of interest, the adjoint equation (5.146a) is solved backwards in time.

Several kind of observation functions also need $\dot{\mathbf{x}} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}$ in addition to $\mathbf{x}$. For instance when considering jitter one is interesting in the time difference between two subsequent times τ_1 and τ_2 when a specific unknown reaches or crosses a given value c (with equal signs of the time derivative). For the frequency of the jitter we have $f = 1/T = 1/(\tau_2 - \tau_1)$. Let the specific unknown be $\boldsymbol{x}_i(t, \boldsymbol{p})$. The time moment τ for which $\boldsymbol{x}_i(\tau, \boldsymbol{p}) = c$ may be determined by inverse interpolation between two time points t_1 and t_2 and known values $\boldsymbol{x}_i$ obtained by time integration such that $\boldsymbol{x}_i(t_1, \boldsymbol{p}) < c < \boldsymbol{x}_i(t_2, \boldsymbol{p})$. Of course τ depends on $\boldsymbol{p}$, so more precisely we have $\boldsymbol{x}_i(\tau(\boldsymbol{p}), \boldsymbol{p}) = c$. By differentiation we obtain: $\mathrm{d}(\tau)/\mathrm{d}\boldsymbol{p} = -[\mathrm{d}(\boldsymbol{x}_i)/\mathrm{d}t]^{-1}\mathrm{d}(\boldsymbol{x}_i)/\mathrm{d}\boldsymbol{p}$.

Hence we are also interested in a more general case than (5.140)

$$\mathbf{H}(\dot{\mathbf{x}}(\mathbf{p}), \mathbf{x}(\mathbf{p}), \mathbf{p}) = \int_{t_0}^{t_{\text{end}}} \mathbf{F}(\dot{\mathbf{x}}(t, \mathbf{p}), \mathbf{x}(t, \mathbf{p}), \mathbf{p})\mathrm{d}t. \tag{5.148}$$

By a similar analysis as presented in [96] for (5.144)–(5.145) we derive ($\hat{\dot{\mathbf{x}}} = \partial\dot{\mathbf{x}}/\partial\mathbf{p}$)

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\mathbf{p}}\mathbf{H}(\dot{\mathbf{x}}(\mathbf{p}), \mathbf{x}(\mathbf{p}), \mathbf{p}) &= \int_{t_0}^{t_{\text{end}}} \Big(\frac{\partial\mathbf{F}}{\partial\dot{\mathbf{x}}}\cdot\hat{\dot{\mathbf{x}}} + \frac{\partial\mathbf{F}}{\partial\mathbf{x}}\cdot\hat{\mathbf{x}} + \frac{\partial\mathbf{F}}{\partial\mathbf{p}}\Big)\,dt \\
&= -\Big(\zeta^T(t_{\text{end}})\mathbf{C}_x(t_{\text{end}}) - \frac{\partial\mathbf{F}}{\partial\dot{\mathbf{x}}}(t_{\text{end}})\Big)\boldsymbol{\chi}_{\mathbf{p}_0}(t_{\text{end}}) - \zeta^T(t_{\text{end}})\mathbf{C}_p(t_{\text{end}}) \\
&\quad + \Big(\zeta^T(t_0)\mathbf{C}_x(t_0) - \frac{\partial\mathbf{F}}{\partial\dot{\mathbf{x}}}(t_0)\Big)\boldsymbol{\chi}_{\mathbf{p}_0}(t_0) + \zeta^T(t_0)\mathbf{C}_p(t_0) \\
&\quad + \int_{t_0}^{t_{\text{end}}} \Big(\Big[\frac{d\zeta^T}{dt}\mathbf{C}_x - \zeta^T\mathbf{G}_x - \frac{d}{dt}(\frac{\partial\mathbf{F}}{\partial\dot{\mathbf{x}}}) + \frac{\partial\mathbf{F}}{\partial\mathbf{x}}\Big]\cdot\boldsymbol{\chi}_{\mathbf{p}_0} \\
&\quad + \frac{d\zeta^T}{dt}\mathbf{C}_{\mathbf{p}} - \zeta^T\Big(\mathbf{G}_p - \mathbf{S}_p\Big) + \frac{\partial\mathbf{F}}{\partial\mathbf{p}}\Big)dt.
\end{aligned} \tag{5.149}$$

which holds for any $\zeta(t) \in \mathbb{R}^{n \times n_o}$. If ζ is chosen such that

$$\mathbf{C}_x^T\frac{d\zeta}{dt} - \mathbf{G}_x^T\zeta = \frac{d}{dt}(\frac{\partial\mathbf{F}}{\partial\dot{\mathbf{x}}})^T - \Big(\frac{\partial\mathbf{F}}{\partial\mathbf{x}}\Big)^T, \tag{5.150}$$

$$\text{with 'initial' value} \quad \mathbf{C}_x^T\zeta(t_{\text{end}}) = (\frac{\partial\mathbf{F}}{\partial\dot{\mathbf{x}}})^T(t_{\text{end}}), \tag{5.151}$$

a significant reduction occurs in (5.149) and $\hat{\boldsymbol{x}}(t)$ is not explicitly needed. This generalizes the result in [96] (see also [64]). Note that $\boldsymbol{\chi}_{\mathbf{p}_0}(0) = \hat{\mathbf{x}}(0, \mathbf{p}_0) = \hat{\mathbf{x}}_{\text{DC}}(\mathbf{p}_0)$, which is the sensitivity of the DC-solution, which one needs to determine explicitly. Some efficiency is gained by calculating $\zeta^T(0)\mathbf{C}_x(0)\hat{\mathbf{x}}_{\text{DC}} = [\mathbf{C}_x^T(0)\zeta(0)]^T\hat{\mathbf{x}}_{\text{DC}}$ (when $n_p \gg 1$). Note however that (5.151) can be satisfied only when the right-hand side is in the range of $\mathbf{C}_x^T$. Because in (5.150)–(5.151) the right-hand sides are evaluated at $\mathbf{x}(t, \mathbf{p})$, in general, the solution ζ will depend on $\mathbf{p}$, even in the case of constant matrices $\mathbf{C}_x$ and $\mathbf{G}_x$. This is in contrast to [64].

Summing up, the *backward adjoint method* for computing the sensitivity of the observable $\boldsymbol{\Gamma}$ with respect to parameter variations around a nominal parameter setting $\mathbf{p}_0$ is carried out by the following steps

1. Solve the network DAE (5.132a) for $\mathbf{x} = \mathbf{x}(t, \mathbf{p}_0)$, on the interval $[t_0, t_{\text{end}}]$;
2. Solve the backward adjoint problem (5.146a), subject to the initial condition (5.146b) for $\boldsymbol{\lambda}$ on the interval $[t_{\text{end}}, t_0]$, i.e., backward in time;
3. Compute the observable sensitivity $d\boldsymbol{\Gamma} \,/\, d\mathbf{p}$ using the expression (5.147).

Carrying out the backward adjoint method, one has to consider several aspects we do not address here. Amongst these are the evaluations of the partial derivatives like $\mathbf{C}_x$ along the solution trajectory. On the one hand, these derivatives are usually not available as a closed function but are approximated by finite differences. On the other hand, the evaluation points, i.e., points on the trajectory $\mathbf{x}(t, \mathbf{p}_0)$ are also available as approximations only. Furthermore, the integral in the formula (5.147) has to be approximated by a numerical quadrature. The nodes needed in the according scheme may not be met exactly during transient simulation and/or during the backward integration of the adjoint problem. For further reading on these problems we refer to [95, 96].

However, leaving all these aspects aside, one has to integrate two dynamical systems numerically. First the (nonlinear) forward problem (5.132a) for the states and then the linear backward problem (5.146a). The contribution of the COMSON project for transient sensitivity analysis was to add model order reduction (MOR) to the process. More precisely, the idea elaborated during the project was to solve an order reduced variant of the backward problem where the data needed to apply the reduction is calculated from the forward solving phase. In the next section we will describe the very basic idea of MOR and give a brief introduction to the specific technique that was used in this project.

5.3.3 Model Order Reduction (with POD)

Solving a dynamical system with any numerical scheme implies to set up and solve a series of linear equations. In circuit simulation typically the dimension of these systems are in a range of 10^5–10^9. Both the evaluation of the system matrices and right-hand sides, e.g., $\mathbf{M}_l$ and $\mathbf{rhs}_l$ in (5.138)–(5.139), as well as solving the system, i.e., decomposing the system matrices, is computationally costly.

However, in circuit design often the main interest is the analysis of how a circuit block processes an input signal, e.g., if some input signal is amplified or damped by the circuit. That means, one may not be interested in all n internal state variables but only in a limited selection. This concern is described by an input-output variant of the network model. For a linear network problem (5.132b), omitting the parameters for ease of notation, e.g., the corresponding input-output system reads

$$\begin{aligned} \mathbf{C}\dot{\mathbf{x}}(t) + \mathbf{G}\mathbf{x}(t) &= \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{L}\mathbf{x}(t), \end{aligned} \tag{5.152}$$

where $\mathbf{u}(t) \in \mathbb{R}^m$ and $\mathbf{y}(t) \in \mathbb{R}^q$ are the input and the output of the system, injected to and extracted from the system by the matrices $\mathbf{B} \in \mathbb{R}^{n\times m}$ and $\mathbf{L} \in \mathbb{R}^{q\times n}$.

As in an input-output setting, the states $\mathbf{x}$ represent an auxiliary variable only. The idea of MOR is to replace the high-dimensional dynamical system (5.152) by

$$\begin{aligned} \hat{\mathbf{C}}\dot{\mathbf{z}}(t) + \hat{\mathbf{G}}\mathbf{z}(t) &= \hat{\mathbf{B}}\mathbf{u}(t), \\ \tilde{\mathbf{y}}(t) &= \hat{\mathbf{L}}\mathbf{z}(t), \end{aligned} \tag{5.153}$$

where $\mathbf{z}(t) \in \mathbb{R}^r$ and the system matrices $\hat{\mathbf{C}}, \hat{\mathbf{G}} \in \mathbb{R}^{r\times r}$, $\hat{\mathbf{B}} \in \mathbb{R}^{r\times m}$ and $\hat{\mathbf{L}} \in \mathbb{R}^{q\times r}$ are chosen such that $r \ll n$ and $\tilde{\mathbf{y}}(t) \approx \mathbf{y}(t)$.

There are various methods to construct the reduced variant (5.153) from the full problem (5.152). We refer to Chapter 4 for an overview, as well as to [63, 67, 69, 70, 103, 108–110, 121] for further studies.

A large class of MOR methods are based on projection. These methods determine a subspace of dimension r, spanned by a basis of vectors $\mathbf{v}_i \in \mathbb{R}^n$ $(i = 1, \dots, r)$. The original state vector $\mathbf{x}(t)$ is approximated by an element of this subspace that can be written in the form $\mathbf{V}\mathbf{z}(t)$, where $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathbb{R}^{n\times r}$. Hence, one replaces $\mathbf{x}(t)$ by $\mathbf{V}\mathbf{z}(t)$ in (5.152) and projects the equation onto the space subspaces spanned by the columns of $\mathbf{V}$ by a Galerkin approach. In this way, a dynamical system (5.153) emerges where the system matrices are given by

$$\hat{\mathbf{C}} := \mathbf{V}^T\mathbf{C}\mathbf{V}, \quad \hat{\mathbf{G}} := \mathbf{V}^T\mathbf{G}\mathbf{V}, \quad \hat{\mathbf{B}} := \mathbf{V}^T\mathbf{B}, \quad \hat{\mathbf{L}} := \mathbf{L}\mathbf{V}. \tag{5.154}$$

5.3.3.1 Proper Orthogonal Decomposition

While other MOR methods start operating from the matrices $\mathbf{C}, \mathbf{G}, \mathbf{B}$ and $\mathbf{L}$, the method of Proper Orthogonal Decomposition (POD) constructs the matrix $\mathbf{V}$, whose columns span the reduced space the system (5.152) is projected on, from the space that is spanned by the trajectory $\mathbf{x}(t)$, i.e., the solution of the dynamical system. The method applies to nonlinear systems as well.

Recall, that our aim is to construct a reduced model for the backward adjoint problem (5.146). As this is a linear system from which we know the system matrices only after a solution of the underlying forward network problem (5.132a), POD seems to be the best choice for this task.

The mission POD fulfills is to find a subspace approximating a given set of data in an optimal least-squares sense. The basis of this approach is known also as *Principal Component Analysis* and *Karhunen-Loève theorem* from picture and data analysis.

The mathematical formulation of POD [103, 107, 121] is as follows: Given a set of K datapoints $\mathbf{X} := \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$, a subspace $S \subset \mathbb{R}^n$ is searched for that minimizes

$$\|\mathbf{X} - \varrho\mathbf{X}\|_2^2 := \frac{1}{K}\sum_{k=1}^{K} \|\mathbf{x}_k - \varrho\mathbf{x}_k\|_2^2, \tag{5.155}$$

where $\varrho : \mathbb{R}^n \to S$ is the orthogonal projection onto S, which has $\{\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_r\}$ as an orthonormal basis of S.

This problem is solved, applying the Singular Value Decomposition (SVD) to the matrix $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_K) \in \mathbb{R}^{n\times K}$, which is called *snapshot matrix*, as its columns are (approximations to) the solution of the dynamical system (5.152) at timepoints $t_1, \dots, t_K \in [t_0, t_{\text{end}}]$. The SVD applied to the matrix $\mathbf{X}$, provides three matrices:

$$\boldsymbol{\Phi} \in \mathbb{R}^{n\times n} \quad \text{orthogonal,}$$

$$\boldsymbol{\Psi} \in \mathbb{R}^{K\times K} \quad \text{orthogonal,}$$

$$\boldsymbol{\Sigma} = \text{diag}\,(\sigma_1, \dots, \sigma_\nu) \in \mathbb{R}^{\nu\times\nu} \quad \text{with } \sigma_1 \geq \dots \geq \sigma_\nu > \sigma_{\nu+1} = \ldots = \sigma_K = 0,$$

such that

$$\mathbf{X} = \boldsymbol{\Phi}\begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}\boldsymbol{\Psi}^T, \tag{5.156}$$

where the columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are left and right eigenvectors, respectively, and $\sigma_1, \dots, \sigma_\nu$ are the singular values of $\mathbf{X}$.

Then, for any $r \leq \nu$, taking $\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_r$ as the first r columns of the matrix $\boldsymbol{\Phi}$ is optimal in the sense that it minimizes the projection mismatch (5.155).

Both cases, $n \geq K$ and $n \leq K$, are allowed; in practice one often has $n \gg K$.

Finally, the MOR projection matrix $\mathbf{V}$ in (5.154) is chosen made up of these basis vectors:

$$\mathbf{V} := (\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_r) \in \mathbb{R}^{n\times r}\,.$$

To understand why the first r columns of $\boldsymbol{\Phi}$ solve the minimization problem (5.155) one can recall that for $i = 1, \dots, n$ the ith column $\boldsymbol{\varphi}_i$ of $\boldsymbol{\Phi}$ is actually an eigenvector of the correlation (or covariance) matrix of the snapshots with σ_i^2 as eigenvalue:

$$\mathbf{X}\mathbf{X}^T\boldsymbol{\varphi}_i = \sigma_i^2\boldsymbol{\varphi}_i\,.$$

Algorithm 5.1 BRAM: Backward Reduced Adjoint Method

```
Integrate (5.132a) and store the solutions $\mathbf{x}(t_i, \mathbf{p})$
Build the snapshot matrix $\mathbf{X} = [\mathbf{x}(t_0, \mathbf{p}), \ldots, \mathbf{x}(t_N, \mathbf{p})]$ (where $t_N = t_{\text{end}}$)
Determine the singular value decomposition $\mathbf{X} = \boldsymbol{\Phi}^T \boldsymbol{\Sigma} \boldsymbol{\Psi}$ and dominant singular values $\sigma_1, \ldots, \sigma_r$.
Determine the Proper Orthogonal Decomposition (POD) time-independent projection matrix $\mathbf{V}$, such that $\mathbf{x} \approx \mathbf{V}\tilde{\mathbf{x}}$ and $\frac{d}{dt}\mathbf{x} \approx \mathbf{V}\frac{d}{dt}\tilde{\mathbf{x}}$
if (BRAM II) then
    Include a second forward time integration, now for the reduced system of equations.
end if
Integrate (5.157) backward in time using reduced matrices $\mathbf{V}^{\star}\mathbf{C}_x^T\mathbf{V}$ and $\mathbf{V}^T\mathbf{G}_x^T\mathbf{V}$ and the projected right-hand side $\mathbf{V}^T[\frac{d}{dt}(\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}})^T - \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^T]$
```

Intuitively the correlation matrix $\mathbf{X}\mathbf{X}^T$ detects the principal directions in the data cloud that is made up of the snapshots $\mathbf{x}_1, \ldots, \mathbf{x}_K$. The eigenvectors and eigenvalues can be thought of as directions and radii of axes of an ellipsoid that incloses the cloud of data. Then, the smaller the radii of one axis is, the less information is lost if that direction is neglected.

We abandon to explain the derivation of POD in detail here as in literature e.g., [63, 103, 121] this is well explained. For details on the accuracy of MOR with POD we refer to papers by Petzold et al. [94, 107].

5.3.4 The BRAM Algorithm

In [96] it was observed that a forward analysis in time of (5.132a) automatically provides provides snapshots $\mathbf{x}(t_i, \mathbf{p})$ at time points t_i. This can lead to a reduced system of equations for $\zeta(t) = \mathbf{V}\tilde{\zeta}$ in (5.150)–(5.151)

$$\mathbf{V}^T\mathbf{C}_x^T\mathbf{V}\frac{d\tilde{\zeta}}{dt} - \mathbf{V}^T\mathbf{G}_x^T\mathbf{V}\tilde{\zeta} = \mathbf{V}^T\frac{d}{dt}(\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}})^T - \mathbf{V}^T\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^T, \tag{5.157}$$

$$\text{with 'initial' value} \quad \mathbf{V}^T\mathbf{C}_x^T\mathbf{V}\tilde{\zeta}(t_{\text{end}}) = \mathbf{V}^T(\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}})^T(t_{\text{end}}), \tag{5.158}$$

Then the overall algorithm is described in Algorithm 5.1, without the lines 5–7. Here it is assumed that the matrices are saved after the forward simulation. It is also assumed that for the adjoint system the same step sizes are used as in the forward run. If not, additional interpolation has to be taken into account to determine the reduced matrices at intermediate solutions and also effort has to be spent in LU-decomposition.

Apart from this discussion, the question is why this should work in general (apart from special cases in [96]). The solution $\mathbf{x} \approx \mathbf{V}\tilde{\mathbf{x}}$ depends on the right-hand side $\mathbf{s}$ of (5.132a). Clearly $\mathbf{V}^T\mathbf{s}$ should contain the dominant behaviour of $\mathbf{s}$. If

$\mathbf{V}^T[\frac{d}{dt}(\frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}})^T - \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^T]$ behaves similarly when compared to the right-hand side of (5.150) we may expect a similar good approximation for the solution $\zeta \approx \mathbf{V}\tilde{\zeta}$. Because the right-hand side of (5.157) does not depend on ζ this can be checked in advance, before solving (5.157). In the case of power loss through a resistor we have $\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^T = (\mathbf{A}\mathbf{x})^T$ (for some matrix $\mathbf{A}$) and we have to check if $(\mathbf{A}\mathbf{x})^T \approx \mathbf{V}^T\tilde{\mathbf{x}}^T\mathbf{V}^T\mathbf{A}^T$.

Another point of attention is that the projection matrix $\mathbf{V}$ found implies that we more or less are looking to the sensitivity of the solution $\tilde{\mathbf{x}}$ of

$$\frac{d}{dt}[\mathbf{V}^T\mathbf{q}(\mathbf{V}\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p})] + \mathbf{V}^T\mathbf{j}(\mathbf{V}\tilde{\mathbf{x}}(t,\mathbf{p}),\mathbf{p}) = \mathbf{V}^T\mathbf{s}(t,\mathbf{p}) \tag{5.159}$$

rather than for the solution $\boldsymbol{x}$ of (5.132a). By this it is clear that $\mathbf{V}$ depends on $\mathbf{p}$ and thus

$$\mathbf{x}(t,\mathbf{p}) \approx \mathbf{V}(\mathbf{p})\tilde{\mathbf{x}}(t,\mathbf{p}) \Longrightarrow \frac{\partial \mathbf{x}}{\partial \mathbf{p}} \approx \frac{\partial \mathbf{V}}{\partial \mathbf{p}}\tilde{\mathbf{x}} + \mathbf{V}\frac{\partial \tilde{\mathbf{x}}}{\partial \mathbf{p}}. \tag{5.160}$$

The question is: can we ignore the first term at the right-hand side of (5.160). Here the last term represents the change inside the space defined by the span of the columns of $\mathbf{V}$. The first term represents the effect by the change of this space itself. One may expect that this term is smaller than the last term ('the first term will in general require more energy'), especially when the reduction is more or less determined by topology. In several tests we made, this first term indeed was much smaller than the other term.

Note that we not intend to solve (5.159) by using a fixed projection matrix $\mathbf{V}$, valid for $\mathbf{p} = \mathbf{p}_0$, for several different values of $\mathbf{p}$. The danger of obtaining improper results when doing this was pointed out by [83]. Contrarily, we always apply an up-to-date matrix $\mathbf{V}(\mathbf{p})$. However, this example shows that $\frac{\partial \mathbf{V}}{\partial \mathbf{p}}$ is not always negligible.

One can collect $\tilde{\mathbf{V}} = [\mathbf{V}(\mathbf{p}_1), \ldots, \mathbf{V}(\mathbf{p}_k)]$ and apply an additional SVD to $\tilde{\mathbf{V}}$. This procedure provides a larger, uniform, projection matrix $\mathbf{V}$.

In [95] the parameter dependency of the singular values for POD was analysed for a battery charger, for a ring oscillator, and for a car transceiver example. Also the nr of dominant singular values as function of $\mathbf{p}$ was studied. Finally the angle between the subspaces for different $\mathbf{p}$ was studied. Note that one can use a matlab function for this based on the algorithm by Knyazev-Argentati [98].

Finally, in [95] a modification was introduced in Algorithm 5.1 by introducing the lines 5–7. Note that the additional step 6 is cheap. We obtain the solution of the POD-reduced system. In [94, 103, 107, 121] error estimates are determined for the approximation error of the POD approximation. Actually, in step 8, BRAM II determines the sensitivity of the POD solution. In Fig. 5.35 [95] the singular values of POD after 3,500 snapshots within a simulation from $t_0 = 0\,\text{ms}$ and $t_{\text{end}} = 200\,\text{ms}$ for a Li-ion charger for different values of the area of a capacitor. The parameter

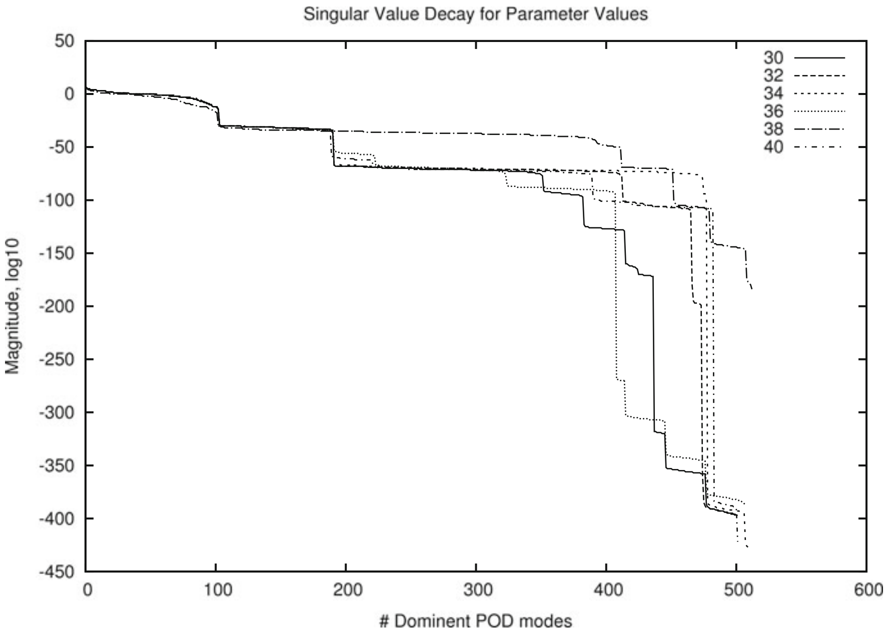


Fig. 5.35 Singular values of POD after 3,500 snapshots for a Li-ion charger for different values of the area of a capacitor

p took values $p = 30, 32, 34, 36, 38, 40$. Clearly the first 100 singular values are enough for a good reconstruction, which as a by-product als shows a high potential for the application of the BRAM methods as the dimension of the problem can be reduced by roughly a factor 35. In Fig. 5.36 [95] the angle in the rotation of the principle vector is studied, the nominal being for $p = 30$. The apparent jump to 90° rotation near the cut off point is due to matrix diagonal zero padding introduced in the general case for principle vector analysis. These large 90° rotations are not due to principle vectors influenced by parameter changes and should not be taken into account.

5.3.5 Sensitivity by Uncertainty Quantification

A modern approach to Uncertainty Quantification is to expand a solution $\mathbf{x}(t, \mathbf{p})$ in a series of orthogonal polynomials in which the $\boldsymbol{p}$ is argument of the (multidimensional) polynomials and the t appears in the coefficients. If the $\mathbf{p}$ are subject to variations such a representation is called a generalized Polynomial Chaos (gPC) expansion. Having established the expansion, this provides facilities similar like a response surface model: fast and accurate statistics and sensitivity.

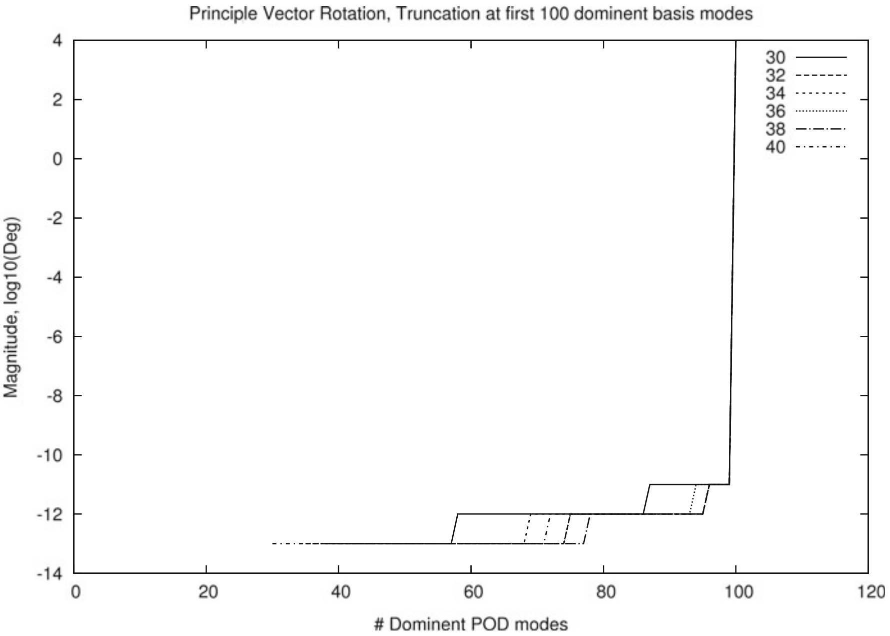


Fig. 5.36 Principle vector rotation as a function of the capacitor area for the problem in Fig. 5.35

In this section we shortly summarize some basic items. We also point out how a strategy for parameterized Model Order Reduction (pMOR) fits here. This strategy contains a generalization of one of the pMOR algorithms described in Sect. 5.1 of this Chapter.

We will denote parameters by $\mathbf{p} = (p_1, \ldots, p_q)^T$ and assume a probability space given $(\Omega, \mathscr{A}, \mathscr{P})$ with $\mathscr{P} : \mathscr{A} \rightarrow \mathbb{R}$ (measure; in our case the range will be [0, 1]) and $\mathbf{p} : \Omega \rightarrow Q \subseteq \mathbb{R}^q$, $\omega \mapsto \mathbf{p}(\omega)$. Here we will assume that the p_i are independent random variables, with factorizable joint probability density $\rho(\mathbf{p})$.

For a function $f : Q \rightarrow \mathbb{R}$, the mean or expected value is defined by

$$< f >= \int_\Omega f(\mathbf{p}(\omega))\mathrm{d}\mathscr{P}(\omega) = \int_Q f(\mathbf{p})\ \rho(\mathbf{p})\mathrm{d}\mathbf{p}. \qquad (5.161)$$

A bilinear form $< f, g >$ is defined by

$$< f, g >= \int_Q f(\mathbf{p})\ g(\mathbf{p})\ \rho(\mathbf{p})\mathrm{d}\mathbf{p} =< f\ g > . \qquad (5.162)$$

The last form is convenient when products of more functions are involved. Similar definitions hold for vector- or matrix-valued functions $\mathbf{f} : Q \rightarrow \mathbb{R}^{m \times n}$.

We assume a complete orthonormal basis of polynomials $(\phi_i)_{i \in \mathbb{N}}$, $\phi_i : \mathbb{R}^q \rightarrow \mathbb{R}$, given with $< \phi_i, \phi_j >= \delta_{ij}$ $(i, j, \geq 0)$. When $q = 1$, ϕ_i has degree i. To treat

Table 5.3 One-dimensional orthogonal polynomials related to well-known probability density functions

Distribution	Polynomial	Weight function	Support range
Gaussian	Hermite $H_n(p)$	$e^{-\frac{p^2}{2}}$	$(-\infty, \infty)$
Uniform	Legendre $P_n(p)$	1	$[-1, 1]$
Beta	Jacobi $P_n^{\alpha,\beta}(p)$	$(1-p)^\alpha(1+p)^\beta$	$[-1, 1]$
Exponential	Laguerre $L_n(p)$	e^{-p}	$[0, \infty)$
Gamma	Generalized Laguerre $L_n^{(\alpha)}(p)$	$p^\alpha e^{-p}$	$[0, \infty)$

a uniform distribution (i.e., for studying effects caused by robust variations) one can use Legendre polynomials; for a Gaussian distribution one can use Hermite polynomials [100, 123, 124]. Some one-dimensional polynomials are mentioned in Table 5.3. A polynomial $\phi_{\mathbf{i}}$ on $\mathbb{R}^q$ can be defined from one-dimensional polynomials: $\phi_{\mathbf{i}}(\mathbf{p}) = \prod_{d=1}^{q} \phi_{i_d}(p_d)$. Actually $\mathbf{i}$ orders a vector $\mathbf{i} = (i_1, \ldots, i_q)^T$; however we will simply write ϕ_i, rather then $\phi_{\mathbf{i}}$. An example is given in (5.163), using Legendre polynomials. Note that, due to normalization, $L_0(p) = 1/\sqrt{2}$, $L_1(p) = \sqrt{3/2}\,p$, $L_2(p) = \frac{1}{2}\sqrt{\frac{5}{3}}\,(3p^2-1)$ – see also [87]. In [88] one finds algorithms how to efficiently generate orthogonal polynomials from a given weight function.

$$\begin{aligned}
\phi_0(\mathbf{p}) &= L_0(p_1)\,L_0(p_2),\\
\phi_1(\mathbf{p}) &= L_1(p_1)\,L_0(p_2),\\
\phi_2(\mathbf{p}) &= L_0(p_1)\,L_1(p_2),\\
\phi_3(\mathbf{p}) &= L_2(p_1)\,L_0(p_2),\\
\phi_4(\mathbf{p}) &= L_1(p_1)\,L_1(p_2),\\
\phi_5(\mathbf{p}) &= L_0(p_1)\,L_2(p_2).
\end{aligned} \tag{5.163}$$

We will denote a dynamical system by

$$\mathbf{F}(\mathbf{x}(t,\mathbf{p}), t, \mathbf{p}) = 0, \quad \text{for } t \in [t_0, t_1]. \tag{5.164}$$

Here $\mathbf{F}$ may contain differential operators. The solution $\mathbf{x} \in \mathbb{R}^n$ depends on t and on $\mathbf{p}$. In addition initial and boundary values are assumed. In general these may depend on $\mathbf{p}$ as well.

A solution $\mathbf{x}(t,\mathbf{p}) = (x_1(t,\mathbf{p}), \ldots, x_n(t,\mathbf{p}))^T$ of the dynamical system becomes a random process. We assume that second moments $< x_j^2(t,\mathbf{p}) >$ are finite, for all $t \in [t_0, t_1]$ and $j = 1, \ldots, n$. We express $\mathbf{x}(t,\mathbf{p})$ in a Polynomial Chaos expansion

$$\mathbf{x}(t,\mathbf{p}) = \sum_{i=0}^{\infty} \mathbf{v}_i(t)\,\phi_i(\mathbf{p}), \tag{5.165}$$

where the coefficient functions $\mathbf{v}_i(t)$ are defined by

$$\mathbf{v}_i(t) =< \mathbf{x}(t, \mathbf{p}), \phi_i(\mathbf{p}) > . \tag{5.166}$$

Continuity/smoothness follow from the solution $\mathbf{x}(t, \mathbf{p})$ and similarly the construction of expected values and variances.

A finite approximation $\mathbf{x}^m(t, \mathbf{p})$ to $\mathbf{x}(t, \mathbf{p})$ is defined by

$$\mathbf{x}^m(t, \mathbf{p}) = \sum_{i=0}^{m} \mathbf{v}_i(t)\ \phi_i(\mathbf{p}). \tag{5.167}$$

For long time range integration m may have to be chosen larger than for short time ranges. Further below we will describe how the coefficient functions $\mathbf{v}_i(t)$ can be efficiently approximated.

For functions $\mathbf{x}(t, \mathbf{p})$ that depend smoothly on $\mathbf{p}$ convergence rates for $||\mathbf{x}(t, .) - \mathbf{x}^m(t, .)||$, in the norm associated with (5.162), are known. For instance, for one-dimensional functions $x(p)$ that depend on a scalar parameter p such that $x^{(1)}, \ldots, x^{(k)}$ are continuous (i.e., derivatives w.r.t. p), one has

$$||x(.) - x_H^m(.)||_{\mathrm{L}^2_\rho} \leq C \frac{1}{m^{k/2}}\ ||x^{(k)}(.)||_{\mathrm{L}^2_\rho}, \quad \text{(Hermite expansion [65])}, \tag{5.168}$$

$$||x(.) - x_L^m(.)||_{\mathrm{L}^2_\rho} \leq C \frac{1}{m^k} \sqrt{\sum_{i=0}^{k} ||x^{(i)}(.)||^2_{\mathrm{L}^2_\rho}}, \quad \text{(Legendre expansion [124])}. \tag{5.169}$$

Here the L^2_ρ-norms include the weighting/density function $\rho(.)$. Note that the upperbound in (5.169) actually involves a Sobolev-norm. In [72] one also finds upperbounds using seminorms (that involve less derivatives).

For more general distributions $\rho(.)$ convergence may not be true. For instance, polynomials in a lognormal variable are not dense in L^2_ρ. For convergence one needs to require that the probability measure is uniquely determined by its moments [81]. One at least needs that the expected value of each polynomial has to exist. This has a practical impact. The imperfections in a manufacturing process cause some variability in the components of an electronic circuit. To address the variability, corresponding parameters or functions are replaced by random variables or random fields for uncertainty quantification. However, the statistics of the parameters often do not obey traditional probability distributions like Gaussian, uniform, beta or others. In such a case one may have to construct probability distributions or probability density functions, respectively, which approximate the true statistics at a sufficient accuracy. Thereby, one has to match corresponding data obtained from measurements and observations of electronic devices. The resulting probability distribution functions should be continuous and all moments of the random variables should be finite such that a broad class of methods like, e.g., Polynomial Chaos, is applicable.

The integrals (5.166) can be computed by (quasi) Monte Carlo, or by multi-dimensional quadrature. We assume quadrature grid points $\mathbf{p}^1$, $\mathbf{p}^2, \ldots, \mathbf{p}^K$ and quadrature weights w_k, $1 \leq k \leq K$, such that

$$< \mathbf{x}(t,\mathbf{p}), \phi_i(\mathbf{p}) > \approx \sum_{k=1}^{K} w_k \ \mathbf{x}(t,\mathbf{p}^k) \ \phi_i(\mathbf{p}^k). \tag{5.170}$$

We solve (5.164) for $\mathbf{x}(t,\mathbf{p}^k)$, $k = 1, \ldots, K$ (K deterministic simulations). Here any suitable numerical solver for (5.164) can be used. In fact (5.170) is a (discrete) inner-product with weighting function $w_K(\mathbf{p}) = \sum_{k=1}^{K} w_k \ \delta(\mathbf{p} - \mathbf{p}^k)$. This approach is called Stochastic Collocation [100, 123, 124]. Afterwards we determine

$$\mathbf{v}_i(t) = \sum_{k=0}^{K} w_k \ \mathbf{x}(t,\mathbf{p}^k) \ \phi_i(\mathbf{p}^k), \quad \text{for each } i. \tag{5.171}$$

Here the Polynomial Chaos expansion is just a post-processing step.

Only for low dimensions q, tensor-product grids of Gaussian quadrature are used. Gaussian quadrature points are optimal for accuracy. In higher-dimensional cases ($q > 1$) one prefers sparse grids [123, 124], like the Smolyak algorithm. Sparse grids may have options for refinement. Note that Gaussian points do not offer this refinement. Stroud-3 and Stroud-5 formulas [116] have become popular [122].

An alternative approach to Stochastic Collocation is provided by Stochastic Galerkin. After, inserting an expansion of the solution, in polynomials in $\mathbf{p}$, into the equations one orthogonally projects the residue of the equations to the subspace spanned by these polynomials. By this, one gets one big system of differential equations in which the $\mathbf{v}_i$ are the unknowns [100, 123, 124]. In practice, Stochastic Collocation is much more easily combined with dedicated software for the simulation problem at hand than is the case with Stochastic Galerkin. Theoretically the last approach is more accurate. However, statistics obtained with Stochastic Collocation is very satisfactory.

We note that the expansion $\mathbf{x}(t,\mathbf{p})$, see (5.165), gives full detailed information when varying $\mathbf{p}$. From this the actual (and probably biased) range of solutions can be determined. These can be different from envelope approximations based on mean and variances.

Because of the orthogonality, the mean of $\mathbf{x}(t,\mathbf{p})$ and of $\mathbf{x}^m(t,\mathbf{p})$ are equal and are given by

$$\mathbf{E}_p[\mathbf{x}(t,\mathbf{p})] = \int_Q \mathbf{x}(t,\mathbf{p})\rho(\mathbf{p}) \, d\mathbf{p} = \mathbf{v}_0(t) = \int_Q \mathbf{x}^m(t,\mathbf{p})\rho(\mathbf{p}) \, d\mathbf{p}. \tag{5.172}$$

Using (5.171), we get an approximative value. The integrals in (5.172) involve all p_k together. One may want to consider effects of p_i and p_j separately. This restricts the parameter space $\mathbb{R}^q$ to a one-dimensional subset with individual distribution densities $\rho_i(p)$ and $\rho_j(p)$.

A covariance function of $\mathbf{x}(t,\mathbf{p})$ can also be easily expressed

$$\begin{aligned}
R_{\mathbf{xx}}(t_1,t_2) &= \mathbf{E}_p[(\mathbf{x}(t_1,\mathbf{p}) - \mathbf{E}_p[\mathbf{x}(t_1,\mathbf{p})])^T\,(\mathbf{x}(t_2,\mathbf{p}) - \mathbf{E}_p[\mathbf{x}(t_2,\mathbf{p})])]\\
&= \int_Q (\mathbf{x}(t_1,\mathbf{p}) - \mathbf{E}_p[\mathbf{x}(t_1,\mathbf{p})])^T\,(\mathbf{x}(t_2,\mathbf{p}) - \mathbf{E}_p[\mathbf{x}(t_2,\mathbf{p})])\rho(\mathbf{p})\,d\mathbf{p}\\
&\approx\, <(\mathbf{x}^m(t_1,\mathbf{p}) - \mathbf{E}_p[\mathbf{x}^m(t_1,\mathbf{p})])^T\,(\mathbf{x}^m(t_2,\mathbf{p}) - \mathbf{E}_p[\mathbf{x}^m(t_2,\mathbf{p})])>\\
&= \,<(\sum_{i=1}^m \mathbf{v}_i^T(t_1)\phi_i(\mathbf{p}))\,(\sum_{j=1}^m \mathbf{v}_j(t_2)\phi_j(\mathbf{p}))>\\
&= \sum_{i=1}^m \mathbf{v}_i^T(t_1)\mathbf{v}_i(t_2).
\end{aligned} \tag{5.173}$$

This outcome clearly depends on m. A (scalar) variance is given by

$$\mathrm{Var}_p[\mathbf{x}(t,\mathbf{p})] = R_{\mathbf{xx}}(t,t) \approx \sum_{i=1}^m \mathbf{v}_i^T(t)\mathbf{v}_i(t) = \sum_{i=1}^m ||\mathbf{v}_i(t)||^2 = ||\mathbf{V}_0(t)||^2, \tag{5.174}$$

where $\mathbf{V}_0^T(t) = (\mathbf{0}^T, \mathbf{v}_1^T(t), \ldots, \mathbf{v}_m^T(t))^T$. Note that this equals

$$\mathrm{Var}_p[\mathbf{x}(t,\mathbf{p})] \approx \sum_{i=1}^m \sum_{d=1}^q v_{i,q}^2(t) = \sum_{d=1}^q \sum_{i=1}^m v_{i,q}^2(t) = \sum_{d=1}^q \mathrm{Var}_p[x_d(t,\mathbf{p})]. \tag{5.175}$$

Having a gPC expansion the sensitivity (matrix) w.r.t. $\mathbf{p}$ is easily obtained

$$\mathbf{S}_p(t,\mathbf{p}) = \left[\frac{\partial \mathbf{x}(t,\mathbf{p})}{\partial \mathbf{p}}\right] \approx \sum_{i=0}^m \mathbf{v}_i(t)\,\frac{\partial \phi_i(\mathbf{p})}{\partial \mathbf{p}}. \tag{5.176}$$

One may restrict this to $\mathbf{S}_p(t,\mu_p)$, where $\mu_p = \mathrm{E}[\mathbf{p}]$ and $\frac{\partial \mathbf{x}(t,\mathbf{p})}{\partial \mathbf{p}}$ is the solution of the system that is differentiated w.r.t. $\mathbf{p}$ at $\mathbf{p} = \mu_p$. For a scalar quantity x one can order according to a ‘stochastic influence’ based on

$$\max\{\frac{\partial x}{\partial p_1}\,\sigma_{p_1}, \ldots, \frac{\partial x}{\partial p_q}\,\sigma_{p_q}\}. \tag{5.177}$$

Here $\sigma_{p_i}^2 = \mathrm{Var}[p_i]$. The sensitivity matrix also is subject to stochastic variations. With a gPC expansion one can determine a mean global sensitivity matrix by

$$\mathbf{S}_p(t) = \mathbf{E}_p \left[\frac{\partial \mathbf{x}(t, \mathbf{p})}{\partial \mathbf{p}} \right] \approx \sum_{i=0}^{m} \mathbf{v}_i(t) \int_Q \frac{\partial \phi_i(\mathbf{p})}{\partial \mathbf{p}} \rho(\mathbf{p})\, d\mathbf{p}. \tag{5.178}$$

Note that the integrals at the right-hand side can be determined in advance and stored in tables.

In [85] (see also [84]) a parameterized system in the frequency domain

$$[s\mathbf{C}(\mathbf{p}) + \mathbf{G}(\mathbf{p})]\mathbf{x}(s, \mathbf{p}) = \mathbf{B}\mathbf{u}(s), \tag{5.179}$$

$$\mathbf{y}(s, \mathbf{p}) = \mathbf{B}^T \mathbf{x}(s, \mathbf{p}). \tag{5.180}$$

is considered. Here s is the (angular) frequency. For this system a parameterized MOR approach is proposed, which exploits an expansion of $\mathbf{C}(\mathbf{p})$ and $\mathbf{G}(\mathbf{p})$

$$\mathbf{C}(\mathbf{p}) = \sum_{l_1 \ldots l_q = 0 \ldots 0}^{k_1 \ldots k_q} \Phi_{l_1 \ldots l_q}(\mathbf{p}) \mathbf{C}_{l_1 \ldots l_q}, \tag{5.181}$$

$$\mathbf{G}(\mathbf{p}) = \sum_{l_1 \ldots l_q = 0 \ldots 0}^{k_1 \ldots k_q} \Phi_{l_1 \ldots l_q}(\mathbf{p}) \mathbf{G}_{l_1 \ldots l_q}, \tag{5.182}$$

$$\Phi_{l_1 \ldots l_q}(\mathbf{p}) = p_1^{l_1} p_2^{l_2} \ldots p_q^{l_q}. \tag{5.183}$$

In [71] the parameter variation in $\mathbf{C}$ and $\mathbf{G}$ did come from parameterized layout extraction of RC circuits.

In Algorithm 5.2 it is assumed that a set $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^K$ is given in advance, together with frequencies $s_1, s_2, \ldots, s_K$. Let $\Psi^k = (s_k, \mathbf{p}^k)$. Furthermore, let $\mathbf{A} = s\mathbf{C}(\mathbf{p}) + \mathbf{G}(\mathbf{p})$ and $\mathbf{A}\mathbf{X} = \mathbf{B}$, and, similarly, $\mathbf{A}_k = \mathbf{A}(\Psi^k) = s_k \mathbf{C}(\mathbf{p}^k) + \mathbf{G}(\mathbf{p}^k)$ and $\mathbf{A}_k \mathbf{X}_k = \mathbf{B}$.

A projection matrix $\mathbf{V}$ (with orthonormal columns $\mathbf{v}_i$) is determined such that $\mathbf{X}(s, \mathbf{p}) \approx \bar{\mathbf{X}}(s, \mathbf{p}) \equiv \mathbf{V}\hat{\mathbf{X}}(s, \mathbf{p}) \equiv \sum_{i=1}^{K'} \alpha_i(s, \mathbf{p})\mathbf{v}_i$. Algorithm 5.2 applies a strategy of which a key step is found in [85]. The extension of $\mathbf{V}$ is similar to the recycling of Krylov subspaces [102] and used in MOR by [84]. The refinement introduced in [85] is in the selection from the remaining set (steps 5–6). Note that the residues deal with $\mathbf{B}$ and with $\mathbf{x}$ and not with the effect in $\mathbf{y}$. Hence, one may consider a two-sided projection here. The method of [85] was used in [71] (using expansions of the matrices in moments of $\mathbf{p}$; note that used expressions from layout extraction were linear in $\mathbf{p}$).

Algorithm 5.2 pMOR Strategy in Uncertainty Quantification

```
1: A set $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^K$ is given in advance, together with frequencies $s_1, s_2, \ldots, s_K$. In our case the $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^K$ can come from quadrature points in Stochastic Collocation. Let $\Psi^k = (s_k, \mathbf{p}^k)$. Furthermore, let $\mathbf{A} = s\mathbf{C}(\mathbf{p}) + \mathbf{G}(\mathbf{p})$ and $\mathbf{AX} = \mathbf{B}$, and, similarly, $\mathbf{A}_k = \mathbf{A}(\Psi^k) = s_k\mathbf{C}(\mathbf{p}^k) + \mathbf{G}(\mathbf{p}^k)$ and $\mathbf{A}_k\mathbf{X}_k = \mathbf{B}$.
2: Assume that we have already found some part of the (orthonormal) basis, $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_k)$
3: For any $\Psi^j$, that was not selected before to extend the basis, the actual error formally is given by $\mathbf{E}^j = \mathbf{X}(\Psi^j) - \sum_{i=1}^k \alpha_i(\Psi^j)\mathbf{v}_i$ and thus for the residue we have $\mathbf{R}^j = \mathbf{A}_j\mathbf{E}^j = \mathbf{B} - \sum_{i=1}^k \alpha_i(\Psi^j)\mathbf{A}_j\mathbf{v}_i$. In [85] one determines $\mathbf{R} = \mathbf{B} \perp \mathrm{Span}(\mathbf{A}_j\mathbf{V})$, the residue after orthogonalization of $\mathbf{B}$ against $\mathrm{Span}(\mathbf{A}_j\mathbf{V})$. This step does not require evaluation of a solution.
4: Let $\mathbf{R} = (\mathbf{R}_1, \ldots, \mathbf{R}_m)$, $r_j = \sum_{i=1}^m ||\mathbf{R}_i||$ and determine $j_0$ such that $r_{j_0} = \max_j r_j$.
5: if $(r_{j_0} > \varepsilon)$ then
6:     $\mathbf{X}(\Psi_{j_0})$ may add most significantly rank to the space spanned by $\mathbf{V}$. Hence one now really evaluates $\mathbf{X}_{j_0} = \mathbf{X}(\Psi_{j_0})$ and orthogonalizes this against $\mathbf{V}$ and extends $\mathbf{V}$ with this orthogonal complement. Thus $\mathbf{X}_{j_0} = \mathbf{X}(\Psi_{j_0}) = [\mathbf{A}_k]^{-1}\mathbf{B} = [\mathbf{A}(\Psi^k)]^{-1}\mathbf{B}$ and $\mathbf{V}_k = \mathbf{X}_{j_0} - \mathbf{V}(\mathbf{V}^T\mathbf{X}_{j_0})$ is the expansion to $\mathbf{V}$. One can use a rank-revealing QR for this step (which also includes a tolerance). Note that until now one collects only zero-moments (in the frequency expansion); for refinements see remarks at the end of this Section.
7:     Reduce the set of the $\Psi^k$ with $\Psi^{j_0}$. Go to Step 2.
8: else
9:     Decide for applying MOR on remainder.
10:    if (MOR) then
11:        if (Expressions for $\mathbf{C}(\mathbf{p})$ and $\mathbf{G}(\mathbf{p})$ are explicitly known) then
12:            Expand the matrices $\mathbf{C}(\mathbf{p})$ and $\mathbf{G}(\mathbf{p})$ in polynomials as in (5.181)–(5.182)
13:            Apply the common projection matrix to get the reduced parameterized system.
14:            Apply the collocation to the reduced system (and possibly re-evaluate for parameters used so far the solutions of the reduced system). The solutions of the reduced system at the re-evaluated parameters may be compared to the solutions of the non-reduced system to provide some error control. Note that the expanded expressions provide expressions for the reduced system.
15:            for all $\Psi^k$ do
16:                Evaluate $\mathbf{C}(\mathbf{p}^k)$ and $\mathbf{G}(\mathbf{p}^k)$ of the reduced system.
17:                Solve the reduced system.
18:            end for
19:            One now has a parameterized reduced system.
20:        else
21:            for all $\Psi^k$ do
22:                Evaluate $\mathbf{C}(\mathbf{p}^k)$ and $\mathbf{G}(\mathbf{p}^k)$ of the big system (in the CAD environment, say).
23:                Apply the common projection matrix to get the reduced system.
24:                Solve the reduced system.
25:            end for
26:        end if
27:        Determine the gPC-expansion of the solution of the reduced system.
28:        Perform statistics and/or determine sensitivity of the solution of the reduced system.
29:    else
30:        Use the Krylov space found so far to efficiently solve all remaining solutions $\mathbf{X}(\Psi_j)$. Note that we can use the original expressions in (5.179).
31:        Determine the gPC-expansion of the solution of the original system (5.179).
32:        Perform statistics and/or determine sensitivity of the solution of the original system.
33:    end if
34: end if
```

This procedure assumes that the evaluation of a matrix $\mathbf{A}_k$ (and subsequent matrix vector multiplications) is much cheaper than determining a solution $\mathbf{X}(\Psi_k)$. Note also that after extending the basis $\mathbf{V}$ in the next step the norms of the residues should reduce. This allows for some further efficiency in the algorithm [85]. Finally, we remark that the $\mathbf{X}_k$ are zero order (block) moments at Ψ^k. After determining the LU-decomposition of $\mathbf{A}_k$ one easily includes higher moments as well when extending the basis.

A main conclusion of this section is that for the Stochastic Collocation the expansions (5.181)–(5.183) are not explicitly needed by the algorithm. This facilitates dealing with parameters that come from geometry, like scaling [111–115]. The evaluation can completely be done within the CAD environment of the simulation tool – in which case the expressions remain hidden.

The selection of the next parameter introduces a notion of "dominancy" from an algorithmic point of view: this parameter most significantly needs extension of the Krylov subspace. To invest for this parameter will automatically reduce work for other parameters (several may even drop out of the list because of small residues).

If first order sensitivity matrices are available, like in $\mathbf{C}(\mathbf{p}) = \mathbf{C}_0(\mathbf{p}_0) + \mathbf{C}'(\mathbf{p}_0)\mathbf{p}$ and in $\mathbf{G}(\mathbf{p}) = \mathbf{G}_0(\mathbf{p}_0) + \mathbf{G}'(\mathbf{p}_0)\mathbf{p}$ one can apply a Generalized Singular Value Decomposition [89] to both pairs $(\mathbf{C}_0^T(\mathbf{p}_0), [\mathbf{C}']^T(\mathbf{p}_0))$ and $(\mathbf{G}_0^T(\mathbf{p}_0), [\mathbf{G}']^T(\mathbf{p}_0))$. In [101] this was applied in MOR for linear coupled systems. The low-rank approximations for $\mathbf{C}'(\mathbf{p}_0)$ and $\mathbf{G}'(\mathbf{p}_0)$ (obtained by a Generalized SVD [89]) give way to increase the basis for the columns of $\mathbf{B}$ of the source function. Note that by this one automatically will need MOR methods that can deal with many terminals [68, 97, 120].

In Algorithm 5.2 and in [85] the subspace generated by the basis $\mathbf{V}$ is slightly increasing with each new $\mathbf{p}_k$. A different approach is to apply normal MOR for each $\mathbf{p}_k$, giving bases $\mathbf{V}_k$, and next determine $\mathbf{V}$ by an SVD or rank-revealing QR-factorization of $[\mathbf{V}_k, \ldots, \mathbf{V}_K]$. In [66] this approach is used to obtain a Piecewise $\mathscr{H}_2$-Optimal Interpolation pMOR Algorithm.

To efficiently apply parameterized MOR in Uncertainty Quantification is described in [104, 119]. In [105, 106] sensitivity analysis of the variance did provide ways to identify dominant parameters that contribute most to the variance of a quantity of interest. This approach is different from the low-rank approximations (using the Generalized SVD), mentioned above.

5.4 MOR for Singularly Perturbed Systems

For large systems of ordinary differential equations (ODEs), efficient MOR methods already exist in the linear case, see [125].[4] We want to generalize according techniques to the case of differential-algebraic equations (DAEs). On the one hand,

[4] Section 5.4 has been written by: Kasra Mohaghegh, Roland Pulch and E. Jan W. ter Maten. For an extended version we refer to the Ph.D.-Thesis [135] of the first author and to the papers [136, 137].

a high-index DAE problem can be converted into a lower-index system by analytic differentiations, see [127]. A transformation to index zero yields an equivalent system of ODEs. On the other hand, a regularization is directly feasible in case of semi-explicit systems of DAEs. Thereby, we obtain a singularly perturbed problem of ODEs with an artificial parameter. Thus according MOR techniques can be applied to the ODE system. An MOR approach for DAEs is achieved by considering the limit to zero of the artificial parameter.

We consider a simplified, semi-explicit DAE system to illustrate some concepts only

$$\begin{aligned} \dot{\mathbf{y}}(t) &= \mathbf{f}(\mathbf{y}(t), \mathbf{z}(t)), \qquad & \mathbf{y} : \mathbb{R} \to \mathbb{R}^k, \\ \mathbf{0} &= \mathbf{g}(\mathbf{y}(t), \mathbf{z}(t)), \qquad & \mathbf{z} : \mathbb{R} \to \mathbb{R}^l, \end{aligned} \tag{5.184}$$

with differential and perturbation index 1 or 2. For the construction of numerical methods to solve initial value problems of (5.184), a direct as well as an indirect approach can be used. The direct approach applies an *ε-embedding* of the DAEs (5.184), i.e., the system changes into

$$\begin{aligned} \dot{\mathbf{y}}(t) &= \mathbf{f}(\mathbf{y}(t), \mathbf{z}(t)) \\ \varepsilon\dot{\mathbf{z}}(t) &= \mathbf{g}(\mathbf{y}(t), \mathbf{z}(t)) \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \dot{\mathbf{y}}(t) &= \mathbf{f}(\mathbf{y}(t), \mathbf{z}(t)) \\ \dot{\mathbf{z}}(t) &= \tfrac{1}{\varepsilon}\mathbf{g}(\mathbf{y}(t), \mathbf{z}(t)) \end{aligned} \tag{5.185}$$

with a real parameter $\varepsilon \neq 0$. Techniques for ODEs can be employed for the singularly perturbed system (5.185). The limit $\varepsilon \to 0$ yields an approach for solving the DAEs (5.184). The applicability and quality of the resulting method still has to be investigated.

Alternatively, the indirect approach is based on the *state space form* of the DAEs (5.184) with differential and perturbation index 1 or 2, for nonlinear cases see [139], i.e.,

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t), \Phi(\mathbf{y}(t))) \tag{5.186}$$

with $\mathbf{z}(t) = \Phi(\mathbf{y}(t))$. To evaluate the function Φ, the nonlinear system

$$\mathbf{g}(\mathbf{y}(t), \Phi(\mathbf{y}(t))) = 0 \tag{5.187}$$

is solved for given value $\mathbf{y}(t)$. Consequently, the system (5.186) represents ODEs for the differential variables y and ODE methods can be applied. In each evaluation of the right-hand side in (5.186), a nonlinear system (5.187) has to be solved. More details on techniques based on the ε-embedding and the state space form can be found in [132].

Although some MOR methods for DAEs already exist, several techniques are restricted to ODEs or exhibit better properties in the ODE case in comparison to the DAE case. The direct or the indirect approach enables the usage of MOR schemes for ODEs (5.185) or (5.186), where an approximation with respect to the original DAEs (5.184) follows. The aim is to obtain suggestions for MOR schemes via these strategies, where the quality of the resulting approximations still has to be analyzed in each method.

In this section, we focus on the direct approach for semi-explicit system of DAEs, i.e., the ε-embedding (5.185) is considered. MOR methods are applied to the singularly perturbed system (5.185). Two scenarios exist to achieve an approximation of the behavior of the original DAEs (5.184) by MOR. Firstly, an MOR scheme can be applied to the system (5.185) using a constant $\varepsilon \neq 0$, which is chosen sufficiently small (on a case by case basis) such that a good approximation is obtained. Secondly, a parametric or parameterized Model Order Reduction (pMOR) method yields a reduced description of the system of ODEs, where the parameter ε still represents an independent variable. Hence the limit $\varepsilon \to 0$ causes an approach for an approximation of the original DAEs.

We investigate the two approaches with respect to MOR methods based on an approximation of the transfer function, which describes the input-output behavior of the system in frequency domain.

5.4.1 Model Order Reduction and ε-Embedding

We restrict ourselves to semi-explicit DAE systems of the type (5.188)–(5.189) and introduce $w(t)$ as an output instead of $y(t)$ with exact the same condition. According to (5.184), after linearizing, we can write the system as

$$\mathbf{C}\dot{\mathbf{x}} = -\mathbf{G}\mathbf{x} + \mathbf{B}\mathbf{u}(t), \tag{5.188}$$

$$\mathbf{w}(t) = \mathbf{L}\mathbf{x}(t). \tag{5.189}$$

The solution $\mathbf{x}$ and the matrix $\mathbf{C}$ exhibit the partitioning:

$$\mathbf{x} = \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}, \qquad \mathbf{C} = \begin{pmatrix} \mathbf{I}_{k\times k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{l\times l} \end{pmatrix}.$$

$\mathbf{w}(t)$ is the output of the system. The order of the system is $n = k + l$, where k and l are the dimensions of the differential part and the algebraic part (constraints), respectively, defined in the semi-explicit system (5.184). $\mathbf{B} \in \mathbb{R}^{n\times m}$; $\mathbf{L} \in \mathbb{R}^{p\times n}$. After taking the Laplace transform, the corresponding $p \times m$ matrix-valued rational transfer function is

$$\mathbf{H}(s) = \mathbf{L} \cdot (\mathbf{G} + s\mathbf{C})^{-1} \cdot \mathbf{B} = \mathbf{L} \cdot \left(\mathbf{G} + s \begin{pmatrix} \mathbf{I}_{k\times k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{l\times l} \end{pmatrix}\right)^{-1} \cdot \mathbf{B},$$

provided that $\det(\mathbf{G} + s\mathbf{C}) \neq 0$ and $\mathbf{x}(0) = \mathbf{0}$ and $\mathbf{u}(0) = \mathbf{0}$. Following the direct approach [135], the ε-embedding changes the system (5.188)–(5.189) into:

$$\begin{cases} \mathbf{C}(\varepsilon)\frac{d\mathbf{x}(t)}{dt} = -\mathbf{G}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), & \mathbf{x}(0) = \mathbf{x}_0, \\ \quad \mathbf{w}(t) = \mathbf{L}\mathbf{x}(t), \end{cases} \tag{5.190}$$

where

$$\mathbf{C}(\varepsilon) = \begin{pmatrix} \mathbf{I}_{k\times k} & \mathbf{0} \\ \mathbf{0} & \varepsilon \mathbf{I}_{l\times l} \end{pmatrix} \qquad \text{for } \varepsilon \in \mathbb{R}$$

with the same inner state and input/output as before. For $\varepsilon \neq 0$, the matrix $\mathbf{C}(\varepsilon)$ is regular in (5.190) and the transfer function reads:

$$\mathbf{H}_\varepsilon(s) = L \cdot (\mathbf{G} + s \cdot \mathbf{C}(\varepsilon))^{-1} \cdot \mathbf{B}$$

provided that $\det(\mathbf{G} + s\mathbf{C}(\varepsilon)) \neq 0$. For convenience, we introduce the notation

$$\mathbf{M}(s, \varepsilon) := s\mathbf{C}(\varepsilon) = s \begin{pmatrix} \mathbf{I}_{k\times k} & \mathbf{0} \\ \mathbf{0} & \varepsilon \mathbf{I}_{l\times l} \end{pmatrix}.$$

It holds $\mathbf{M}(s, 0) = s\mathbf{C}$ with $\mathbf{C}$ from (5.188).

Concerning the relation between the original system (5.188)–(5.189) and the regularized system (5.190) with respect to the transfer function, we achieve the following statement. Without loss of generality, the induced matrix norm of the Euclidean vector norm is applied.

Lemma 5.1 *Let* $\mathbf{A},\ \tilde{\mathbf{A}} \in \mathbb{R}^{n\times n}$, $\det(\mathbf{A}) \neq 0$ *and* $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 = \|\Delta\mathbf{A}\|_2$ *where* $\Delta\mathbf{A}$ *is small enough. Then it holds:*

$$\|\mathbf{A}^{-1} - \tilde{\mathbf{A}}^{-1}\|_2 \leq \frac{\|\mathbf{A}^{-1}\|_2^2 \cdot \|\Delta\mathbf{A}\|_2}{1 - \|\mathbf{A}^{-1}\|_2 \cdot \|\Delta\mathbf{A}\|_2}.$$

Proof It holds

$$\|\mathbf{A}^{-1} - \tilde{\mathbf{A}}^{-1}\|_2 = \max_{\|\mathbf{x}\|_2=1} \left\| \mathbf{A}^{-1}x - \tilde{\mathbf{A}}^{-1}x \right\|_2.$$

Suppose $\mathbf{y} := \mathbf{A}^{-1}\mathbf{x},\ \tilde{\mathbf{y}} := \tilde{\mathbf{A}}^{-1}\mathbf{x}$, then the sensitivity analysis of linear systems yields

$$\frac{\|\Delta\mathbf{y}\|_2}{\|\mathbf{y}\|_2} \leq \frac{\kappa(\mathbf{A})}{1 - \kappa(\mathbf{A})\frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}} \left(\frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2} + \underbrace{\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2}}_{=0} \right),$$

where the quantity

$$\kappa(\mathbf{A}) \equiv \left\|\mathbf{A}^{-1}\right\|_2 \|\mathbf{A}\|_2$$

is the relative *condition number*. So by substituting the value of $\kappa(\mathbf{A})$ we have:

$$\|\mathbf{y}-\tilde{\mathbf{y}}\|_2 \leq \frac{\left\|\mathbf{A}^{-1}\right\|_2 \cdot \|\Delta\mathbf{A}\|_2 \cdot \left\|\mathbf{A}^{-1}\right\|_2 \|\mathbf{x}\|_2}{1-\|\mathbf{A}^{-1}\|_2 \cdot \|\Delta\mathbf{A}\|_2}$$

then

$$\|\mathbf{A}^{-1}-\tilde{\mathbf{A}}^{-1}\|_2 \leq \frac{\|\mathbf{A}^{-1}\|_2^2 \cdot \|\Delta\mathbf{A}\|_2}{1-\|\mathbf{A}^{-1}\|_2 \cdot \|\Delta\mathbf{A}\|_2}.$$

□

We conclude from Lemma 5.1 that

$$\lim_{\Delta\mathbf{A}\to 0} \tilde{\mathbf{A}}^{-1} = \mathbf{A}^{-1},$$

for example.

Theorem 5.1 *For fixed $s \in \mathbb{C}$ with* $\det(\mathbf{G}+\mathbf{M}(s,0)) \neq 0$ *and $\varepsilon \in \mathbb{R}$ satisfying*

$$|s| \cdot |\varepsilon| \leq \frac{c}{\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\|_2} \tag{5.191}$$

for some $c \in (0,1)$, the transfer functions $\mathbf{H}(s)$ and $\mathbf{H}_\varepsilon(s)$ of the systems (5.188)–(5.189) *and* (5.190) *exist and it holds*

$$\|\mathbf{H}(s)-\mathbf{H}_\varepsilon(s)\|_2 \leq \|\mathbf{L}\|_2 \cdot \|\mathbf{B}\|_2 \cdot K(s) \cdot |s| \cdot |\varepsilon|$$

with

$$K(s) = \frac{1}{1-c} \left\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\right\|_2^2.$$

Proof Let $\mathbf{A} = \mathbf{G}+\mathbf{M}(s,0)$ and $\tilde{\mathbf{A}} = \mathbf{G}+\mathbf{M}(s,\varepsilon)$. The condition (5.191) guarantees that the matrices $\tilde{\mathbf{A}}$ are regular. The definition of the transfer functions implies:

$$\|\mathbf{H}(s)-\mathbf{H}_\varepsilon(s)\|_2 \leq \|L\|_2 \cdot \left\|\mathbf{A}^{-1}-\tilde{\mathbf{A}}^{-1}\right\|_2 \cdot \|\mathbf{B}\|_2.$$

We obtain:

$$\left\|\mathbf{A}-\tilde{\mathbf{A}}\right\|_2 = \|\mathbf{M}(s,0)-\mathbf{M}(s,\varepsilon)\|_2 = |s| \cdot \left\|\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \varepsilon\mathbf{I}_{l\times l} \end{pmatrix}\right\|_2 = |s| \cdot |\varepsilon|.$$

Applying the Lemma 5.1, the term at the right-hand side of the expression above becomes:

$$\begin{aligned}\left\|\mathbf{A}^{-1}-\tilde{\mathbf{A}}^{-1}\right\|_2 &\le \frac{\|\mathbf{A}^{-1}\|_2^2\cdot\|\mathbf{M}(s,0)-\mathbf{M}(s,\varepsilon)\|_2}{1-\|\mathbf{A}^{-1}\|_2\cdot\|\mathbf{M}(s,0)-\mathbf{M}(s,\varepsilon)\|_2}\\ &\le \frac{1}{1-c}\left\|\mathbf{A}^{-1}\right\|_2^2\cdot\|\mathbf{M}(s,0)-\mathbf{M}(s,\varepsilon)\|_2\\ &\le K(s)\,\|\mathbf{M}(s,0)-\mathbf{M}(s,\varepsilon)\|_2\,.\end{aligned}$$

Thus the proof is completed. □

It is clear that for inequality (5.191) we have:

$$\begin{aligned}s\neq 0\in\mathbb{C}: |\varepsilon| &\le \frac{c}{|s|\cdot\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\|_2}\\ s=0\in\mathbb{C}: &\qquad \varepsilon\text{ arbitrary}\end{aligned}$$

We conclude from Theorem 5.1 that

$$\lim_{\varepsilon\to 0}\mathbf{H}_\varepsilon(s)=\mathbf{H}(s)$$

for each $s\in\mathbb{C}$ with $\mathbf{G}+s\mathbf{C}$ regular. The relation (5.191) gives feasible domains of ε

$$\begin{aligned}|s|\le 1: |\varepsilon| &\le \frac{c}{\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\|_2},\\ |s|>1: |\varepsilon| &\le \frac{c}{|s|\cdot\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\|_2}.\end{aligned}$$

We also obtain the uniform convergence

$$\|\mathbf{H}(s)-\mathbf{H}_\varepsilon(s)\|_2\le\hat{K}\,|\varepsilon|\quad\text{for all }\ s\in S$$

in a compact domain $S\subset\mathbb{C}$ and $\varepsilon\le\delta$ with:

$$\begin{aligned}\delta &= c\cdot\min_{s\in S}\frac{1}{\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\|_2}\quad\text{for }\ \tilde{S}=\emptyset,\\ \delta &= c\cdot\left[\min_{s\in S}\frac{1}{\|(\mathbf{G}+\mathbf{M}(s,0))^{-1}\|_2}\right]\cdot\underbrace{\left[\min_{s\in\tilde{S}}\frac{1}{|s|}\right]}_{\le 1}\quad\text{for }\ \tilde{S}\neq\emptyset,\end{aligned}$$

with $\tilde{S}:=\{z\in S: |z|\ge 1\}$. Furthermore, Theorem 5.1 implies the property

$$\lim_{s\to 0}\mathbf{H}(s)-\mathbf{H}_\varepsilon(s)=0$$

for fixed ε assuming $\det \mathbf{G} \neq 0$. However, we are not interested in the limit case of small variables s.

For reducing the DAE system (5.188)–(5.189), we have two ways to handle the artificial parameter ε, which results in two different scenarios. In the first scenario, we fix a small value of the parameter ε. Thus we use one of the standard techniques for the reduction of the corresponding ODE system. Finally, we achieve a reduced ODE (with small ε inside). The ODE system with small ε represents a regularized DAE. Any reduction scheme for ODEs is feasible. Recent research shows that the Poor Man's TBR (PMTBR), see [138], can be applied efficiently to the ODE case. Figure 5.37 indicates the steps for the first scenario.

In the second scenario, the parameter ε is considered as an independent variable (value not predetermined). We can use the parametric MOR for reducing the corresponding ODE system. The applied parametric MOR is based on [128, 129] in this case. The limit $\varepsilon \to 0$ yields the results in an approximation of original DAEs (5.188)–(5.189). The existence of the approximation in this limit still has to be analyzed. Figure 5.38 illustrates the strategy for the second scenario.

Fig. 5.37 The approach of the ε-embedding for MOR in the first scenario

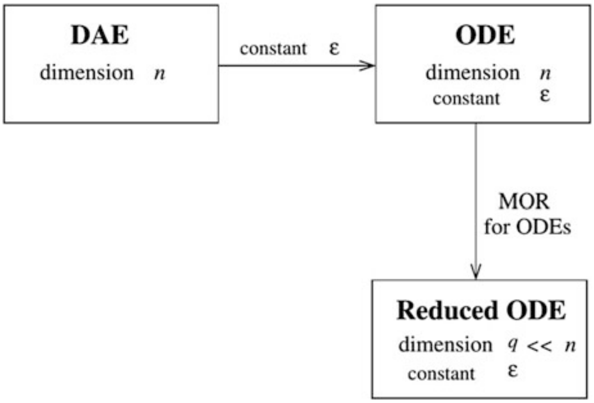
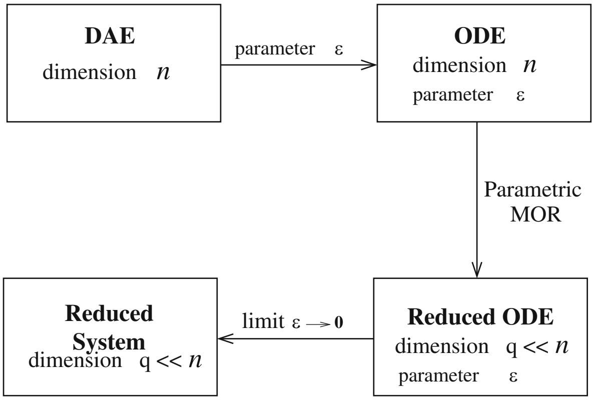


Fig. 5.38 The approach of the ε-embedding for MOR in the second scenario



Theorem 5.1 provides the theoretical background for the both scenarios. We apply an MOR scheme based on an approximation of the transfer function to the system of ODEs (5.190). Let $\tilde{\mathbf{H}}_\varepsilon(s)$ be a corresponding approximation of $\mathbf{H}_\varepsilon(s)$.

It follows

$$\|\mathbf{H}(s) - \tilde{\mathbf{H}}_\varepsilon(s)\|_2 \leq \|\mathbf{H}(s) - \mathbf{H}_\varepsilon(s)\|_2 + \|\mathbf{H}_\varepsilon(s) - \tilde{\mathbf{H}}_\varepsilon(s)\|_2 \tag{5.192}$$

for each $s \in \mathbb{C}$ with $\det(\mathbf{G} + s\mathbf{C}) \neq 0$. Due to Theorem 5.1, the first term becomes small for sufficiently small parameter ε. However, ε should not be chosen smaller than the machine precision on a computer. The second term depends on the applicability of an efficient MOR method to the ODEs (5.190). Thus $\tilde{\mathbf{H}}_\varepsilon(s)$ can be seen as an approximation of the transfer function $\mathbf{H}(s)$ belonging to the system of DAEs (5.188)–(5.189).

5.4.2 Test Example and Numerical Results

We consider a substitute model of a transmission line (TL), see [130], which consists of N cells. Each cell includes a capacitor, an inductor and two resistors, see Fig. 5.39. This TL model represents a scalable benchmark problem (both in differential part and algebraic part but not separately), because we can select the number N of cells. The used physical parameters are

$$C = 10^{-14}\,\mathrm{F/m},\ L = 10^{-8}\,\mathrm{H},\ R = 0.1\,\Omega/\mathrm{m},\ G = 10\,\mathrm{S/m}.$$

We apply modified nodal analysis, see [131], to the RLC circuit and then the state variables $\mathbf{x} \in \mathbb{R}^{3N+3}$ consist of the voltages at the nodes, the currents traversing the

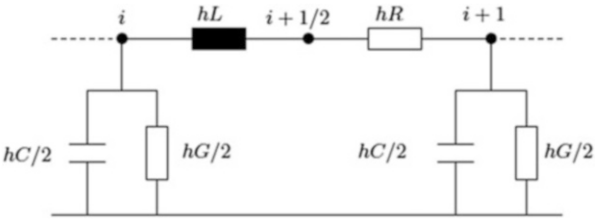


Fig. 5.39 One cell of the RLC transmission line

inductances L and the currents at the boundaries of the circuit:

$$\begin{array}{cc} (V_0, V_1, \ldots, V_N), & (I_{1/2}, I_{3/2}, \ldots, I_{N-1/2}), \\ (V_{1/2}, V_{3/2}, \ldots, V_{N-1/2}), & (I_0, I_N). \end{array}$$

So far we have $3N + 3$ unknowns and only $3N + 1$ equations. Thus two boundary conditions are necessary. Equations for the main nodes and the intermediate nodes in each cell are

$$\begin{array}{l} \frac{h}{2}C\dot{V}_0 + \frac{h}{2}GV_0 + I_{1/2} - I_0 = 0, \\ hC\dot{V}_i + hGV_i + I_{i+1/2} - I_{i-1/2} = 0, \; i = 1, \ldots, N-1, \\ \frac{h}{2}C\dot{V}_N + \frac{h}{2}GV_N + I_N - I_{N-1/2} = 0, \end{array}$$

$$\begin{array}{l} -I_{i+1/2} + \frac{V_{i+1/2} - V_{i+1}}{hR} = 0, \\ hL\dot{I}_{i+1/2} + (V_{i+1/2} - V_i) = 0, \; i = 0, 1, \ldots, N-1, \end{array}$$

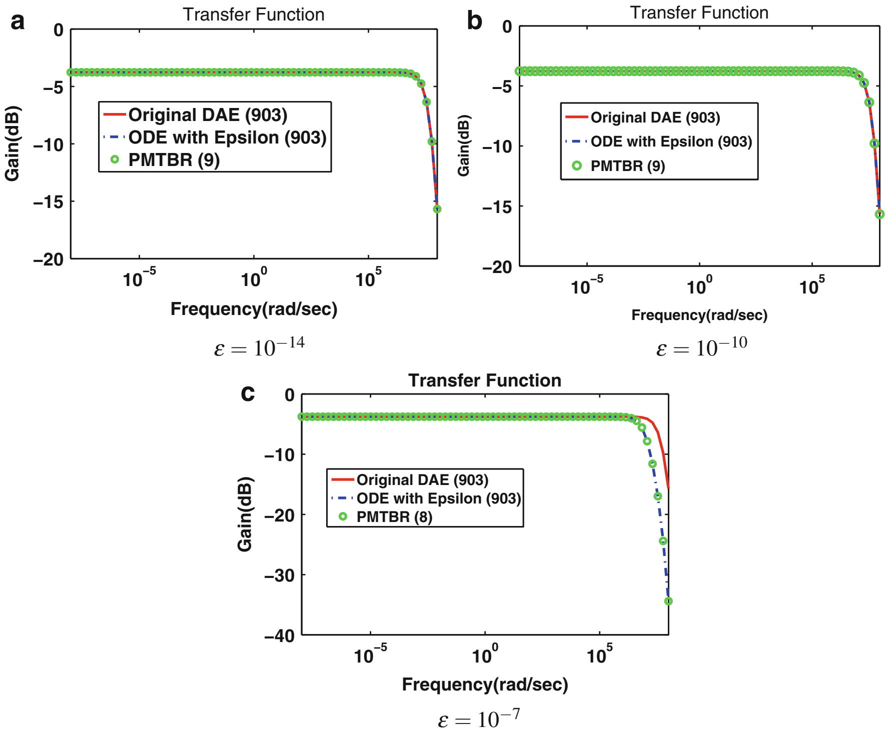
where the variable $h > 0$ represents a discretization step size in space. We apply the boundary conditions

$$\begin{array}{l} I_0 - u(t) = 0, \\ L_1\dot{I}_N + V_N = 0 \end{array}$$

with $L_1 > 0$ and an independent current source u. Now a direct approach (ε-embedding) is used. For the first simulation the variable ε is fixed to 10^{-14} and 10^{-7}, respectively, and the PMTBR method [138] is used as a reduction scheme for the ODE system. For all runs we selected the number of cells to $N = 300$, which results in the order $n = 903$ of the original system of DAEs (5.188)–(5.189). Figure 5.40 shows the transfer function both for the DAE and the ODE (including ε) and the reduced ODE with fixed ε for frequencies $s = i\omega$ with $\omega \in \mathbb{R}$. The number in parentheses shows the order of the systems.

Finally the second scenario with parametric MOR is studied. We apply the PIMTAB parametric MOR following [133, 134]. The limit $\varepsilon \to 0$ gives the result for the reduced DAE.The error plot for the parametric reduction scheme is shown in Fig. 5.41. The error plot shows an overall nice match for the case of $\varepsilon = 0, 10^{-10}$ and as the value for the parameter ε increases, the accuracy of the method and of the reduction algorithm decreases. It is also important to mention that the order of the reduced system in the second scenario is nearly half of the one in the first scenario.

Table 5.4 shows which value for the parameter ε is acceptable for the both scenarios.



$\varepsilon = 10^{-14}$

$\varepsilon = 10^{-10}$

$\varepsilon = 10^{-7}$

Fig. 5.40 Original transfer function for DAE and ODE and reduced transfer function of PMTBR in case of three different parameters ε. The frequency ω ranges from 10^{-8} to 10^{8}

Table 5.4 Acceptance of the method: different values for ε are mentioned. A dash indicates that the error is not calculated; A. and N.A indicate accepted and not accepted, respectively

Value used for ε :	0	10^{-14}	10^{-10}	10^{-7}
Scenario with fixed ε	Same as DAE	A.	A.	N.A.
Scenario with parametric ε	A.	-	A.	N.A.

5.4.3 Conclusions

In this section we applied the ε-embedding to approximate a linear system of DAEs by a system of ODEs. We did consider the transfer function in the frequency domain as a function of ε and proved uniform convergence for frequencies s in a compact region S where the matrix $\mathbf{G} + s\mathbf{C}$ is regular (and thus its inverse uniformly bounded). This motivated the usage of MOR methods for ODEs. Most of the reduction schemes are designed and adopted for linear ODEs. Well-known methods are PMTBR (Poor Man's Truncated Balanced Realization [138]) and the spectral zeros preservation MOR of Antoulas [126].

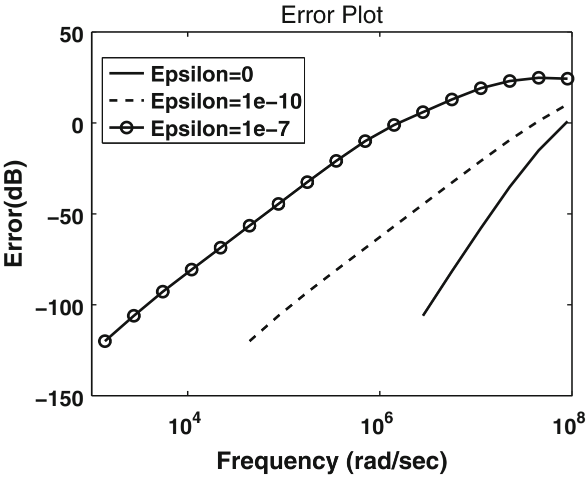


Fig. 5.41 Absolute error plot for the transfer function in the ε-embedding, reduction carried out by parametric MOR with PIMTAB, $\varepsilon = 0, 10^{-10}, 10^{-7}$

In the first scenario we applied a fixed ε and studied for a transmission line model the behavior of the transfer functions of the DAE, of the ODE and of the reduced model obtained with PMTBR for $\varepsilon = 10^{-14},\ 10^{-10},\ 10^{-7}$. Already for the last value the transfer functions between DAE and ODE differ significantly. If we choose bigger values for ε, the system is more friendly but the error is larger and the solution will be changed. On the other hand the transfer function obtained by PMTBR is able to approximate quite well the transfer function of the ODE.

In the second approach we applied the parametric MOR technique PIMTAB [133, 134] to the parameterized ODE. Here we do not need to predefine the value of the ε. We obtain a parameterized MOR that gives a reduced model for $\varepsilon = 0$ for which the transfer function approximates well the one for the DAE [135, 137].

References

References for Section 5.1

1. Achar, R., Nakhla, M.S.: Simulation of high-speed interconnects. Proc. IEEE **89**(5), 693–728 (2001)
2. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)

3. Benner, P., Mehrmann, V., Sorensen, D.C. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin/Heidelberg (2005)
4. Bi, Y.: Effects of paramater variations on integrated circuits. MSc.-Thesis, Delft University of Technology/Technical University Eindhoven, The Netherlands (2007)
5. Bi, Y., van der Kolk, K.-J., Ioan, D., van der Meijs, N.P.: Sensitivity computation of interconnect capacitances with respect to geometric parameters. In: Proceedings of the IEEE International Conference on Electrical Performance of Electronic Packaging (EPEP), San Jose, CA, USA, pp. 209–212 (2008)
6. Ciuprina, G., Ioan, D., Niculae, D., Fernández Villena, J., Silveira, L.M.: Parametric models based on sensitivity analysis for passive components. In: Intelligent Computer Techniques in Applied Electromagnetics. Studies in Computational Intelligence Series, vol. 119, pp. 231–239. Springer, Berlin/Heidelberg (2008)
7. Ciuprina, G., Loan, D., Mihalache, D., Seebacher, E.: Domain partitioning based parametric models for passive on-chip components. In: Roos, J., Costa, L. (eds.) Scientific Computing in Electrical Engineering 2008. Mathematics in Industry, vol. 14, pp. 37–44. Springer, Berlin/Heidelberg (2010)
8. Daniel, L., Siong, O.C., Low, S.C., Lee, K.H., White, J.K.: A multiparameter moment-matching model-reduction approach for generating geometrically parametrized interconnect performance models. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **23**, 678–693 (2004)
9. Davis, T.A.: Direct Methods for Sparse Linear Systems. The Fundamentals of Algorithms. SIAM, Philadelphia (2006)
10. Davis, T.A., Palamadai Natarajan, E.: Algorithm 907: KLU, a direct sparse solver for circuit simulation problems. ACM Trans. Math. Softw. **37**(3), Article 36 (2010)
11. Elfadel, I.M., Ling, D.L.: A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 66–71 (1997)
12. El-Moselhy, T.A., Elfadel, I.M., Daniel, L.: A capacitance solver for incremental variation-aware extraction. In: Proceedings of the IEEE/ACM International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 662–669 (2008)
13. Feldmann, P., Freund, R.W.: Efficient linear circuit analysis by Padé approximation via the Lanczos process. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **14**(5), 639–649 (1995)
14. Fernández Villena, J., Schilders, W.H.A., Silveira, L.M.: Parametric structure-preserving model order reduction. In: IFIP International Conference on Very Large Scale Integration, VLSI – SoC 2007, Atlanta, pp. 31–36 (2007)
15. Freund, R.W.: Sprim: structure-preserving reduced-order interconnect macro-modeling. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD) 2004, San Jose, pp. 80–87 (2004)
16. Gunupudi, P.K., Khazaka, R., Nakhla, M.S., Smy, T., Celo, D.: Passive parameterized time-domain macromodels for high-speed transmission-line networks. IEEE Trans. Microw. Theory Tech. **51**(12), 2347–2354 (2003)
17. Heydari, P., Pedram, M.: Model reduction of variable-geometry interconnects using variational spectrally-weighted balanced truncation. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 586–591 (2001)
18. Jaimoukha, I.M., Kasenally, E.M.: Krylov subspace methods for solving large Lyapunov equations. SIAM J. Numer. Anal. **31**, 227–251 (1994)
19. Kamon, M., Wang, F., White, J.: Generating nearly optimally compact models from Krylov-subspace based reduced-order models. IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. **47**(4), 239–248 (2000)
20. Kula, S.: Reduced order models of interconnects in high frequency integrated circuits. Ph.D.-Thesis, Politehnica University of Bucharest (2009)

21. Li, J.-R., Wang, F., White, J.: Efficient model reduction of interconnect via approximate system Grammians. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 380–383 (1999)
22. Li, P., Liu, F., Li, X., Pileggi, L.T., Nassif, S.R.: Modeling interconnect variability using efficient parametric model order reduction. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE), Munich, pp. 958–963 (2005)
23. Li, X., Li, P., Pileggi, L.: Parameterized interconnect order reduction with explicit-and-implicit multi-parameter moment matching for inter/intra-die variations. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 806–812 (2005)
24. Li, Y., Bai, Z., Su, Y., Zeng, X.: Model order reduction of parameterized interconnect networks via a two-directional Arnoldi process. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(9), 1571–1582 (2008)
25. Liu, Y., Pileggi, L.T., Strojwas, A.J.: Model order reduction of RC(L) interconnect including variational analysis. In: Proceedings of the 36th ACM/IEEE Design Automation Conference (DAC), New Orleans, pp. 201–206 (1999)
26. Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **AC-26**(1), 17–32 (1981)
27. Odabasioglu, A., Celik, M., Pileggi, L.T.: PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **17**(8), 645–654 (1998)
28. Phillips, J.: Variational interconnect analysis via PMTBR. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 872–879 (2004)
29. Phillips, J.R., Silveira, L.M.: Poor Man's TBR: a simple model reduction scheme. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **24**(1), 43–55 (2005)
30. Phillips, J., Daniel, L., Silveira, L.M.: Guaranteed passive balancing transformations for model order reduction. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **22**(8), 1027–1041 (2003)
31. Pillage, L.T., Rohrer, R.A.: Asymptotic waveform evaluation for timing analysis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **9**(4), 352–366 (1990)
32. Saad, Y.: Iterative Methods for Sparse Linear Systems. Pws Publishing Co., Boston (1996)
33. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
34. Silveira, L.M., Kamon, M., Elfadel, I., White, J.K.: A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In: Proceedings of the International Conference on Computer Aided-Design (ICCAD), San Jose, pp. 288–294 (1996)
35. Yang, F., Zeng, X., Su, Y., Zhou, D.: RLCSYN: RLC equivalent circuit synthesis for structure-preserved reduced-order model of interconnect. In: Proceedings of the International Symposium on Circuits and Systems, New Orleans, pp. 2710–2713 (2007)
36. Yu, H., He, L., Tan, S.X.D.: Block structure preserving model order reduction. In: BMAS – IEEE Behavioral Modeling and Simulation Workshop, San Jose, pp. 1–6 (2005)
37. Zhu, Z., Phillips, J.: Random sampling of moment graph: a stochastic Krylov-reduction algorithm. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE), Nice, pp. 1502–1507 (2007)

References for Section 5.2

38. Bi, Y., van der Meijs, N., Ioan, D.: Capacitance sensitivity calculation for interconnects by adjoint field technique. In: Proceedings of the 12th IEEE Workshop on Signal Propagation on Interconnects (SPI-2008), pp. 1–4 (2008)

39. Bossavit, A.: Most general non-local boundary conditions for the Maxwell equations in a bounded region. COMPEL: Int. J. Comput. Math. Electr. Electron. Eng. **19**(2), pp. 239–245 (2000)
40. CHAMELEON-RF website. http://www.chameleon-rf.org
41. Ciuprina, G., Ioan, D., Niculae, D., Fernandez Villena, J., Silveira, L.: Parametric models based on sensitivity analysis for passive components. In: Wiak, S., Krawczyk, A., Dolezel, I. (eds.) Intelligent Computer Techniques in Applied Electromangetics. Studies in Computational Intelligence, vol. 119, pp. 231–239. Springer, Berlin (2008)
42. Clemens, M., Weiland, T.: Discrete electromagnetism with the finite integration technique. Prog. Electromagn. Res. **32**, 65–87 (2001)
43. Deschrijver, D., Mrozowski, M., Dhaene, T., De Zutter, D.: Macromodeling of multiport systems using a fast implementation of the vector fitting method. IEEE Microw. Wirel. Compon. Lett. **18**(6), 383–385 (2008)
44. Ferranti, F., Antonini, G., Dhaene, T., Knockaert, L.: Passivity-preserving parameterized model order reduction for PEEC based full wave analysis. In: Proceedings of the 14th IEEE Workshop on Signal Propagation on Interconnects, pp. 65–68 (2010)
45. Goel, A.K.: High Speed VLSI Interconnections. Wiley Series in Microwave and Optical Engineering. Wiley-IEEE Press, John Wiley & Sons, Hoboken, NJ, USA (2007)
46. Gustavsen, B.: Improving the pole relocating properties of vector fitting. IEEE Trans. Power Deliv. **21**(3), 1587–1592 (2006)
47. Gustavsen, B., Semlyen, A.: Rational approximation of frequency domain responses by vector fitting. IEEE Trans. Power Deliv. **14**(3), 1052–1061 (1999)
48. Ioan, D., Ciuprina, G.: Reduced order models of on-chip passive components and interconnects, workbench and test structures. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 447–467. Springer, Berlin (2008)
49. Ioan, D., Ciuprina, G., Radulescu, M.: Algebraic sparsefied partial equivalent electric circuit – ASPEEC. In: Anile, A.M., Alì, G., Mascali, G. (eds.) Scientific Computing in Electrical Engineering. Series Mathematics in Industry vol. 9, pp. 45–50. Springer, Berlin (2006)
50. Ioan, D., Ciuprina, G., Radulescu, M.: Absorbing boundary conditions for compact modeling of on-chip passive structures. COMPEL: Int. J. Comput. Math. Electr. Electron. Eng. **25**(3), 652–659 (2006)
51. Ioan, D., Ciuprina, G., Radulescu, M., Seebacher, E.: Compact modeling and fast simulation of on-chip interconnect lines. IEEE Trans. Magn. **42**(4), 547–550 (2006)
52. Ioan, D., Ciuprina, G., Kula, S.: Reduced order models for HF interconnect over lossy semiconductor substrate. In: Proceedings of the 11th IEEE Workshop on SPI, pp. 233–236 (2007)
53. Ioan, D., Ciuprina, G., Schilders, W.: Parametric models based on the adjoint field technique for RF passive integrated components. IEEE Trans. Magn. **44**(6), 1658–1661 (2008)
54. Ioan, D., Schilders, W., Ciuprina, G., van der Meijs, N., Schoenmaker, W.: Models for integrated components coupled with their environment. COMPEL: Int. J. Comput. Math. Electr. Electron. Eng. **27**(4), 820–828 (2008)
55. Kinzelbach, H.: Statistical variations of interconnect parasitics: extraction and circuit simulation. In: Proceedings of the 10th IEEE Workshop on Signal Propagation on Interconnects (SPI), pp. 33–36 (2006)
56. Kula, S.: Reduced order models of interconnects in high frequency integrated circuits. Ph.D.-Thesis, Politehnica University of Bucharest (2009)
57. Palenius, T., Roos, J.: Comparison of reduced-order interconnect macromodels for time-domain simulation. IEEE Trans. Microw. Theory Tech. **52**(9), 2240-Ű2250 (2004)
58. Răduleţ, R., Timotin, A., Ţugulea, A.: The propagation equations with transient parameters for long lines with losses. Rev. Roum. Sci. Tech. **15**(4), 585–599 (1979)

59. Ştefănescu, A.: Parametric models for interconnections from analogue high frequency integrated circuits. Ph.D.-Thesis, Politehnica University of Bucharest (2009)
60. Stefanescu, A., Ioan, D., Ciuprina, G.: Parametric models of transmission lines based on first order sensitivities. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering 2008. Mathematics in Industry, vol. 14, pp. 29–36. Springer, Berlin/Heidelberg (2010)
61. The Vector Fitting Web Site. http://www.energy.sintef.no/produkt/VECTFIT/index.asp
62. van der Meijs, N., Fokkema, J.: VLSI circuit reconstruction from mask topology. VLSI J. Integr. **2**(3), 85–119 (1984)

References for Section 5.3

63. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM Publications, Philadelphia (2005)
64. Armbruster, H., Feldmann, U., Frerichs, M.: Analysis based reduction using sensitivity analysis. In: Proceedings of the 10th IEEE Workshop on Signal Propagation on Interconnects (SPI), pp. 29–32 (2006)
65. Augustin, F., Gilg, A., Paffrath, M., Rentrop, P., Wever, U.: Polynomial chaos for the approximation of uncertainties: chances and limits. Eur. J. Appl. Math. **19**, 149–190 (2008)
66. Baur, U., Beattie, C., Benner, P., Gugercin, S.: Interpolatory projection methods for parameterized model reduction. SIAM J. Comput. **33**, 2489–2518 (2011)
67. Benner, P.: Advances in balancing-related model reduction for circuit simulation. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 469–482. Springer, Berlin/Heidelberg (2010)
68. Benner, P., Schneider, A.: Model reduction for linear descriptor dystems with many ports. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, pp. 137–143. Springer, Berlin/New York (2012)
69. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin (2005)
70. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Berlin (2011)
71. Bi, Y., van der Kolk, K.-J., Fernández Villena, J., Silveira, L.M., van der Meijs, N.: Fast statistical analysis of RC nets subject to manufacturing variabilities. In: Proceedings of the DATE 2011, Grenoble, 14–18 Mar 2011
72. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods – Fundamentals in Single Domains. Springer, Berlin (2010)
73. Cao, Y., Petzold, L.R.: A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. SIAM J. Sci. Comput. **26**(2), 359–374 (2004)
74. Cao, Y., Li, S., Petzold, L.: Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. SIAM J. Sci. Comput. **149**, 171–191 (2002)
75. Cao, Y., Li, S., Petzold, L., Serban, R.: Adjoint sensitivity for differential-algebraic equations: the adjoint DAE system and its numerical solution. SIAM J. Sci. Comput. **24**(3), 1076–1089 (2002)
76. Conn, A.R., Haring, R.A., Visweswariah, C., Wu, C.W.: Circuit optimization via adjoint Lagrangians. In: Proceedings of the ICCAD, San Jose, pp. 281–288 Nov 1997
77. Conn, A.R., Coulman, P.K., Haring, R.A., Morrill, G.L., Visweswariah, C., Wu, C.W.: JiffyTune: circuit optimization using time-domain sensitivities. IEEE Trans. CAD ICs Syst. **17**(12), 1292–1309 (1998)
78. Daldoss, L., Gubian, P., Quarantelli, M.: Multiparameter time-domain sensitivity computation. IEEE Trans. Circuits Syst. I: Fund. Theory Appl. **48**(11), 1296–1307 (2001)

79. Echeverría Ciaurri, D.: Multi-level optimization: space mapping and manifold mapping. Ph.D.-Thesis, University of Amsterdam (2007). http://dare.uva.nl/document/45897
80. Echeverría, D., Lahaye, D., Hemker, P.W.: Space mapping and defect correction. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 157–176. Springer, Berlin/Heidelberg (2008)
81. Ernst, O.G., Mugler, A., Starkloff, H.-J., Ullmann, E.: On the convergence of generalized polynomial chaos expansions. ESAIM: Math. Model. Numer. Anal. **46**, 317–339 (2012)
82. Errico, R.M.: What is an adjoint model?. Bull. Am. Meteorol. Soc. **78**, 2577–2591 (1997)
83. Feng, L.: Parameter independent model order reduction. Math. Comput. Simul. **68**(3), 221–234 (2005)
84. Feng, L., Benner, P.: A robust algorithm for parametric model order reduction. PAMM Proc. Appl. Math. Mech. **7**, 1021501–1021502 (2007)
85. Fernández Villena, J., Silveira, L.M.: Multi-dimensional automatic sampling schemes for multi-point modeling methodologies. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **30**(8), 1141–1151 (2011)
86. Fijnvandraat, J.G., Houben, S.H.M.J., ter Maten, E.J.W., Peters, J.M.F.: Time domain analog circuit simulation. J. Comput. Appl. Math. **185**, 441–459 (2006)
87. Gautschi, W.: OPQ: a Matlab suite of programs for generating orthogonal polynomials and related quadrature rules (2002). http://www.cs.purdue.edu/archives/2002/wxg/codes
88. Gautschi, W.: Orthogonal polynomials (in Matlab). J. Comput. Appl. Math. **178**, 215–234 (2005)
89. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
90. Günther, M., Feldmann, U., ter Maten, J.: Modelling an discretization of circuit problems. In: Schilders, W.H.A., ter Maten, E.J.W. (eds.) Handbook of Numerical Analysis, Volume XIII. Special Volume: Numerical Methods in Electromagnetics, pp. 523–659, Chapter 6. Elsevier Science, Amsterdam/Boston (2005)
91. Häusler, R., Kinzelbach, H.: Sensitivity-based stochastic analysis method for power variations. In: Proceedings of the Analog '06. VDE Verlag (2006)
92. Hemker, P.W., Echeverría, D.: Manifold mapping for multilevel optimization. In: Ciuprina, G., Ioan, D. (eds.) Scientific Computing in Electrical Engineering SCEE 2006. Series Mathematics in Industry, vol. 11, pp. 325–330. Springer, Berlin/New York (2007)
93. Hocevar, D.E., Yang, P., Trick, T.N., Epler, B.D.: Transient sensitivity computation for MOSFET circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **4**(4), 609–620 (1985)
94. Homescu, C., Petzold, L.R., Serban, R.: Error estimation for reduced-order models of dynamical systems. SIAM Rev. **49**(2), 277–299 (2007)
95. Ilievski, Z.: Model order reduction and sensitivity analysis. Ph.D.-Thesis, TU Eindhoven (2010). http://alexandria.tue.nl/extra2/201010770.pdf
96. Ilievski, Z., Xu, H., Verhoeven, A., ter Maten, E.J.W., Schilders, W.H.A., Mattheij, R.M.M.: Adjoint transient sensitivity analysis in circuit simulation. In: Ciuprina, G., Ioan, D. (eds.) Scientific Computing in Electrical Engineering SCEE 2006. Series Mathematics in Industry, vol. 11, pp. 183–189. Springer, Berlin/New York (2007)
97. Ionutiu, R.: Model order reduction for multi-terminal Systems – with applications to circuit simulation. Ph.D.-Thesis, TU Eindhoven (2011). http://alexandria.tue.nl/extra2/716352.pdf
98. Knyazev, A.V., Argentati, M.E.: Principle angles between subspaces in an A-based scalar product: algorithms and perturbation estimates. SIAM J. Sci. Comput. **23**(6), 2009-Ű2041 (2002). [Algorithm available in Matlab, The Mathworks, http://www.mathworks.com/]
99. Lang, J., Verwer, J.G.: On global error estimation and control for initial value problems. SIAM J. Sci. Comput. **29**(4), 1460-Ű1475 (2007)
100. Le Maître, O.P., Knio, O.M.: Spectral Methods for Uncertainty Quantification, with Applications to Computational Fluid Dynamics. Springer, Dordrecht (2010)

101. Lutowska, A.: Model order reduction for coupled systems using low-rank approximations. Ph.D.-Thesis, TU Eindhoven (2012). http://alexandria.tue.nl/extra2/729804.pdf
102. Parks, M.L., de Sturler, E., Mackey, G., Johnson, D.D., Maiti, S.: Recycling Krylov subspaces for sequences of linear systems. SIAM J. Sci. Comput. **28**(5), 1651–1674 (2006)
103. Pinnau, R.: Model reduction via proper orthogonal decomposition. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 95–109. Springer, Berlin/Heidelberg (2008)
104. Pulch, R., ter Maten, E.J.W.: Stochastic Galerkin methods and model order reduction for linear dynamical systems. Provisionally accepted for International Journal for Uncertainty Quantification (2015)
105. Pulch, R., ter Maten, E.J.W., Augustin, F.: Sensitivity analysis of linear dynamical systems in uncertainty quantification. PAMM - Proceedings in Applied Mathematics and Mechanics, Vol. 13, Issue 1, pp. 507–508 (2013) DOI:10.1002/pamm.201310246
106. Pulch, R., ter Maten, E.J.W., Augustin, F.: Sensitivity analysis and model order reduction for random linear dynamical systems. Mathematics and Computers in Simulation 111, pp. 80–95 (2015) DOI: http://dx.doi.org/10.1016/j.matcom.2015.01.003
107. Rathinam, M., Petzold, L.R.: A new look at proper orthogonal decomposition. SIAM J. Numer. Anal. **41**(5), 1893–1925 (2003)
108. Schilders, W.H.A.: Introduction to model order reduction. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13, pp. 3–32. Springer, Berlin/Heidelberg (2008)
109. Schilders, W.H.A.: The need for novel model order reduction techniques in the electronics industry. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 3–23. Springer, Berlin (2011)
110. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin/Heidelberg (2008)
111. Stavrakakis, K.K.: Model order reduction methods for parameterized systems in electromagnetic field simulations. Ph.D.-Thesis, TU-Darmstadt (2012)
112. Stavrakakis, K., Wittig, T., Ackermann, W., Weiland, T.: Linearization of parametric FIT-discretized systems for model order reduction. IEEE Trans. Magn. **45**(3), 1380–1383 (2009)
113. Stavrakakis, K., Wittig, T., Ackermann, W., Weiland, T.: Three dimensional geometry variations of FIT systems for model order reduction. In: 2010 URSI International Symposium on Electromagnetic Theory, pp. 788–791 (2010)
114. Stavrakakis, K., Wittig, T., Ackermann, W., Weiland, T.: Model order reduction methods for multivariate parameterized dynamical systems obtained by the finite integration theory. In: 2011 URSI General Assembly and Scientifc Symposium, p. 4 (2011)
115. Stavrakakis, K., Wittig, T., Ackermann, W., Weiland, T.: Parametric model order reduction by neighbouring subspaces. In: Michielsen, B., Poirier, J.-R. (eds.) Scientific Computing in Electrical Engineering SCEE 2010. Series Mathematics in Industry, vol. 16, pp. 443–451. Springer, Berlin/New York (2012)
116. Stroud, A.: Approximate Calculation of Multiple Integrals. Prentice Hall, Englewood Cliffs (1971)
117. SUMO (SUrrogate MOdeling) Lab. IBCN research group of the Department of Information Technology (INTEC), Ghent University (2012). http://www.sumo.intec.ugent.be/
118. ter Maten, E.J.W., Heijmen, T.G.A., Lin, A., El Guennouni, A.: Optimization of electronic circuits. In: Cutello, V., Fotia, G., Puccio, L. (eds.) Applied and Industrial Mathematics in Italy II, Selected Contributions from the 8th SIMAI Conference. Series on Advances in Mathematics for Applied Sciences, vol. 75, pp. 573–584. World Scientific Publishing Co. Pte. Ltd., Singapore (2007)
119. ter Maten, E.J.W., Pulch, R., Schilders, W.H.A., Janssen, H.H.J.M.: Efficient calculation of Uncertainty Quantification. In: Fontes, M., Günther, M., Marheineke, N. (eds) Progress in

Industrial Mathematics at ECMI 2012, Series Mathematics in Industry Vol. 19, Springer, pp. 361–370 (2014)
120. Ugryumova, M.V.: Applications of model order reduction for IC modeling. Ph.D.-Thesis, TU Eindhoven (2011). http://alexandria.tue.nl/extra2/711015.pdf
121. Volkwein, S.: Model reduction using proper orthogonal decomposition (2008). http://www.uni-graz.at/imawww/volkwein/POD.pdf
122. Xiu, D.: Numerical integration formulas of degree two. Appl. Numer. Math. **58**, 1515–1520 (2008)
123. Xiu, D.: Fast numerical methods for stochastic computations: a review. Commun. Comput. Phys. **5**(2–4), 242–272 (2009)
124. Xiu, D.: Numerical Methods for Stochastic Computations – A Spectral Method Approach. Princeton University Press, Princeton (2010)

References for Section 5.4

125. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. Advances in Design and Control. SIAM, Philadelphia (2005)
126. Antoulas, A.C.: A new result on passivity preserving model reduction. Syst. Control Lett. **54**, 361–374 (2005)
127. Ascher, U.M., Petzold, L.R.: Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM, Philadelphia (1998)
128. Daniel, L., Siong, O.C., Chay, L.S., Lee, K.H., White, J.: A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. IEEE Trans. Comput. Aided Des. **23**(5), 678–693 (2004)
129. Feng, L., Benner, P.: A robust algorithm for parametric model order reduction based on implicit moment matching. PAMM Proc. Appl. Math. Mech. **7**, 1021501–1021502 (2007)
130. Günther, M.: Partielle differential-algebraische Systeme in der numerischen Zeitbereichsanalyse elektrischer Schaltungen. Nr. 343 in Fortschritt-Berichte VDI Serie 20. VDI, Düsseldorf (2001)
131. Günther, M., Feldmann, U.: CAD based electric circuit modeling in industry I: mathematical structure and index of network equations. Surv. Math. Ind. **8**, 97–129 (1999)
132. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, 2nd edn. Springer, Berlin (1996)
133. Li, Y., Bai, Z., Su, Y., Zeng, X.: Parameterized model order reduction via a two-directional Arnoldi process. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 868–873 (2007)
134. Li, Y.-T., Bai, Z., Su, Y., Zeng, X.: Model order reduction of parameterized interconnect networks via a two-directional Arnoldi process. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(9), 1571–1582 (2008)
135. Mohaghegh, K.: Linear and nonlinear model order reduction for numerical simulation of electric circuits. Ph.D.-Thesis, Bergische Universität Wuppertal, Germany. Available at Logos Verlag, Berlin (2010)
136. Mohaghegh, K., Pulch, R., Striebel, M., ter Maten, J.: Model order reduction for semi-explicit systems of differential algebraic systems. In: Troch, I., Breitenecker, F. (eds) Proceedings MATHMOD 09 Vienna – Full Papers CD Volume, pp. 1256–1265 (2009)
137. Mohaghegh, K., Pulch, R., ter Maten, J.: Model order reduction using singularly perturbed systems, provisionally accepted for J. of Applied Numerical Mathematics (APNUM) (2015)

138. Phillips, J., Silveira, L.M.: Poor's man TBR: a simple model reduction scheme. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE), vol. 2, pp. 938–943 (2004)
139. Schwarz, D.E., Tischendorf, C.: Structural analysis of electric circuits and consequences for MNA. Int. J. Circuit Theory Appl. **28**, 131–162 (2000)

Chapter 6
Advanced Topics in Model Order Reduction

Davit Harutyunyan, Roxana Ionutiu, E. Jan W. ter Maten, Joost Rommes, Wil H.A. Schilders, and Michael Striebel

Abstract This chapter contains three advanced topics in model order reduction (MOR): nonlinear MOR, MOR for multi-terminals (or multi-ports) and finally an application in deriving a nonlinear macromodel covering phase shift when coupling oscillators. The sections are offered in a preferred order for reading, but can be read independently.

Section 6.1, written by Michael Striebel and E. Jan W. ter Maten, deals with MOR for nonlinear problems. Well-known methods like TPWL (Trajectory PieceWise Linear) and POD (Proper Orthogonal Decomposition) are presented. Development for POD led to some extensions: Missing Point Estimation, Adapted POD, DEIM (Discrete Empirical Interpolation Method).

D. Harutyunyan (✉)
ASML, De Run 6501, 5504 DR Veldhoven, The Netherlands
e-mail: Davit.Harutyunyan@asml.com.

R. Ionutiu
ATTE, ABB Switzerland Ltd, Austraße, 5300 Turgi, Switzerland
e-mail: Roxana.Ionutiu@ch.abb.com

E.J.W. ter Maten
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Gaußstraße 20, D-42119 Wuppertal, Germany

Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O.Box 513, 5600 Eindhoven, The Netherlands
e-mail: Jan.ter.Maten@math.uni-wuppertal.de; E.J.W.ter.Maten@tue.nl

W.H.A. Schilders
Department of Mathematics and Computer Science, CASA, Eindhoven University of Technology, P.O.Box 513, 5600 Eindhoven, the Netherlands
e-mail: W.H.A.Schilders@tue.nl

J. Rommes
Mentor Graphics, DSM/AMS, Le Viseo – Bâtiment B, 110 rue Blaise Pacal, Inovalee, 38330 Montbonnot, France
e-mail: Joost_Rommes@mentor.com

M. Striebel
ZF Lenksysteme GmbH, Richard-Bullinger-Straße 77, D-73527 Schwäbisch Gmünd, Germany
e-mail: Michael.Striebel@zf-lenksysteme.com

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_6

Section 6.2, written by Roxana Ionutiu and Joost Rommes, deals with the multi-terminal (or multi-port) problem. A crucial outcome of the research is that one should detect "important" internal unknowns, which one should not eliminate in order to keep a sparse reduced model. Such circuits come from verification problems, in which lots of parasitic elements are added to the original design. Analysis of effects due to parasitics is of vital importance during the design of large-scale integrated circuits, since it gives insight into how circuit performance is affected by undesired parasitic effects. Due to the increasing amount of interconnect and metal layers, parasitic extraction and simulation may become very time consuming or even unfeasible. Developments are presented, for reducing systems describing *R* and *RC* netlists resulting from parasitic extraction. The methods exploit tools from graph theory to improve sparsity preservation especially for circuits with multi-terminals. Circuit synthesis is applied after model reduction, and the resulting reduced netlists are tested with industrial circuit simulators. With the novel *RC* reduction method SparseMA, experiments show reduction of 95 % in the number of elements and 46x speed-up in simulation time.

Section 6.3, written by Davit Harutyunyan, Joost Rommes, E. Jan W. ter Maten and Wil H.A. Schilders, addresses the determination of phase shift when perturbing or coupling oscillators. It appears that for each oscillator the phase shift can be approximated by solving an additional scalar ordinary differential equation coupled to the main system of equations. This introduces a nonlinear coupling effect to the phase shift. That just one scalar evolution equation can describe this is a great outcome of Model Order Reduction. The motivation behind this example is described as follows. Design of integrated RF circuits requires detailed insight in the behavior of the used components. Unintended coupling and perturbation effects need to be accounted for before production, but full simulation of these effects can be expensive or infeasible. In this section we present a method to build nonlinear phase macromodels of voltage controlled oscillators. These models can be used to accurately predict the behavior of individual and mutually coupled oscillators under perturbation at a lower cost than full circuit simulations. The approach is illustrated by numerical experiments with realistic designs.

6.1 Model Order Reduction of Nonlinear Network Problems

The dynamics of an electrical circuit can in general be described by a nonlinear, first order, differential-algebraic equation (DAE) system of the form[1]:

$$\frac{d}{dt}\mathbf{q}(\mathbf{x}(t)) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \tag{6.1a}$$

[1] Section 6.1 has been written by Michael Striebel and E. Jan W. ter Maten.

completed with the output mapping

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t)). \tag{6.1b}$$

In the state equation (6.1a), which arises from applying modified nodel analysis (MNA) to the network graph, $\mathbf{x}(t) \in \mathbb{R}^n$ represents the unknown vector of circuit variables at time $t \in \mathbb{R}$; $\mathbf{q}, \mathbf{j} : \mathbb{R}^n \to \mathbb{R}^n$ describe the contribution of reactive and nonreactive elements, respectively and $\mathbf{B} \in \mathbb{R}^{n \times m}$ distributes the input excitation $\mathbf{u} : \mathbb{R} \to \mathbb{R}^m$. The system's response $\mathbf{y}(t) \in \mathbb{R}^p$ is a possibly nonlinear function $\mathbf{h} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ of the system's state $\mathbf{x}(t)$ and inputs $\mathbf{u}(t)$.

In circuit design, (6.1a) is often not considered to describe the overall design but rather to be a model of a subcircuit or subblock. Connection to and communication with a block's environment is done via its terminals, i.e. external nodes. Therefore, we assume in the remainder of this document that the inputs $\mathbf{u}(t)$ and outputs $\mathbf{y}(t)$ denote terminal voltages and terminal currents, respectively, or vice versa, which are injected and extracted linearly, i.e., the output mapping is assumed to be of the form

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t), \tag{6.1c}$$

with $\mathbf{C} \in \mathbb{R}^{p \times n}$.

The dimension n of the unknown vector $\mathbf{x}(t)$ is of the order of the number of elements in the circuit, which can easily reach hundreds of millions. Therefore, one may solve the network equations (6.1a) and (6.1c) by means of computer algebra in an unreasonable amount of time only.

Model order reduction (MOR) aims to replace the original model (6.1a) and (6.1c) by a system

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{q}}(\mathbf{z}(t)) + \hat{\mathbf{j}}(\mathbf{z}(t)) + \hat{\mathbf{B}}\mathbf{u}(t) &= \mathbf{0}, \\ \hat{\mathbf{y}}(t) &= \hat{\mathbf{C}}\mathbf{z}(t), \end{aligned} \tag{6.2}$$

with $\mathbf{z}(t) \in \mathbb{R}^r$; $\hat{\mathbf{q}}, \hat{\mathbf{j}} : \mathbb{R}^r \to \mathbb{R}^r$ and $\hat{\mathbf{B}} \in \mathbb{R}^{r \times m}$ and $\hat{\mathbf{C}} \in \mathbb{R}^{p \times r}$, which can compute the system response $\hat{\mathbf{y}}(t) \in \mathbb{R}^p$ that is sufficiently close to $\mathbf{y}(t)$ given the same input signal $\mathbf{u}(t)$, but in much less time.

6.1.1 Linear Versus Nonlinear Model Order Reduction

So far most research effort was spent on developing and analysing MOR techniques suitable for linear problems. For an overview on these methods we refer to [1].

When trying to transfer approaches from linear MOR, fundamental differences emerge.

To see this, first consider a linear problem of the form

$$\begin{aligned} \mathbf{E}\frac{d}{dt}\mathbf{x}(t) + \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{0}, \quad &\text{with } \mathbf{E}, \mathbf{A} \in \mathbb{R}^{n\times n}, \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \end{aligned} \tag{6.3}$$

The state $\mathbf{x}(t)$ is approximated in a lower dimensional space of dimension $r \ll n$, spanned by basis vectors which we subsume in $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathbb{R}^{n\times r}$:

$$\mathbf{x}(t) \approx \mathbf{V}\mathbf{z}(t), \quad \text{with } \mathbf{z}(t) \in \mathbb{R}^r . \tag{6.4}$$

The reduced state $\mathbf{z}(t)$, i.e., the coefficients of the expansion in the reduced space, is defined by a reduced dynamical system. Applying Galerkin technique, this reduced system arises from projecting (6.3) on a test space spanned by the columns of some matrix $\mathbf{W} \in \mathbb{R}^{n\times r}$. There, $\mathbf{W}$ and $\mathbf{V}$ are chosen, such that their columns are biorthonormal, i.e., $\mathbf{W}^T\mathbf{V} = \mathbf{I}_{r\times r}$. The Galerkin projection[2] yields

$$\begin{aligned} \hat{\mathbf{E}}\frac{d}{dt}\mathbf{z}(t) + \hat{\mathbf{A}}\mathbf{z}(t) + \hat{\mathbf{B}}\mathbf{u}(t) = \mathbf{0}, \\ \mathbf{y}(t) = \hat{\mathbf{C}}\mathbf{z}(t) \end{aligned} \tag{6.5}$$

with $\hat{\mathbf{E}} = \mathbf{W}^T\mathbf{E}\mathbf{V}$, $\hat{\mathbf{A}} = \mathbf{W}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{r\times r}$ and $\hat{\mathbf{B}} = \mathbf{W}^T\mathbf{B} \in \mathbb{R}^{r\times m}$, $\hat{\mathbf{C}} = \mathbf{C}\mathbf{V} \in \mathbb{R}^{p\times r}$. The system matrices $\hat{\mathbf{E}}$, $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, $\hat{\mathbf{C}}$ of this reduced substitute model are of smaller dimension and constant, i.e., need to be computed only once. However, $\hat{\mathbf{E}}, \hat{\mathbf{A}}$ are usually dense whereas the system matrices $\mathbf{E}$ and $\mathbf{A}$ are usually very sparse.

Applying the same technique directly to the nonlinear system means obtaining the reduced formulation (6.2) by defining $\hat{\mathbf{q}}(\mathbf{z}) = \mathbf{W}^T\mathbf{q}(\mathbf{V}\mathbf{z})$ and $\hat{\mathbf{j}}(\mathbf{z}) = \mathbf{W}^T\mathbf{j}(\mathbf{V}\mathbf{z})$. Clearly, $\hat{\mathbf{q}}$ and $\hat{\mathbf{j}}$ map from $\mathbb{R}^r$ to $\mathbb{R}^r$.

To solve network problems of type (6.2) numerically, usually multistep methods are used. This means that at each timepoint t_l a nonlinear equation

$$\alpha\hat{\mathbf{q}}(\mathbf{z}_l) + \hat{\boldsymbol{\beta}}_l + \hat{\mathbf{j}}(\mathbf{z}_l) + \hat{\mathbf{B}}\mathbf{u}(t_l) = \mathbf{0} \tag{6.6}$$

has to be solved for $\mathbf{z}_l$ which is the approximation of $\mathbf{z}(t_l)$. In the above equation α is the integration coefficient of the method and $\hat{\boldsymbol{\beta}}_l \in \mathbb{R}^r$ contains history from previous timesteps. Newton techniques that are used to solve (6.6) usually require an update of the system's Jacobian matrix in each iterations ν:

$$\hat{\mathbf{J}}_l^{(\nu)} = \left(\alpha\frac{\partial\hat{\mathbf{q}}}{\partial\mathbf{z}} + \frac{\partial\hat{\mathbf{j}}}{\partial\mathbf{z}}\right)\Big|_{\mathbf{z}=\mathbf{z}_l^{(\nu)}} = \mathbf{W}^T\left[\alpha\frac{\partial\mathbf{q}}{\partial\mathbf{x}} + \frac{\partial\mathbf{j}}{\partial\mathbf{x}}\right]\Big|_{\mathbf{x}^{(\nu)}=\mathbf{V}\mathbf{z}_l^{(\nu)}}\mathbf{V}. \tag{6.7}$$

[2] Most frequently $\mathbf{V}$ is constructed to be orthogonal, such that $\mathbf{W} = \mathbf{V}$ can be chosen.

The evaluation of the reduced system, i.e., $\hat{\mathbf{q}}$ and $\hat{\mathbf{j}}$, necessitates in each step the back projection of the argument $\mathbf{z}$ to its counterpart $\mathbf{Vz}$ followed by the evaluation of the full system $\mathbf{q}$ and $\mathbf{j}$ and the projection to the reduced space with $\mathbf{W}$ and $\mathbf{V}$.

Consequently, with respect to computation time no reduction will be obtained unless additional measures are taken or other strategies are pursued.

6.1.2 Some Nonlinear MOR Techniques

In MOR for linear systems especially methods based on Krylov subspaces [19] and balanced realization [30] are well understood and highly elaborated. Hence, it seems likely to adapt them to nonlinear problems, too. In the following, we shortly describe these approaches and give references for further reading.

6.1.2.1 Krylov Subspace Methods in Nonlinear MOR

In linear MOR Krylov subspace methods are used to construct reduced order models of systems (6.3) such that the moments, i.e., the coefficients in a Taylor expansion of the frequency domain transfer function of original and reduced system match up to a certain order. The transfer function $\mathbf{H} : \mathbb{C} \to \mathbb{C}^{p\times m}$ is defined by the linear equation $\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} + \mathbf{A})^{-1}\mathbf{B}$.

It is not straightforward to define a transfer function for the nonlinear problem (6.1a) and (6.1c). Instead, there are Krylov based techniques that deal with bilinear systems (6.8) or linear periodically time varying (LPTV) problems (6.9).

$$\text{bilinear system:} \qquad \frac{d}{dt}\hat{\mathbf{x}}(t) + \hat{\mathbf{A}}\hat{\mathbf{x}}(t) + \hat{\mathbf{N}}\hat{\mathbf{x}}(t)\mathbf{u}(t) + \hat{\mathbf{B}}\mathbf{u}(t) = \mathbf{0} \tag{6.8}$$

$$\text{LPTV system:} \qquad \frac{d}{dt}[\mathbf{E}(t)\mathbf{x}(t)] + \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) = \mathbf{0} \tag{6.9}$$

The type of problem (6.8) arises from expanding a nonlinear problem $\dot{\mathbf{x}}(t) + \mathbf{f}(\mathbf{x}(t)) + \mathbf{Bu}(t) = \mathbf{0}$ around an equilibrium point. Systems of type (6.9) with matrices $\mathbf{E}(t), \mathbf{A}(t)$ that are periodic with some period T one gets when linearising the system (6.1) around a periodic steady state solution with $\mathbf{x}^0(t + T) = \mathbf{x}^0(t)$.

Volterra-series expansion, followed by multivariable Laplace-transformation and multimoment expansions are the key to apply Krylov subspace based MOR. For further reading we refer to [15] and the references therein.

In case of the LPTV systems, a timevarying system function $\mathbf{H}(s,t)$ can be defined. This plays the role of a transfer function and can be determined by a differential equation. $\mathbf{H}(s,t)$ has to be determined in terms of time- or frequency samples on $[0, T)$ for one s. Krylov techniques can then be applied to get a reduced system with which samples for different frequencies s can be constructed. We refer to [21] and the references therein.

Given a nonlinear problem (6.1a) and (6.1c) of dimension n, the bilinear system that is reduced actually has a dimension of $n + n^2 + n^3 + \cdots$, depending on the order of the expansion. Similar, the system in the LPTV case that is subject to reduction has dimension $k \cdot n$ with k being the number of timesamples in the initial determination of $\mathbf{H}(s, t)$. Therefore, it seems that these methods are suitable for small to medium sized nonlinear problems only.

6.1.2.2 Balanced Truncation in Nonlinear MOR

The energy $L_c(\mathbf{x}_0)$ that is needed to drive a system to a given state $\mathbf{x}_0$ and the energy $L_o(\mathbf{x}_0)$ the system provides to observe the state $\mathbf{x}_0$ it is in are the main terms in Balanced Truncation. A system is called balanced if states that are hard to reach are also hard to observe and vice versa, i.e. $L_c(\mathbf{x})$ large implies $L_o(\mathbf{x})$. Truncation, i.e. reduced order modelling is then done by eliminating these states.

For linear problems L_c and L_o are connected directly, by means of algebraic calculation, to the reachability and observability Gramians $\mathbf{P}$ and $\mathbf{Q}$, respectively. These can be computed from Lyapunov equations, involving the system matrices $\mathbf{E}, \mathbf{A}, \mathbf{B}, \mathbf{C}$ of the linear system (6.3). Balancing is reached by transforming the state space such that $\mathbf{P}$ and $\mathbf{Q}$ are simultaneously diagonalised:

$$\mathbf{P} = \mathbf{Q} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$$

with the so called Hankel singular values $\sigma_1, \ldots, \sigma_n$. From the basis that arises from the transformation only those basis vectors that correspond to large Hankel singular values are kept. The main advantage of this approach is that there exists an a priori computable error bound for the truncated system.

In transferring Balanced Truncation to nonlinear problems, three main tracks can be recognized. Energy consideration is the common ground for the three directions.

In the approach suggested in [20] the energy functions arise from solving Hamilton-Jacobi differential equations. Similar to the linear case, a state-space transformation is searched such that L_c and L_o are formulated as quadratic form with diagonal matrix. The magnitude of the entries are then basis to truncation again. The transformation is now state dependent, and instead of singular values, we get singular value functions. As the Hamilton-Jacobi system has to be solved and the varying state-space transformations have to be computed, it is an open issue, how the theory could be applied in a computer environment.

In Sliding Interval Balancing [46], the nonlinear problem is first linearised around a nominal trajectory, giving a linear time varying system like (6.9). At each state finite time reachability and observability Gramians are defined and approximated by truncated Taylor series expansion. Analytic calculations, basically the series expansions, connect the local balancing transformation smoothly. This necessary step is the limiting factor for this approach in circuit simulation.

Finally, balancing is also applied to bilinear systems (6.8). Here the key tool are so called algebraic Gramians arising from generalised Lyapunov equations. However, no one-to-one connection between these Gramians and the energy functions L_c, L_o can be made, but rather they can serve to get approximative bounds for the aforementioned. Furthermore, convergence parameters have to be introduced to guarantee the solvability of the generalised Lyapunov equations. For further details we refer to [9, 14] and the references therein.

6.1.3 TPWL and POD

In view of high dimensional problems in circuit simulation and feasibility in a computational environment, Trajectory PieceWise Linearization (TPWL) and Proper Orthogonal Decomposition (POD) are amongst the most promising approaches for the time being. The basic idea of TPWL is to replace nonlinearity with a collection of linear substitute problems and apply MOR on these. The background of POD is to identify a low dimensional manifold the solution resides on and reformulate the problem in such a way that it is solved in terms of the basis of this principal manifold.

In the following we give more details on the steps done for both approaches.

6.1.3.1 Trajectory PieceWise Linearization

The idea of TPWL [33], is to represent the full nonlinear system (6.1a) and (6.1c) by a set of order reduced linear models that can reproduce the typical behaviour of the system.

Since its introduction in [33, 34], TPWL has gained a lot of interest and several adaptions have been made, see e.g., [18, 39, 49]. In the following we will basically follow the lines in the original works [33, 34] and briefly mention alternatives that have been suggested.

For extracting a model, a training input $\bar{\mathbf{u}}(t)$ for $t \in [t_{\text{start}}, t_{\text{end}}]$ is chosen and a transient simulation is run in order to get a trajectory, i.e. a collection of points $\mathbf{x}_0, \ldots, \mathbf{x}_N$, approximating $\mathbf{x}(t_i)$ at timepoints $t_{\text{start}} = t_0 < t_1 < \cdots < t_N = t_{\text{end}}$. The training input is chosen such that the trajectory it causes, reflects the typical state of the system. On the trajectory, points $\{\mathbf{x}_0^{\text{lin}}, \ldots, \mathbf{x}_s^{\text{lin}}\} \subset \{\mathbf{x}_0, \ldots, \mathbf{x}_N\}$ are chosen around which the nonlinear functions $\mathbf{q}$ and $\mathbf{j}$ are linearised:

$$\mathbf{q}(\mathbf{x}(t)) \approx \mathbf{q}(\mathbf{x}_i^{\text{lin}}) + \mathbf{E}_i \cdot \left(\mathbf{x}(t) - \mathbf{x}_i^{\text{lin}}\right); \quad \mathbf{j}(\mathbf{x}(t)) \approx \mathbf{j}(\mathbf{x}_i^{\text{lin}}) + \mathbf{A}_i \cdot \left(\mathbf{x}(t) - \mathbf{x}_i^{\text{lin}}\right), \tag{6.10}$$

with $\mathbf{E}_i = \frac{\partial \mathbf{q}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_i^{\text{lin}}}$ and $\mathbf{A}_i = \frac{\partial \mathbf{j}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_i^{\text{lin}}}$.

Then the nonlinear state-space equation (6.1a) can locally be replaced locally around $\mathbf{x}_i^{\text{lin}}$ for $i = 1, \ldots, s$ by

$$\frac{d}{dt}\left[\mathbf{E}_i\mathbf{x}(\mathbf{t}) + \boldsymbol{\delta}_i\right] + \mathbf{A}_i\mathbf{x}(t) + \boldsymbol{\gamma}_i + \mathbf{Bu}(t) = \mathbf{0}, \tag{6.11}$$

with $\boldsymbol{\delta}_i = \mathbf{q}(\mathbf{x}_i^{\text{lin}}) - \mathbf{E}_i\mathbf{x}_i^{\text{lin}}$ and $\boldsymbol{\gamma}_i = \mathbf{j}(\mathbf{x}_i^{\text{lin}}) - \mathbf{A}_i\mathbf{x}_i^{\text{lin}}$.

One approach, used by Rewieński [33], to get a model that represents the nonlinear problem on a larger range, is to combine the local models (6.11) to

$$\frac{d}{dt}\left(\sum_{i=0}^{s} w_i(\mathbf{x}(\mathbf{t}))\left[\mathbf{E_i}\mathbf{x}(t) + \boldsymbol{\delta}_i\right]\right) + \sum_{i=0}^{s} w_i(\mathbf{x}(\mathbf{t}))\left[\mathbf{A}_i\mathbf{x}(t) + \boldsymbol{\gamma}_i\right] + \mathbf{Bu}(t) = \mathbf{0}, \tag{6.12a}$$

where $w_i : \mathbb{R}^n \to [0, 1]$ for $s = 1, \ldots, s$ is a state-dependent weight-function. The weighting functions w_i are chosen such that $w_i(\mathbf{x}(t))$ is large for $\mathbf{x}$ close to $\mathbf{x}_i^{\text{lin}}$ and such that $w_0(\mathbf{x}(t)) + \cdots + w_s(\mathbf{x}(t)) = 1$.

A different way to define a global substitute model, suggested by Voß [49] is

$$\sum_{i=0}^{s} w_i(\mathbf{x}(t))\left(\mathbf{E}_i\frac{d}{dt}\mathbf{x}(t) + \mathbf{A}_i\mathbf{x}(t) + \gamma_i\right) + \mathbf{Bu}(t) = \mathbf{0}. \tag{6.12b}$$

Although different in definition, in deployment both approaches (6.12a) and (6.12b) are equivalent, as we will see later.

Figure 6.1 illustrates the idea: Along a training trajectory, extracted from a full dimensional simulation, a set of locally valid linear models is created. When this model is used for simulation with a different input, the existing linear models are turned on and off, adapted to the state the system is in at one moment.

Simulation of the piecewise linearized system (6.12a) or (6.12b) may already be faster than simulation of the original nonlinear system. However, the linearized system can be reduced by using model order techniques for linear systems to increase efficiency.

The main difference between linear MOR and TPWL is that the latter introduces in addition to the application of a linear MOR technique the selection of lineariza-

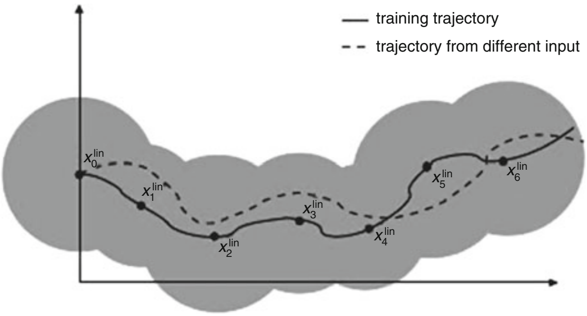


Fig. 6.1 TPWL – model extraction and usage

tion points (to get a linear problem) and the weighting of the linear submodels (to recover the global nonlinear behavior).

Reducing the System

Basically, any MOR-technique for linear problems can be applied to the linear submodels (6.11), i.e., $(\mathbf{E}_i, \mathbf{A}_i, [\mathbf{B}, \boldsymbol{\gamma}_i], \mathbf{C})$. Note that we did extend the columns of $\mathbf{B}$ with $\boldsymbol{\gamma}_i$ – thus MOR may exploit refinements for multiple terminals (see, f.i., Sect. 6.2 in this Chapter). Originally Rewieński [33] proposed the usage of Krylov-based reduction. Vasilyev, Rewieński and White [40] introduced Balanced Truncation to TPWL and Voß [49] uses Poor Man's TBR (PMTBR) as linear MOR kernel. Each of these methods creates local subspaces, spanned by the columns of projection matrices $\mathbf{V}_i \in \mathbb{R}^{n \times r_i}$ for $i = 0, \dots, s$. For some comparisons on different MOR methods used within TPWL, see [29]. For comparison between TPWL and POD (see Sect. 6.1.3.2), see [7, 42].

In a second step one global subspace is created from the information contained in the local subspaces. This is done by applying a singular value decomposition (SVD) on the aggregated matrix $\mathbf{V}_{\text{agg}} = [\mathbf{V}_0, \mathbf{x}_0^{\text{lin}}; \dots; \mathbf{V}_s, \mathbf{x}_s^{\text{lin}}]$. Note that the $\mathbf{x}_j^{\text{lin}}$ are "snapshots" in time of the nonlinear solution. Their span actually forms a POD-subspace (see Sect. 6.1.3.2) that is collected on-the-fly within TPWL. The inclusion reduces the error of the solution of the reduced model [7].

The final reduced subspace is then spanned by the r dominating left singular vectors, subsumed in $\mathbf{V} \in \mathbb{R}^{n \times r}$. Furthermore let $\mathbf{W} \in \mathbb{R}^{n \times r}$ be the corresponding test matrix, where often we have $\mathbf{W} = \mathbf{V}$. Then a reduced order model for the piecewise-linearized system (6.12a) is

$$\frac{d}{dt}\left(\sum_{i=0}^{s} w_i(\mathbf{V}\mathbf{z}(t))\left[\hat{\mathbf{E}}_i\mathbf{z}(t) + \hat{\boldsymbol{\delta}}_i\right]\right) + \sum_{i=0}^{s} w_i(\mathbf{V}\mathbf{z}(t))\left[\hat{\mathbf{A}}_i\mathbf{z}(t) + \hat{\boldsymbol{\gamma}}_i\right] + \hat{\mathbf{B}}\mathbf{u}(t) = \mathbf{0}, \tag{6.13}$$

with $\hat{\mathbf{E}}_i = \mathbf{W}^T\mathbf{E}_i\mathbf{V}$, $\hat{\mathbf{A}}_i = \mathbf{W}^T\mathbf{A}_i\mathbf{V}$, $\hat{\boldsymbol{\delta}}_i = \mathbf{W}^T\boldsymbol{\delta}_i$, $\hat{\boldsymbol{\gamma}}_i = \mathbf{W}^T\boldsymbol{\gamma}_i$ and $\hat{\mathbf{B}} = \mathbf{W}^T\mathbf{B}$.

Selection of Linearization Points

A crucial point in TPWL is to decide, which linearization points $\mathbf{x}_0^{\text{lin}}, \dots, \mathbf{x}_s^{\text{lin}}$ should be chosen. With a large number of such points, we could expect to find a linear model suitable to reproduce the nonlinear behaviour locally. But, this would especially cause to store huge amount of data, making the final model slow. On the other hand, if too few points are chosen to linearise around, the nonlinear behaviour will not be reflected correctly. Different strategies to decide upon adding a new linearization point, and hence a new model automatically exist:

- In the original work, Rewieński [33, 34] suggests to check at each accepted timepoint t during simulation for the relative distance of the current state $\mathbf{x}_k \approx \mathbf{x}(t_k)$ of the nonlinear problem to all yet existing i linearization states

$\mathbf{x}_0^{\mathrm{lin}}, \ldots, \mathbf{x}_{i-1}^{\mathrm{lin}}$. If the minimum is equal to or greater than some parameter $\alpha > 0$, i.e.

$$\min_{0 \le j \le i-1} \left(\frac{\|\mathbf{x}_k - \mathbf{x}_j^{\mathrm{lin}}\|_\infty}{\|\mathbf{x}_j^{\mathrm{lin}}\|_\infty} \right) \ge \alpha, \tag{6.14}$$

$\mathbf{x}_k$ becomes the $(i+1)$st linearization point. Accordingly, a new linear model, arising from linearizing around $\mathbf{x}_{i+1}^{\mathrm{lin}} = \mathbf{x}_k$ is added to the collection. The parameter α is chosen depending on the steady state of the system (6.1a).

- In [49] the mismatch of nonlinear and linear system motivates the creation of a new linearization point and an additional linear model: at each timepoint during training both the nonlinear and a currently valid linear system are computed in parallel with the same stepsize. If the difference of the two approximations to the true solution at a timepoint t_{k+1} becomes too large, a new linear model is created from linearizing the nonlinear system around the state the system was in at the previous timepoint t_k.
- The strategy pursued by Dong an Roychowdhury [18] is similar to (6.14). Here, not deviations between states but function evaluations at the current approximation $\mathbf{x}_k$ and the linearization points $\mathbf{x}_j^{\mathrm{lin}}$ are considered.
- In Martinez [28] an optimization criterion is used to determine the linearization points. The technique exploits the Hessian of the system as an error bound metric.

Determination of the Weights

When replacing the full nonlinear problem with the TPWL model (6.13) the weights $w_i : \mathbb{R}^n \to [0, 1]$ are responsible for switching between the linear submodels, i.e., for choosing the linear model that reflects best the behaviour caused by the nonlinearity.

Besides the specifications of the desired behaviour, made before, one wants to have minimum complexity, i.e., one aims at having to deal with a combination of just a small number of linear submodels at each timepoint. It is hence obvious that the weight functions have to be nonlinear in nature. Again, different strategies exist:

- Both Rewieński [33] and Voß [49] use

$$w_i(\mathbf{x}) = e^{-\frac{\beta}{m} \cdot d_i(\mathbf{x})}, \quad \text{with } d_i(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_i^{\mathrm{lin}}\|_2 \text{ and } m = \min_i d_i(\mathbf{x}). \tag{6.15a}$$

The constant β adjusts how abrupt the change of models is. A typical value is $\beta = 25$.

- Dong and Roychowdhury [18], however, use

$$w_i(\mathbf{x}) = \left(\frac{m}{d_i(\mathbf{x})} e^{\frac{-d_i(\mathbf{x}) - m}{M}} \right)^{\mu}, \tag{6.15b}$$

where $d_i(\mathbf{x})$ and m are the same as in (6.15a) and M is the minimum distance, taken in the 2-norm, amongst the linearization points. The parameter μ is chosen from $\{1, 2\}$.

In both cases, the weights are normalized such that $\sum_i w_i(\mathbf{x}) = 1$.

Clearly the nonlinearity of the weights causes the TPWL-model (6.13) arising from (6.12a) – and similar the reduced model that would originate from (6.12b) – to be nonlinear still. That means, after applying a numerical integration scheme to (6.13) , still a nonlinear problem has to be solved to get an approximation $\mathbf{z}_k \approx \mathbf{z}(t_k)$. To overcome this problem, both Rewieński [33] and Voß [49] decouple the evaluation of the weights from the time discretisation by replacing

$$w_i(\mathbf{V}\mathbf{z}_k) \rightsquigarrow w_i(\mathbf{V}\tilde{\mathbf{z}}_k) \quad \text{with } \tilde{\mathbf{z}}_k \approx \mathbf{z}_k,$$

i.e., for calculating $\mathbf{z}_k$ from the discretisation of (6.13) at $t = t_k$, $\mathbf{z}_k$ in the weighting is replaced by a cheaper approximation $\tilde{\mathbf{z}}_k$. It is easy to see that with this action, (6.12a) and (6.12b) are equivalent.

Note: The work of Dong and Roychowdhury [18] does actually not consider a piecewise linear a but piecewise polynomial approach, i.e., the Taylor expansions in (6.10) contain one more coefficient, leading to the need for reducing local bilinear systems. Tiwary and Rutenbar [39] look into details of implementing a TPWL-technique in an economic way.

TPWL and Time-Domain MOR

In [53] the TPWL approach is combined with wavelet expansions that are defined directly in the time-domain. For wavelets in circuit simulation we refer to [16, 17] and for technical details to [10, 11, 51, 52]. After linearizing a differential equation

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{B}\mathbf{u}(t) \tag{6.16}$$

at $\mathbf{x}_i = \mathbf{x}(t_i)$, we obtain that $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{x}_i$ is approximately given by

$$\frac{d}{dt}\tilde{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}_i) + \mathbf{A}(\tilde{\mathbf{x}}(t) - \mathbf{x}_i) + \mathbf{B}\mathbf{u}(t), \tag{6.17}$$

$$= \mathbf{A}\tilde{\mathbf{x}}(t) + \mathbf{f}(\mathbf{x}_i) - \mathbf{A}\mathbf{x}_i + \mathbf{B}\mathbf{u}(t). \tag{6.18}$$

where $\mathbf{A} = \frac{\partial f(x)}{\partial x}(t_i)$. The output request $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$ transfers to $\tilde{\mathbf{y}}(t) = \mathbf{C}\mathbf{x}_i + \mathbf{C}\tilde{\mathbf{x}}(t)$, in which the first term is known. Thus it is sufficient to consider on an interval $[0, T]$ the sum of the solutions of the two problems

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{6.19}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{6.20}$$

and

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{f}(\mathbf{x}_i) - \mathbf{A}\mathbf{x}_i, \tag{6.21}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t). \tag{6.22}$$

Assuming T being integer (see [53] for the more general case), for a wavelet order J we get $M = 2^J \cdot T + 3$ basis functions $\theta_j(t)$, $j = 1, \ldots M$. We can write $\mathbf{x}(t) = \mathbf{H}_1\theta(t)$, and $\mathbf{x}(t) = \mathbf{H}_2\theta(t)$, respectively, where $\theta(t) = (\theta_1(t), \ldots, \theta_M(t))$ and $\mathbf{H}_1,\ \mathbf{H}_2 \in \mathbb{R}^{n\times M}$. We can plug these expressions into (6.19) and into (6.21). However, note that in (6.19) the source term is time-dependent, while in (6.21) the source term is constant. Hence rather then to consider (6.19), one considers

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\delta(t), \tag{6.23}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{6.24}$$

where $\delta(t)$ is an impulse excitation with the property $\int_{0_-}^{t} \delta(\tau)d\tau = 1$. Then, for (6.23), the matrix $\mathbf{H}_1$ satisfies

$$\mathbf{H}_1\frac{d}{dt}\theta(t) = \mathbf{A}\mathbf{H}_1\theta(t) + \mathbf{B}\delta(t) \tag{6.25}$$

Assuming that the wavelets have their support in $[0, T]$ we derive

$$\mathbf{H}_1\theta(t) = \mathbf{A}\mathbf{H}_1\int_{0_-}^{t}\theta(\tau)d\tau + \mathbf{B}. \tag{6.26}$$

In [53] one applies collocation using M collocation points. Next $\mathbf{H}_1$ is found after solving the resulting Sylvester equation. A similar approach is done for (6.21). Now one determines $\mathbf{V}_i = \mathit{Orthog}(\mathbf{H}_1, \mathbf{H_2}, \mathbf{x}_i)$ (note that this is similar to the multiple terminal approach mentioned before for the frequency domain case). From $(\mathbf{V}_1, \ldots, \mathbf{V}_s)$ one determines an overall orthonormal basis $\mathbf{V} \in \mathbb{R}n \times r$ that is used for the projection as before.

6.1.3.2 Proper Orthogonal Decomposition and Adaptions

The Proper Orthogonal Decomposition (POD) method, also known as the Principal Component Analysis and Karhunen–Loève expansion, provides a technique for analysing multidimensional data [24, 27].

In this section we briefly describe some basics of POD. For a more detailed introduction to POD in MOR we refer to [31, 47]. For further studies we point to [32], which addresses error analysis for MOR with POD and [50] where the connection of POD to balanced model reduction can be found.

POD sets work on data extracted from a benchmark simulation. In a finite dimensional setup like it is given by (6.1a), K snapshots of the state $\mathbf{x}_i \approx \mathbf{x}(t_i)$, the system is in during the training interval $[t_{\text{start}}, t_{\text{end}}]$, are collected in a snapshot matrix

$$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_K) \quad \in \mathbb{R}^{n\times K}. \tag{6.27}$$

The snapshots, i.e., the columns of $\mathbf{x}$, span a space of dimension $k \leq K$. We search for an orthonormal basis $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ of this space that is optimal in the sense that the time-averaged error that is made when the snapshots are expanded in the space spanned by just $r < k$ basis vectors to $\tilde{\mathbf{x}}_{r,i}$,

$$\langle \|\mathbf{x} - \tilde{\mathbf{x}}_r\|_2^2 \rangle \quad \text{with the averaging operator} \quad \langle \mathbf{f} \rangle = \frac{1}{K}\sum_{i=1}^{K} \mathbf{f}_i \tag{6.28}$$

is minimised. This least squares problem is solved by computing the eigenvalue decomposition of the state covariance matrix $\frac{1}{K}\mathbf{x}\mathbf{x}^T$ or, equivalently by the singular value decomposition (SVD) of the snapshot matrix (assuming $K > n$)

$$\mathbf{x} = \mathbf{UST} \quad \text{with} \quad \mathbf{U} \in \mathbb{R}^{n\times n}, \mathbf{T} \in \mathbb{R}^{K\times K} \text{ and } \mathbf{S} = \left(\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & \mathbf{0}_{n\times(K-n)} \\ & & \sigma_n & \end{array}\right), \tag{6.29}$$

where $\mathbf{U}t$ and $\mathbf{T}$ are orthogonal and the singular values satisfy $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n \geq 0$. The matrix $\mathbf{V} \in \mathbb{R}^{n\times r}$ whose columns span the reduced subspace is now build from the first r columns of $\mathbf{u}$, where the truncation r is chosen such that

$$\frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2} \geq \frac{d}{100}, \tag{6.30}$$

where usually $d = 99$ is usually a reasonable choice. For the, in this way constructed matrix, it holds $\mathbf{V}^T\mathbf{V} = \mathbf{I}_{r\times r}$. Therefore, Galerkin projection as described above can be applied to create a reduced system (6.2).

However, as mentioned in Sect. 6.1.1 the cost for evaluating the nonlinear functions $\mathbf{q}$, $\mathbf{j}$ is not reduced. In the following we describe some adaptions to POD that have been made to overcome this problem.

6.1.3.3 Missing Point Estimation

The Missing Point Estimation (MPE) was proposed by Astrid [2, 4] to reduce the cost of updating system information in the solution process of time varying systems arising in computational fluid dynamics. Verhoeven and Astrid [3] brought the MPE approach forward to circuit simulation.

Once a POD basis is constructed, there is no Galerkin projection deployed. Instead a numerical integration scheme is applied which in general leads to system of n nonlinear equations, analogue to (6.6), for the r dimensional unknown $\mathbf{z}_l$, that approximate $\mathbf{z}(t_l)$. In MPE this system is reduced to dimension g with $r \leq g < n$ by discarding $n-g$ equations. Formally this can be described by multiplying the system with a selection matrix[3] $\mathbf{P}_g \in \{0, 1\}^{g\times n}$, stating a g-dimensional overdetermined problem

$$\alpha\bar{\mathbf{q}}(\mathbf{V}\mathbf{z}_l) + \mathbf{P}_g\boldsymbol{\beta}_l + \bar{\mathbf{j}}(\mathbf{V}\mathbf{z}_l) + \mathbf{P}_g\mathbf{B}\mathbf{u}(t_l) = \mathbf{0}, \tag{6.31}$$

with $\bar{\mathbf{q}}(\mathbf{V}\mathbf{z}_l) = \mathbf{P}_g\mathbf{q}(\mathbf{V}\mathbf{z}_l)$ and $\bar{\mathbf{j}}(\mathbf{V}\mathbf{z}_l) = \mathbf{P}_g\mathbf{j}(\mathbf{V}\mathbf{z}_l)$. The system (6.31) is solved at each timepoint t_l for $\mathbf{z}_l$ in the least-squares sense [3, 41, 44, 45].

The effect of $\mathbf{P}_g$ operating on $\mathbf{q}(\cdot)$ and $\mathbf{j}(\cdot)$ is the same as evaluating only the $g \ll n$ components of $\mathbf{q}$ and $\mathbf{j}$ corresponding to the columns $\mathbf{P}_g$ has a 1 in.

The choice of $\mathbf{P}_g$ is motivated by identifying the g most dominant state variables, i.e., components of $\mathbf{x}$. In terms of the POD basis this is connected to restricting the orthogonal $\mathbf{V}$ to $\tilde{\mathbf{V}} = \mathbf{P}_g\mathbf{V} \in \mathbb{R}^{g\times r}$ in an optimal way. This in turn goes down to

$$\min_{\mathbf{P}_g} \| \left(\tilde{\mathbf{V}}^T\tilde{\mathbf{V}}\right)^{-1} - \mathbf{I}_{r\times r}\|. \tag{6.32}$$

Details on reasoning and solving (6.32) can be found in [4].

6.1.3.4 Adapted POD

A second approach to reduce the work of evaluating the nonlinear functions, Adapted POD, was proposed in [41, 43–45]. Having done an SVD (6.29) on the snapshot matrix, not directly a projection matrix $\mathbf{V}$ is defined from the singular values and vectors. Instead the matrix $\mathbf{L} = \mathbf{u}\Sigma \in \mathbb{R}^{n\times n}$, with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ is defined. Hence, $\mathbf{L}$ arises from scaling the left-singular vectors with the corresponding singular values. Although $\mathbf{L}$ is not orthogonal, its columns are. Next we transform the original system (6.1a) by writing $\mathbf{x}(t) = \mathbf{L}\mathbf{w}(t)$ with $\mathbf{w}(t) \in \mathbb{R}^n$ and using the Galerkin approach:

$$\frac{d}{dt}\left[\mathbf{L}^T\mathbf{q}(\mathbf{L}\mathbf{w}(t))\right] + \mathbf{L}^T\mathbf{j}(\mathbf{L}\mathbf{w}(t)) + \mathbf{L}^T\mathbf{B}\mathbf{u}(t) = \mathbf{0}. \tag{6.33}$$

At this point, $\mathbf{L}$ and $\mathbf{L}^T$ are treated as two different matrices, one acting on the parameter of the function, the other on the value. For both $\mathbf{L}$ and $\mathbf{L}^T$ we identify the r and g, most dominant columns. A measure for the significance of a column vector $v \in \mathbb{R}^n$ is its 2-norm $\|v\|_2$.

[3] This means, the matrix has exactly one non-zero entry per row at most one non-zero per column.

As the columns of $\mathbf{L}$ are ordered according to the singular values, we will pick the first r columns in this case. Now $\mathbf{L}$ and $\mathbf{L}^T$ are approximated by matrices that agree with the respective matrix in the selected r and g selected columns but have the $n-r$ and $n-g$, respectively, remaining columns set to $\mathbf{0} \in \mathbb{R}^n$. This can be expressed with the help of selection matrices $\mathbf{P}_r \in \{0,1\}^{r\times n}$ and $\mathbf{P}_g \in \{0,1\}^{g\times n}$, respectively:

$$\mathbf{L} \approx \mathbf{L}\mathbf{P}_r^T\mathbf{P}_r \quad \text{and} \quad \mathbf{L}^T \approx \mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g. \tag{6.34}$$

We may conclude $\mathbf{L}^T \approx \mathbf{P}_r^T\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g$, insert these approximations in (6.33) and multiply with $\mathbf{P}_r$, bearing in mind that $\mathbf{P}_r\mathbf{P}_r^T = \mathbf{I}_{r\times r}$:

$$\frac{d}{dt}\left[\mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{q}(\mathbf{L}\mathbf{P}_r^T\mathbf{P}_r\tilde{\mathbf{w}})\right] + \mathbf{P}_r\mathbf{L}^T\mathbf{P}_g^T\mathbf{P}_g\mathbf{j}(\mathbf{L}\mathbf{P}_r^T\mathbf{P}_r\tilde{\mathbf{w}}) + \mathbf{P}_r^T\mathbf{L}^T\mathbf{B}\mathbf{u} = \mathbf{0}. \tag{6.35}$$

Note that due to the approximations to $\mathbf{L}$ and $\mathbf{L}^T$ in the above equation $\mathbf{w}$ has changed to $\tilde{\mathbf{w}}$ which can merely be an approximation to the former. We introduce $\boldsymbol{\Sigma}_r = \operatorname{diag}(\sigma_1,\ldots,\sigma_r)$ and let $\mathbf{V} \in \mathbb{R}^{n\times r}$ be the first r columns of $\mathbf{u}$. In this wa we have $\mathbf{L}\mathbf{P}_r^T = \mathbf{V}\mathbf{S}_r$. Finally we scale (6.35) with $\boldsymbol{\Sigma}_r^{-1}$ and introduce a new unknown $\mathbf{z} = \boldsymbol{\Sigma}_r\mathbf{P}_r\tilde{\mathbf{w}} \in \mathbb{R}^r$ from which we can reconstruct the full state by approximation $\mathbf{x} \approx \mathbf{V}\mathbf{z}$. We end up with

$$\frac{d}{dt}\left[\mathbf{W}_{r,g}\bar{\mathbf{q}}(\mathbf{V}\mathbf{z})\right] + \mathbf{W}_{r,g}\bar{\mathbf{j}}(\mathbf{V}\mathbf{z}) + \tilde{\mathbf{B}}\mathbf{u}(t) = \mathbf{0}, \tag{6.36}$$

with $\bar{\mathbf{q}}(\mathbf{V}\mathbf{z}) = \mathbf{P}_g\mathbf{q}(\mathbf{V}\mathbf{z})$, $\bar{\mathbf{j}}(\mathbf{V}\mathbf{z}) = \mathbf{P}_g\mathbf{j}(\mathbf{V}\mathbf{z})$, $\mathbf{W}_{r,g} = \mathbf{V}^T\mathbf{P}_g^T \in \mathbb{R}^{r\times g}$ and $\tilde{\mathbf{B}} = \mathbf{V}^T\mathbf{B}$.

Here $\mathbf{P}_g$ has the same effect as noted in the previous subsection: not the full nonlinear functions $\mathbf{q}$ and $\mathbf{j}$ have to be evaluated but g components only.

6.1.3.5 Discrete Empirical Interpolation

Recently, Chaturantabut and Sorensen [12, 13] did present the Discrete Empirical Interpolation Method (DEIM) as a further modification of POD. It originates from partial differential equations (PDEs) where the nonlinearities exhibit a special structure. It can, however, be applied to general nonlinearities as well. We give a brief introduction of how this may look like in circuit simulation problems.

Given a nonlinear function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$, the essential idea of DEIM is to approximate $\mathbf{f}(\mathbf{x})$ by projecting it on a subspace, spanned by the basis $\{\mathbf{u}_1,\ldots,\mathbf{u}_g\} \subset \mathbb{R}^n$:

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{U}\mathbf{c}(\mathbf{x}), \tag{6.37}$$

where $\mathbf{U} = (\mathbf{u}_1,\ldots,\mathbf{u}_g) \in \mathbb{R}^{n\times g}$ and $\mathbf{c}(\mathbf{x}) \in \mathbb{R}^g$ is the coefficient vector. Forcing equality in (6.37) would state an overdetermined system for the $g < n$ coefficients

$\mathbf{c}t(\mathbf{x})$. Instead accordance in g rows is required, which can be expressed by

$$\mathbf{P}_g\mathbf{f}(\mathbf{x}) = (\mathbf{P}_g\mathbf{U})\mathbf{c}(\mathbf{x}), \tag{6.38}$$

with a selection matrix $P_g \in \{0,1\}^{g\times n}$. If $\mathbf{P}_g\mathbf{U}$ is non-singular, (6.38) has a unique solution $\mathbf{c}(\mathbf{x})$ and, hence $\mathbf{f}(\mathbf{x})$ can be approximated by

$$\mathbf{f}(\mathbf{x}) \approx \mathbf{U}\left(\mathbf{P}_g\mathbf{U}\right)^{-1}\mathbf{P}_g\mathbf{f}(\mathbf{x}), \tag{6.39}$$

which means that $\mathbf{f}(\mathbf{x})$ is interpolated at the entries specified by $\mathbf{P}_g$.

In (6.39), $\mathbf{U}\left(\mathbf{P}_g\mathbf{U}\right)^{-1}$ can be computed in advance, and, again, the multiplication $\mathbf{P}_g\mathbf{f}(\mathbf{x})$ corresponds to evaluating only those entries of $\mathbf{f}$, addressed by $\mathbf{P}_g$.

Using the notations introduced before, POD with the DEIM modification yields a reduced model (6.2) with

$$\hat{\mathbf{q}}(\mathbf{z}(t)) = \hat{\mathbf{W}}\,\bar{\mathbf{q}}(\mathbf{z}(t)), \quad \hat{\mathbf{j}}(\mathbf{z}(t)) = \hat{\mathbf{W}}\,\bar{\mathbf{j}}(\mathbf{z}(t)), \quad \hat{\mathbf{B}} = \mathbf{V}^T\mathbf{B}, \quad \hat{\mathbf{C}} = \mathbf{C}\mathbf{V}^T, \tag{6.40}$$

with $\hat{\mathbf{W}} = \mathbf{V}^T\mathbf{U}\left(\mathbf{P}_g\mathbf{U}\right)^{-1}$ and $\bar{\mathbf{q}}(\cdot) = \mathbf{P}_g\mathbf{q}(\cdot)$ and $\bar{\mathbf{j}}(\cdot) = \mathbf{P}_g\mathbf{j}(\cdot)$. Here POD provides the state-space part of the reduction, i.e., $\mathbf{V}$. And DEIM determines the subspace on which $\mathbf{q}$ and $\mathbf{j}$ is projected, hence the columns of the matrix $\mathbf{U} \in \mathbb{R}^{n\times g}$ and the selection $\mathbf{P}_g$.

The reduced subspace, suitable for representing a nonlinear function $\mathbf{f}$ on, is constructed from an SVD on a matrix $\mathbf{F} = (\mathbf{f}(\mathbf{x}_1),\ldots,\mathbf{f}(\mathbf{x}_K)) \in \mathbb{R}^{n\times K}$ whose columns are snapshots of the function evaluations. The matrix $\mathbf{U}$ in (6.39) consists then of the g most dominant left singular vectors of $\mathbf{F}$.

The core of DEIM is the construction of the selection $\mathbf{P}_g \in \{0,1\}^{g\times n}$. A set of indices $\{\rho_1,\ldots,\rho_g\} \subset \{1,\ldots,n\}$, determined by the DEIM-algorithm, define the selection matrix, meaning that $\mathbf{P}_g$ has a 1 in the ith row and ρ_ith column (for $i = 1,\ldots,g$) and 0 elsewhere.

The first index, ρ_1 is chosen to be the index of the largest (in absolute value) entry in $\mathbf{u}_1$. In step $l = 2,\ldots,g$ the residual

$$\mathbf{r}_{l+1} = \mathbf{U}_{l+1} - \mathbf{U}_l\left(\mathbf{P}_l\mathbf{U}_l\right)^{-1}\mathbf{P}_l\mathbf{U}_{l+1}$$

is computed where $\mathbf{U}_l = (\mathbf{u}_1,\ldots,\mathbf{u}_l)$ and $\mathbf{P}_l \in \{0,1\}^{l\times n}$ is constructed from the indices $\rho_1,\ldots,\rho_l$ (cp. (6.39)). Then, the index corresponding to entry of the residual $\mathbf{r}_{l+1}$ the largest magnitude of is taken as index ρ_{l+1}.

Setting up the selection matrix with this algorithm, $\mathbf{P}_g\mathbf{U}$ in (6.37) is guaranteed to be regular. For a detailed description and discussion, including error estimates we refer to [12, 13].

Note: Originally, DEIM is constructed in the context of discretisation and approximation of PDEs with a special structure of the nonlinearity involved. Considering network problem (6.1a) that leads to the reduced problem (6.40), we constructed a uniform DEIM-approximation, i.e., $\mathbf{U}$ and $\mathbf{P}_g$ for the both nonlinearities, $\mathbf{q}$ and $\mathbf{j}$ involved. This could probably be approached in a different way, too.

6.1.4 Other Approaches

We shortly address some other approaches. In [5, 6, 8, 48] Krylov-subspace methods are applied to bilinear and quadratic-bilinear ODE-systems. One exploits the observation that several nonlinear functions can be generated by extending the system first with additional unknowns for which simple differential equations are introduced. In [48] also the application to DAEs is discussed. In [22] a transformation from a set of nonlinear differential equations to another set of equivalent nonlinear differential equations that involve only quadratic terms of state variables is described to which Volterra analysis is applied to derive a reduced model.

We already mentioned [20] for nonlinear balancing in which the energy functions arise from solving Hamilton-Jacobi differential equations. Related work is on cross Gramians for dissipative and symmetric nonlinear systems [25, 26].

In [37, 38] interpolating input-output behavior of nonlinear systems is studied. This is related to table modelling.

6.1.5 Numerical Experiments

For testing purposes, a time-simulator, has been implemented in `octave`. The underlying DAE integration scheme used here is CHORAL [23], a Rosenbrock-Wanner type of method, adapted to circuitry problems. Besides performing transient-analysis, TPWL and POD models can be extracted and reused in simulations.

To show the performance of TPWL and POD when applied to an example from circuit design, the nonlinear transmission line in Fig. 6.2, taken from [33] is chosen. Only the diodes introduce the designated nonlinearity to the circuit, as the current $\imath_d$ traversing a diode is modeled by $\imath_d(v) = \exp(40 \cdot v) - 1$ where v is the voltage drop between the diode's terminals. The resistors and capacitors contained in the model have unit resistance and capacitance $(R = C = 1)$, respectively. The current

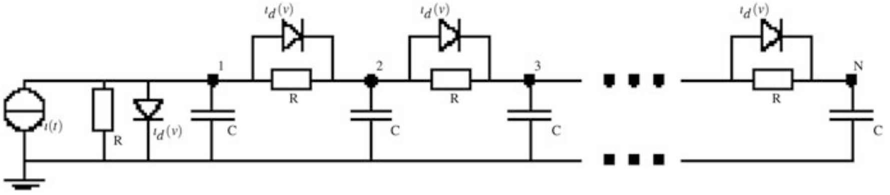


Fig. 6.2 Nonlinear transmission line

source between node 1 and ground marks the input to the system $u(t) = \imath(t)$ and the output of the system is chosen to be the voltage at node 1: $y(t) = v_1(t)$.

Introducing the state vector $\mathbf{x}(t) = (v_1(t), \ldots, v_N(t))^T \in \mathbb{R}^N$, where $v_i(t)$ describes the voltage at node $i \in \{1, \ldots, N\}$ modified nodal analysis yields:

$$\begin{aligned} \frac{d}{dt}\mathbf{x}(t) + \mathbf{j}(\mathbf{x}(t)) + \mathbf{B}u(t) = \mathbf{0} \\ y(t) = \mathbf{C}\mathbf{x}(t), \end{aligned} \tag{6.41}$$

where $\mathbf{B} = \mathbf{C}^T = (1, 0, \ldots, 0)^T \in \mathbb{R}^N$ and $\mathbf{j} : \mathbb{R}^N \to \mathbb{R}^N$ with

$$\mathbf{j}(\mathbf{x}) = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix} \cdot \mathbf{x} - \begin{pmatrix} 2 - e^{40x_1} - e^{40(x_1 - x_2)} \\ e^{40(x_1 - x_2)} - e^{40(x_2 - x_3)} \\ \vdots \\ e^{40(x_{N-2} - x_{N-1})} - e^{40(x_{N-1} - x_N)} \\ e^{40(x_{N-1} - x_N)} - 1 \end{pmatrix}$$

We choose $N = 100$, causing a problem of dimension $n = 100$.

For extracting a model a shifted Heaviside function was used as training input. Resimulation was done both with the training input and with a cosine function on the interval $[t_{\text{start}}, t_{\text{end}}] = [0, 10]$:

$$u_{\text{train}}(t) = H(t-3) = \begin{cases} 0 & t < 3 \\ 1 & t \geq 3 \end{cases} \qquad u_{\text{resim}}(t) = \frac{1}{2}\left(1 + \cos\left(\frac{2\pi}{10}t\right)\right).$$

The TPWL-model was extracted with the Arnoldi-method as suggested in [33], leading to a order reduced model of dimension 10. For choosing linearization points, the strategy proposed by Rewieński with $\alpha = 0.0167$ in (6.14) has been tested. With this setting, 27 linear models are constructed. Also the extended strategy described in Voß [49] is implemented, but does not show much different results for the transmission line. A more detailed discussion on the model extraction and statistics on which models are chosen can be found in [35, 36].

For the transmission line, also a POD model as well as a POD model that has been modified with the Discrete Empirical Interpolation Method (DEIM) algorithm is constructed. By choosing $d = 99.9$ in (6.30) a reduced model of dimension 4 is constructed. Applying the DEIM algorithm the nonlinear $\mathbf{q}$ and $\mathbf{j}$ where reduced to order 5. Figure 6.3 displays the singular values form snapshots collected during a training run and the behaviour of the coverage function (6.30). Note, that only 38 singular values are shown, although the full system is of dimension 100. This is caused by the time domain simulation: with tolerances specified for the timestepping mechanism, only 38 time steps where necessary to resolve the system. However, also with more snapshots, the gradient of the singular values does not change remarkably.

Figures 6.4 and 6.5 show the trajectories, i.e., the behaviour in time, of the voltages at nodes 1 and ten, when the training signal is and when the cosine like

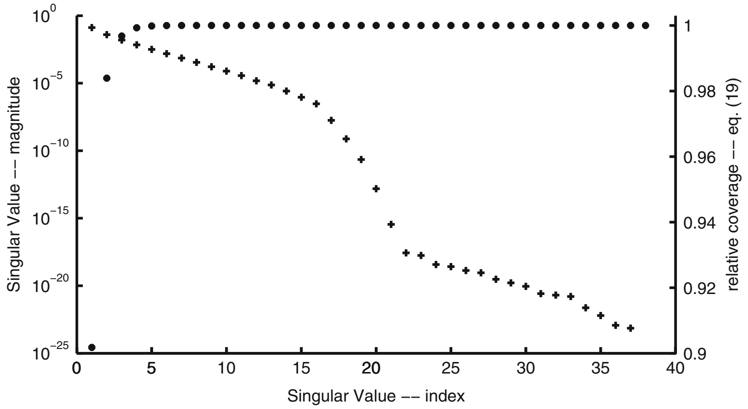


Fig. 6.3 Transmission line: singular values (+) & coverage (∗)

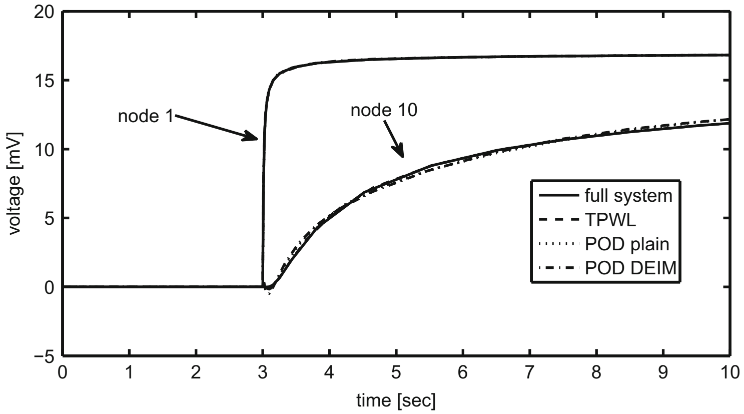


Fig. 6.4 Nonlinear transmission line: resimulation results

signal is applied at the input, respectively. The plots show the signals reproduced by using the full model, the TPWL-model and the plain POD and DEIM-adapted POD model. Slight deviations from the reference solution are obvious, but, in total, a good matching is observable. However, the TPWL-model seems to have problems following the reference solution, when a signal, different to the input is applied. This indicates that there are still improvements possible.

Finally, Table 6.1 gathers the performance of the models, measured in time consumption. Clearly, simulation with the TPWL model is cheaper than using the full network as not the full nonlinearity has to be evaluated. Still, POD, adapted with DEIM is superior, as no decision has to be made, which model to use. Furthermore, as predicted in Sect. 6.1.1 applying only projection without taking care of the nonlinearity, does not guarantee cheaper to evaluate model: the plain POD model, used for simulation, causes equal or even increased computational expenses.

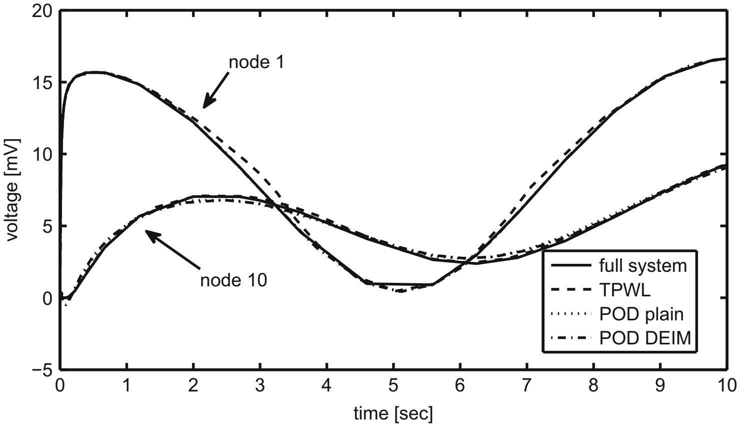


Fig. 6.5 Transmission line: different input

Table 6.1 Transmission line: performance of nonlinear MOR techniques

	Resimulation (s)	Changed input (s)
Full problem	6.67	4.66
TPWL model	4.35	3.47
POD model	6.51	5.23
POD-DEIM model	1.98	1.63

6.2 Model Order Reduction for Multi-terminal Circuits

Analysis of effects due to parasitics is of vital importance during the design of large-scale integrated circuits and derived products.[4] One way to model parasitics is by means of parasitic extraction, which results in large linear $RCL(k)$ networks. In ESD analysis [65, 75], for instance, the interconnect network is modeled by resistors with resistances that are based on the metal properties. In other (RF) applications one needs RC or even $RCLk$ extractions to deal accurately with higher frequencies as well.

The resulting parasitic networks may contain up to millions of resistors, capacitors, and inductors, and hundreds of thousands of internal nodes, and thousands of external nodes (nodes with connections to active elements such as transistors). Simulation of such large networks within reasonable time is often not possible [62, 63], and including such networks in full system simulations may be even unfeasible. Hence, there is need for much smaller networks that accurately or even exactly describe the behavior of the original network, but allow for fast analysis.

[4] Section 6.2 has been written by Roxana Ionutiu and Joost Rommes. For an extended treatment on the topics of this section see also the Ph.D. Thesis of the first author [68].

In this section we describe recently developed methods for the reduction of large R networks, and present a new approach for the reduction of large RC networks. We show how insights from graph theory, numerical linear algebra, and matrix reordering algorithms can be used to construct a reduced network that shows sparsity preservation especially for circuits with multi-terminals (ports). Hence it allows for the same number of external nodes, but needs much fewer internal nodes and circuit elements (resistors and capacitors). Circuit synthesis is applied after model reduction, and the resulting reduced netlists are tested with industrial circuit simulators. For related literature we refer to [55–57].

The section is organized as follows. Section 6.2.1 revisits recent work on reduction of R networks [83, 84]. It provides the basis for understanding how graph theoretical tools can be used to significantly improve the sparsity of the reduced models, which are later synthesized [70] into reduced netlists. Section 6.2.2 deals with the reduction of RC networks. Section 6.2.2.1 first reviews an existing method which employs *Pole Analysis* via *Congruence Transformations* (*PACT*) [73] to reduce RC netlists with multi-terminals. In Sect. 6.2.2.2 the new method *Sparse Modal Approximation (SparseMA)* is presented, where graph-theoretical tools are brought in to enhance sparsity preservation for the reduced models. The numerical results for both R and RC netlist reduction are presented in Sect. 6.2.3. Section 6.2.4 concludes.

6.2.1 *Reduction of R Networks*

In this section we review the approach for reducing R networks, as developed in [83, 84]. Reduction of R networks, i.e., networks that consist of resistors only, is needed in electro-static discharge analysis (ESD), where large extracted R networks are used to model the interconnect. Accurate modeling of interconnect is required here, since the costs involved may vary from a few cents to millions if, due to interconnect failures, a respin of the chip is needed. An example of a damaged piece of interconnect that was too small to conduct the amount of current is shown in Fig. 6.6.

6.2.1.1 Circuit Equations and Matrices

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

$$\begin{bmatrix} R & P \\ -P^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{i}_b \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{i}_n \end{bmatrix}, \tag{6.42}$$

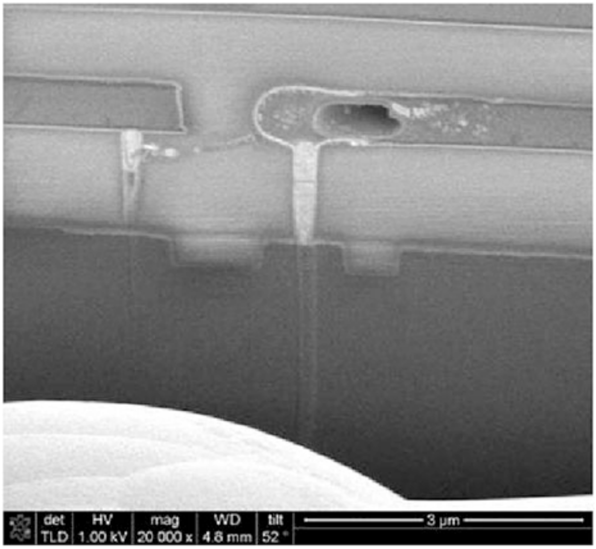


Fig. 6.6 Example of a piece of interconnect that was damaged because it was too small to conduct the amount of current caused by a peak charge

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

The MNA (modified nodal analysis) formulation [60, 76] can be derived from (6.42) by eliminating the resistor currents $\mathbf{i}_b = -R^{-1}P\mathbf{v}$:

$$G\mathbf{v} = \mathbf{i}_n, \tag{6.43}$$

where $G = P^T R^{-1} P \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite. Since currents can only be injected in external nodes, and not in internal nodes of the network, system (6.43) has the following structure:

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{12}^T & G_{22} \end{bmatrix} \begin{bmatrix} \mathbf{v}_e \\ \mathbf{v}_i \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{i}_e, \tag{6.44}$$

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

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

Finally, systems (6.42)–(6.44) must be made consistent by grounding a node gnd, i.e., setting $\mathbf{v}(gnd) = 0$ and removing the corresponding equations. In the following we will still use the notation G for the grounded system matrix, if this does not lead to confusion.

6.2.1.2 Problem Formulation

The problem is: given a very large resistor network described by (6.42), find an equivalent network with (a) the same external nodes, (b) exactly the same path resistances between external nodes, (c) $\hat{n} \ll n$ internal nodes, and (d) $\hat{r} \ll r$ resistors. Additionally, (e) the reduced network must be realizable as a netlist so that it can be (re)used in the design flow as subcircuit of large systems.

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

6.2.1.3 Existing Approaches

There are several approaches to deal with large resistor networks. In some cases the need for an equivalent reduced network can be circumvented in some way: due to sparsity of the original network, memory usage and computational complexity are *in principle* not an issue, since solving linear systems with the related conductance matrices is typically of complexity $O(n^{\alpha})$, where $1 < \alpha \leq 2$, instead of the traditional $O(n^3)$ [79]. Of course, α depends on the sparsity and will rapidly increase as sparsity decreases. This also explains why eliminating all internal nodes does not work in practice: the large reduction in unknowns is easily undone by the enormous increase in number of resistors, mutually connecting all external nodes.

However, if we want to (re)use the network in full system simulations, a reduced equivalent network is needed to limit simulation times or make simulation possible at all. In [77] approaches based on large-scale graph partitioning packages such as (h)METIS [72] are described, but only applied to small networks. Structure preserving projection methods for model reduction [66, 86], finally, have the disadvantage that they lead to dense reduced-order models if the number of terminals is large. There is commercial software [59, 64] available for the reduction of parasitic reduction networks.

6.2.1.4 Improved Approach

Knowing that eliminating all internal nodes is not an option and that projection methods lead to dense reduced-order models, we use concepts from matrix reordering algorithms such as AMD [54] and BBBD [88], usually used as preprocessing step for (parallel) LU- or Cholesky-factorization, to determine which nodes to eliminate. The fill-in reducing properties of these methods also guarantee sparsity of the reduced network. Similar ideas have also been used in [77, 89].

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

Our approach can be summarized as follows:

1. The first step is to compute the strongly connected components [61] of the network. The presence of strongly connected components is very natural in extracted networks: a piece of interconnect connecting two other elements such as diodes or transistors, for instance, results in an extracted network with two terminals, disconnected from the rest of the extracted circuit. By splitting the network into connected components, we have simplified the problem of reduction because we can deal with the connected components one by one.
2. The second step is to selectively eliminate internal nodes in the individual connected components. For resistor networks, this can be done using the Schur complement [67], and no approximation error is made. The key here is that those internal nodes are eliminated that give the least fill-in. First, (Constrained) AMD [62] is used to reorder the unknowns such that the terminal nodes will be among the last to eliminate. To find the optimal reduction, internal nodes are eliminated one-by-one in the order computed by AMD, while keeping track of the reduced system with fewest resistors.

   Since the ordering is chosen to minimize fill-in, the resulting reduced matrix is sparse. Note that all operations are exact, i.e., we do not make any approximations. As a result, the path resistances between external nodes remain equal to the path resistances in the original network.
3. Finally, the reduced conductance matrix can be realized as a reduced resistor network that is equivalent to the original network. This is done easily by unstamping the values in the G matrix intro the corresponding resistor values and their node connections in the netlist [69]. Since the number of resistors (and number of nodes) is smaller than in the original network, also the resulting netlist is smaller in size.

An additional reduction could be obtained by removing relatively large resistors from the resulting reduced network. However, this will introduce an approximation error that might be hard to control a priori, since no sharp upper bounds on the error are available [87]. Another issue that is subject to further research is that the optimal ratio of number of (internal) nodes to resistors (sparsity) may also depend on the ratio of number of external to internal nodes, and on the type of simulation that will be done with the network.

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

6.2.2 Reduction of RC Networks

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

Following the problem description in [73], consider the modified nodal analysis (MNA) description of an input *impedance* type RC circuit, driven by input currents:

$$(\mathbf{G} + s\mathbf{C})\mathbf{x}(s) = \mathbf{B}\mathbf{u}(s), \tag{6.45}$$

where $\mathbf{x}$ denote the node voltages, and $\mathbf{u}$ represent the currents injected into the terminals (also called ports or external nodes). The number of internal nodes is n, and the number of terminals is p, thus $\mathbf{G} \in \mathbb{R}^{(p+n)\times(p+n)}$, $\mathbf{C} \in \mathbb{R}^{(p+n)\times(p+n)}$ and $\mathbf{B} \in \mathbb{R}^{(p+n)\times p}$. A natural choice for the system outputs are the voltage drops at the terminal nodes, i. e., $\mathbf{y}(s) = \mathbf{B}^T\mathbf{x}(s)$. Thus the transfer function of (6.45) is the input impedance:

$$\mathbf{Z}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \mathbf{B}^T(\mathbf{G} + s\mathbf{C})^{-1}\mathbf{B}. \tag{6.46}$$

Modal approximation is a method to reduce (6.45), by preserving its most dominant eigenmodes. The dominant eigenmodes are a subset of the poles of $\mathbf{Z}(s)$ (i. e. of the generalized eigenvalues $\Lambda(-\mathbf{G}, \mathbf{C})$) and can be computed using specialized eigenvalue solvers (SADPA [80] or SAMDP [82, 85]). For the complete discussion on modal approximation and its implementation we refer to [80, 81, 85]. Here, we emphasize that applying modal approximation to reduce (6.45) directly is unsuitable especially if the underlying RC circuit has many terminals (inputs). This is because modal approximation does not preserve the structure of $\mathbf{B}$ and $\mathbf{B}^T$ during reduction (for ease of understanding we denote the input-output structure loss as *non-preservation of terminals*) [69]. Modeling the input-output connectivity of the reduced model would require synthesis via controlled sources at the circuit terminals, and furthermore would connect all terminals with one-another [69]. In this chapter we present several alternatives for reducing RC netlists where not only the terminals are preserved, but also the sparsity of the reduced models.

Grouping the node voltages so that $\mathbf{x}_P \in \mathbb{R}^p$ are the voltages measured at the terminal nodes (ports), and $\mathbf{x}_I \in \mathbb{R}^n$ are the voltages at the internal nodes, we can partition (6.45) as follows:

$$\left(\begin{bmatrix}\mathbf{G}_P & \mathbf{G}_C^T \\ \mathbf{G}_C & \mathbf{G}_I\end{bmatrix} + s\begin{bmatrix}\mathbf{C}_P & \mathbf{C}_C^T \\ \mathbf{C}_C & \mathbf{C}_I\end{bmatrix}\right)\begin{bmatrix}\mathbf{x}_P \\ \mathbf{x}_I\end{bmatrix} = \begin{bmatrix}\mathbf{B}_P \\ \mathbf{0}\end{bmatrix}\mathbf{u}. \tag{6.47}$$

Since no current is injected into internal nodes, the non-zero contribution from the input is $\mathbf{B}_P \in \mathbb{R}^{(p\times p)}$. Eliminating $\mathbf{x}_I$, system (6.47) is equivalent to:

$$\underbrace{[(\mathbf{G}_P + s\mathbf{C}_P)}_{\mathbf{Y}_P(s)} - \underbrace{(\mathbf{G}_C + s\mathbf{C}_C)^T(\mathbf{G}_I + s\mathbf{C}_I)^{-1}(\mathbf{G}_C + s\mathbf{C}_C)]}_{\mathbf{Y}_I(s)}\mathbf{x}_P = \mathbf{B}_P\mathbf{u} \tag{6.48}$$

$$\mathbf{Y}(s) = \mathbf{Y}_P(s) - \mathbf{Y}_I(s) \tag{6.49}$$

In (6.48) the matrix blocks $(\mathbf{G}_P + s\mathbf{C}_P)$ corresponding to the circuit terminals are isolated. Applying modal approximation on $\mathbf{Y}_I(s)$ would reduce the system and preserve the location of the terminals. This would involve for instance computing the dominant eigenmodes of $(-\mathbf{G}_I, \mathbf{C}_I)$ via a variant of SAMDP (called here *frequency dependent SAMDP*, because the input-output matrices $(\mathbf{G}_C + s\mathbf{C}_C)$ depend on the frequency s). We have implemented this approach, but it turns out that a large number of dominant eigenmodes of $(-\mathbf{G}_I, \mathbf{C}_I)$ would be needed to capture the DC and offset of the full system $\mathbf{Y}(s)$. Instead, two alternatives are presented that improve the quality of the approximation: an existing method called *PACT (Pole Analysis via Congruence Transformations)* [73] and a novel graph-based reduction called *SparseMA (Sparse Modal Approximation).*

6.2.2.1 Existing Method: PACT

In [73] the authors propose to capture the DC and offset of $\mathbf{Y}(s)$ via a congruence transformation which reveals the first two moments of $\mathbf{Y}(s)$ as follows. Since $\mathbf{G}_I$ is symmetric positive definite, the Cholesky factorization $\mathbf{L}\mathbf{L}^T = \mathbf{G}_I$ exists. Using the following congruence transformation:

$$\mathbf{X} = \begin{bmatrix}\mathbf{I} & \mathbf{0} \\ -\mathbf{G}_I^{-1}\mathbf{G}_C & \mathbf{L}^{-T}\end{bmatrix},\ \mathbf{G}' = \mathbf{X}^T\mathbf{G}\mathbf{X} = \begin{bmatrix}\mathbf{G}'_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I}\end{bmatrix},\ \mathbf{C}' = \mathbf{X}^T\mathbf{C}\mathbf{X} = \begin{bmatrix}\mathbf{C}'_P & \mathbf{C}'^T_C \\ \mathbf{C}'_C & \mathbf{C}'_I\end{bmatrix} \tag{6.50}$$

Eqs. (6.48) and (6.49) are rewritten as:

$$\underbrace{[(\mathbf{G}'_P + s\mathbf{C}'_P)}_{\mathbf{Y}'_P(s)} - \underbrace{s^2\mathbf{C}'^T_C(\mathbf{I} + s\mathbf{C}'_I)^{-1}\mathbf{C}'_C]}_{\mathbf{Y}'_I(s)}\mathbf{x}'_P = \mathbf{B}_P\mathbf{u} \tag{6.51}$$

$$\mathbf{Y}'(s) = \mathbf{Y}'_P(s) - \mathbf{Y}'_I(s), \tag{6.52}$$

where:

$$\mathbf{G}'_P = \mathbf{G}_P - \mathbf{G}_C^T\mathbf{M}, \quad \mathbf{M} = \mathbf{G}_I^{-1}\mathbf{G}_C \tag{6.53}$$

$$\mathbf{C}'_P = \mathbf{C}_P - \mathbf{N}^T\mathbf{M} - \mathbf{M}^T\mathbf{C}_C, \quad \mathbf{N} = \mathbf{C}_C - \mathbf{C}_I\mathbf{M} \tag{6.54}$$

$$\mathbf{C}'_C = \mathbf{L}^{-1}\mathbf{N}, \quad \mathbf{C}'_I = \mathbf{L}^{-1}\mathbf{C}_I\mathbf{L}^{-T}. \tag{6.55}$$

In (6.51), the term $\mathbf{Y}'_P(s)$ captures the first two moments of $\mathbf{Y}'(s)$ and is preserved in the reduced model. The reduction is performed on $\mathbf{Y}'_I(s)$ only. In [73] this is done via modal approximation as described next. Using the symmetric eigendecomposition $\mathbf{C}'_I = \mathbf{U}\Lambda'_I\mathbf{U}^T$, $\mathbf{U}^T\mathbf{U} = \mathbf{I}$, the system matrices (6.50) are block diagonalized as follows:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{U} \end{bmatrix}, \quad \mathbf{G}'' = \mathbf{X}'^T\mathbf{G}'\mathbf{X}' = \begin{bmatrix} \mathbf{G}'_P & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \mathbf{G}' \tag{6.56}$$

$$\mathbf{C}'' = \mathbf{X}'^T\mathbf{C}'\mathbf{X}' = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}'^T_C\mathbf{U} \\ \mathbf{U}^T\mathbf{C}'_C & \mathbf{U}^T\mathbf{C}'_I\mathbf{U} \end{bmatrix} = \begin{bmatrix} \mathbf{C}'_P & \mathbf{C}''^T_C \\ \mathbf{C}''_C & \Lambda'_I \end{bmatrix} \tag{6.57}$$

$$\mathbf{Y}''(s) = \mathbf{Y}'_P(s) - s^2[\mathbf{C}''^T_C(\mathbf{I} + s\Lambda'_I)^{-1}\mathbf{C}''_C] \tag{6.58}$$

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

$$\mathbf{Y}''_k(s) = \mathbf{Y}'_P(s) - s^2\sum_{i=1}^{k}\frac{\mathbf{r}_i^T\mathbf{r}_i}{1 + s\lambda'_i}, \quad \mathbf{r}_i^T = \mathbf{C}'^T_C\mathbf{U}_{[:,1:k]}, \quad \lambda'_i = \Lambda'_{I\,[i,i]}. \tag{6.59}$$

In [73], a selection criterion for λ'_i, $i = 1 \ldots k$ is proposed, based on a user-specified error and a maximum frequency. These eigenmodes are computed in [73] via the Lanczos algorithm. The criterion proposed in [81, 85] can also be used to compute the dominant eigenmodes λ'_i via SAMDP.

The advantage of the PACT reduction method is the preservation of the first two moments of $\mathbf{Y}(s)$ in $\mathbf{Y}'_P(s)$. This ensures that the DC and offset of the response is approximated well in the reduced model. The main costs of such an approach are: (1) performing a Cholesky factorization of $\mathbf{C}_I$ (which becomes expensive when n is very large, (2) solving an eigenvalue problem from a dense $\mathbf{C}'_I$ matrix and, most importantly, (3) the fill-in in the port block matrices $\mathbf{G}'_P$, $\mathbf{C}'_P$ and in $\mathbf{C}'_C$. It turns out that (2) can be solved more efficiently by keeping $\mathbf{C}'_I$ as a product of sparse matrices during computation, and will be addressed elsewhere. Avoiding problems (1) and (3) however require new strategies to improve sparsity, and are presented in Sect. 6.2.2.2. The fill-in introduced in $\mathbf{G}'_P$, $\mathbf{C}'_P$ becomes especially important for RC netlists with many terminals [$p \sim O(10^3)$]. Compared to the original model where the port blocks $\mathbf{G}_P$ and $\mathbf{C}_P$ were sparse, the dense $\mathbf{G}'_P$, $\mathbf{C}'_P$ will yield many R and C components during synthesis, resulting in a reduced netlist where almost all the nodes are interconnected. Simulating such netlists might require longer time measures than the original circuit simulation, hence sparser reduced models (and netlists) are desired. Next, we present several ideas for improving the sparsity of RC reduced models via a combination of tools including: netlist partitioning, graph-based node reordering strategies, and efficient algorithms for modal approximation.

6.2.2.2 Improved Graph-Based Method: SparseMA

In this section we present an improved model reduction method for RC circuits, which overcomes the disadvantages of PACT: it requires no matrix factorizations prior to reduction, performs all numerical computations on sparse matrices, and most importantly, preserves the sparsity of the matrix blocks corresponding to the external nodes. The method is called *sparse modal approximation (SparseMA)* and uses tools from graph theory to identify a partitioning and reordering of nodes that, when applied prior to the model reduction step, can significantly improve the sparsity of the reduced model.

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

$$\left(\begin{bmatrix}\mathbf{G}_R & \mathbf{G}_K\\ \mathbf{G}_K^T & \mathbf{G}_S\end{bmatrix} + s\begin{bmatrix}\mathbf{C}_R & \mathbf{C}_K\\ \mathbf{C}_K^T & \mathbf{C}_S\end{bmatrix}\right)\begin{bmatrix}\mathbf{x}_R\\ \mathbf{x}_S\end{bmatrix} = \begin{bmatrix}\mathbf{0}\\ \mathbf{B}_S\end{bmatrix}\mathbf{u}. \tag{6.60}$$

Note that in $\mathbf{B}_S$ the rows corresponding to the promoted m internal nodes are still zero. Similarly to (6.48), the admittance is expressed as:

$$\underbrace{[(\mathbf{G}_S + s\mathbf{C}_S)}_{\mathbf{Y}_S(s)} - \underbrace{(\mathbf{G}_K + s\mathbf{C}_K)^T(\mathbf{G}_R + s\mathbf{C}_R)^{-1}(\mathbf{G}_K + s\mathbf{C}_K)]}_{\mathbf{Y}_R(s)}\mathbf{x}_S = \mathbf{B}_S\mathbf{u} \tag{6.61}$$

$$\mathbf{Y}(s) = \mathbf{Y}_S(s) - \mathbf{Y}_R(s). \tag{6.62}$$

Recall that reducing $\mathbf{Y}_I(s)$ directly from the simple partitioning (6.47) and (6.48) is not a method of choice, because by preserving $\mathbf{Y}_P(s)$ only, the DC and offset of $\mathbf{Y}(s)$ would not be accurately matched. Using instead the improved partitioning (6.60) and (6.61), one aims at better approximating the DC and offset of $\mathbf{Y}(s)$ by preserving $\mathbf{Y}_S(s)$ (which now encaptures not only the external nodes but also a subset of the internal nodes). Finding the partitioning (6.60) only requires a reordering of nodes, thus no Cholesky factorization or fill-introducing congruence transformation is needed prior to the MOR step. One can reduce $\mathbf{Y}_R(s)$ directly with modal approximation (via frequency dependent SAMDP), and preserve the sparsity of the extended port blocks from $\mathbf{Y}_S(s)$.

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

$$\mathbf{Y}_k(s) = \mathbf{Y}_S(s) - \sum_{i=1}^{k} \frac{\mathbf{q}_i^T \mathbf{q}_i}{1 + s\lambda_i} \ , \ \mathbf{q}_i^T = (\mathbf{G}_K + s\mathbf{C}_K)^T \mathbf{V}_{[:,1:k]}, \ \ \lambda_i = \Lambda_{R[i,i]}. \tag{6.63}$$

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

$$\left(\begin{bmatrix} \hat{\mathbf{G}}_R & \hat{\mathbf{G}}_K \\ \hat{\mathbf{G}}_K^T & \mathbf{G}_S \end{bmatrix} + s \begin{bmatrix} \hat{\mathbf{C}}_R & \hat{\mathbf{C}}_K \\ \hat{\mathbf{C}}_K^T & \mathbf{C}_S \end{bmatrix} \right) \begin{bmatrix} \hat{\mathbf{x}}_R \\ \mathbf{x}_S \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_S \end{bmatrix} \mathbf{u}, \tag{6.64}$$

where:

$$\hat{\mathbf{G}}_R = \mathbf{V}_{[:,1:k]}^T \mathbf{G}_R \mathbf{V}_{[:,1:k]} \to \text{diagonal}, \ \ \hat{\mathbf{G}}_K = \mathbf{V}_{[:,1:k]}^T \mathbf{G}_K, \ \ \mathbf{G}_S \to \text{sparse} \tag{6.65}$$

$$\hat{\mathbf{C}}_R = \mathbf{V}_{[:,1:k]}^T \mathbf{C}_R \mathbf{V}_{[:,1:k]} \to \text{diagonal}, \ \ \hat{\mathbf{C}}_K = \mathbf{V}_{[:,1:k]}^T \mathbf{C}_K, \ \ \mathbf{C}_S \to \text{sparse}. \tag{6.66}$$

The remaining problem is how to determine the selected nodes and the partitioning (6.60). Inspired from the results obtained for R networks, we propose to first find the permutation $\mathbf{P}$ which identifies the strongly connected components (sccs) of $\mathbf{G}$. Both $\mathbf{G}$ and $\mathbf{C}$ are reordered according to $\mathbf{P}$, revealing the structure (6.60). With this permutation, the circuit terminals are redistributed according to the sccs of $\mathbf{G}$, and several clusters of nodes can be identified: a large component consisting of internal nodes and very few (or no) terminals, and clusters formed each by internal nodes plus some terminals. We propose to leave all clusters consisting of internal nodes and terminals intact, and denote these nodes as the *selected nodes* mentioned above. If there are still terminals outside these clusters, they are added to these selected nodes and complete the blocks $\mathbf{G}_S$, $\mathbf{C}_S$. The remaining cluster of internal nodes forms $\mathbf{G}_R$ and $\mathbf{C}_R$. The model reduction step is performed on $\mathbf{G}_R$ and $\mathbf{C}_R$ (and implicitly on $\mathbf{G}_K$ and $\mathbf{C}_K$). We also note that matrices $\mathbf{G}_K$ and $\mathbf{C}_k$ resulting from this partitioning usually have many zero columns, thus $\hat{\mathbf{G}}_K$ and $\hat{\mathbf{C}}_K$ will preserve these zero columns.

The procedure is illustrated in Sect. 6.2.3 through a medium-sized example. Larger netlists can be treated via a similar reordering and partitioning strategy, possibly in a recursive manner (for instance when after an initial reordering the number of selected nodes is too large, the same partitioning strategy could be re-applied to $\mathbf{G}_S$ and $\mathbf{C}_S$ and further reduce these blocks). Certainly, other reorderings of **G** and **C** could be exploited, for instance according to a permutation which identifies the sccs of **C** instead of **G**. The choice for either using **G** or **C** to determine the permutation **P** is made according to the structure of the underlying system and may depend on the application. We also emphasize that the reduced models for both PACT and SparseMA are passive [74] and therefore also stable. Passivity is ensured by the fact that all transformations applied throughout are congruence transformations on symmetric positive definite matrices, thus the reduced system matrices remain symmetric positive definite.

6.2.3 Numerical Results

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

6.2.3.1 *R* Network Reduction

Table 6.2 shows results for three resistor networks of realistic interconnect layouts. The number of nodes is reduced by a factor > 10 and the number of resistors by a factor > 3. As a result, the computing time for calculating path resistances in the original network (including nonlinear elements such as diodes) is 10 times smaller.

6.2.3.2 *RC* Network Reduction

We reduce an RC netlist with $n = 3{,}231$ internal nodes and $p = 22$ terminals (external nodes). The structure of the original **G** and **C** matrices is shown in Figs. 6.7 and 6.8, where the $p = 22$ terminals correspond to their first 22 rows and columns.

The permutation revealing the strongly connected components of **G** reorders the matrices as shown in Figs. 6.9 and 6.10. The reordering is especially visible in the "arrow-form" capacitance matrix. There, the $p = 22$ terminal nodes together with

Table 6.2 Results of reduction algorithm

	Network I		Network II		Network III	
	Original	Reduced	Original	Reduced	Original	Reduced
#external nodes	274		3,399		1,978	
#internal nodes	5,558	516	99,112	6,012	101,571	1,902
#resistors	8,997	1,505	161,183	62,685	164,213	39,011
CPU time	10 s	1 s	67 h	7 h	20 h	2 h
Speed up	10×		9.5×		10×	

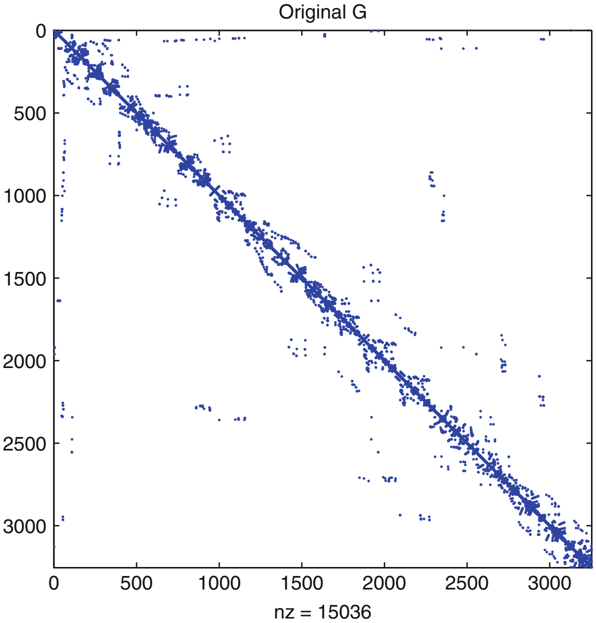


Fig. 6.7 Original **G** matrix

$m = 40$ internal nodes are promoted to the border, revealing the 62 selected nodes that will be preserved in the reduced model (i.e. the $\mathbf{G}_S$ and $\mathbf{C}_S$ blocks in (6.60)). The first $n - m = 3{,}191$ nodes are the remaining internal nodes and form the $\mathbf{G}_R$ and $\mathbf{C}_R$ blocks in (6.60). The $\mathbf{G}_K$ block has only 1 non-zero column, and also in $\mathbf{C}_K$ many zero columns can be identified.

The reduced SparseMA model is obtained according to (6.63) and (6.64) and is shown in Figs. 6.11 and 6.12. The internal blocks $\mathbf{G}_R$ and $\mathbf{C}_R$ were reduced from dimension 3,191 to $\hat{\mathbf{G}}_R$ and $\hat{\mathbf{C}}_R$ of dimension $k = 7$, by interpolating the 7 most dominant eigenmodes of $[\Lambda_R, \mathbf{V}] = eig(-\mathbf{G}_R, \mathbf{C}_R)$. Note that $\hat{\mathbf{G}}_R$ and $\hat{\mathbf{C}}_R$ are diagonal. The selected 62 nodes corresponding to the $\mathbf{G}_S$ and $\mathbf{C}_S$ blocks are preserved, evidently preserving sparsity. The only fill-in introduced by the proposed reduction procedure is in the non-zero columns of $\hat{\mathbf{G}}_K$ and $\hat{\mathbf{C}}_K$. It is worth noticing that $\hat{\mathbf{G}}_K$ only has 1 non-zero column, thus remains sparse.

The sparsity structure of the PACT reduced model (6.59) is shown in Figs. 6.13 and 6.14. The blocks corresponding to the first 22 nodes (the preserved external nodes) are full, as are the capacitive connection blocks to the reduced internal part. Only the reduced internal blocks remain sparse (diagonal).

Aside from sparsity preservation, one is interested in the quality of the approximation for the reduced model. In Fig. 6.15, we show that the SparseMA model accurately matches the original response for a wide frequency range (1 Hz → 10 THz). The Pstar [78] simulations of the synthesized model are identical to the Matlab simulations (the synthesized model was obtained via the RLCSYN unstamping procedure [71, 87]). In Fig. 6.16, the relative errors between the

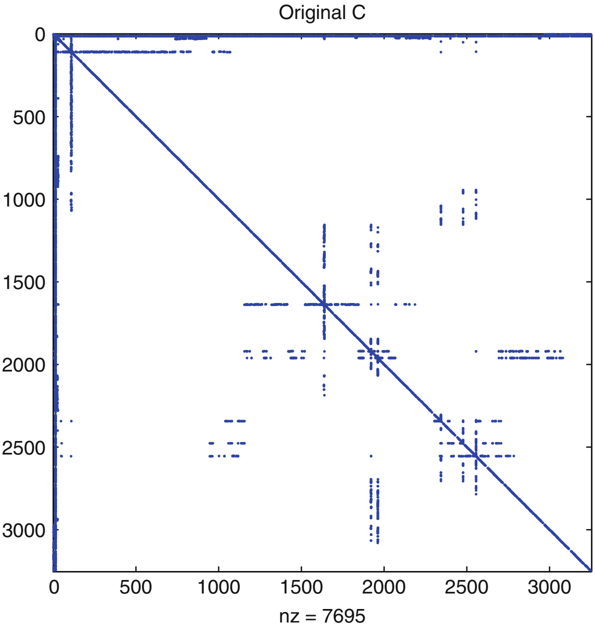


Fig. 6.8 Original **C** matrix

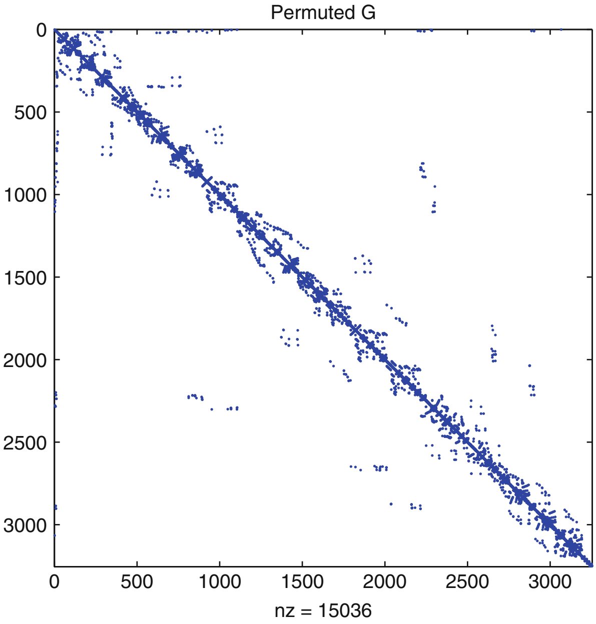


Fig. 6.9 Permuted **G** according to scc(**G**)

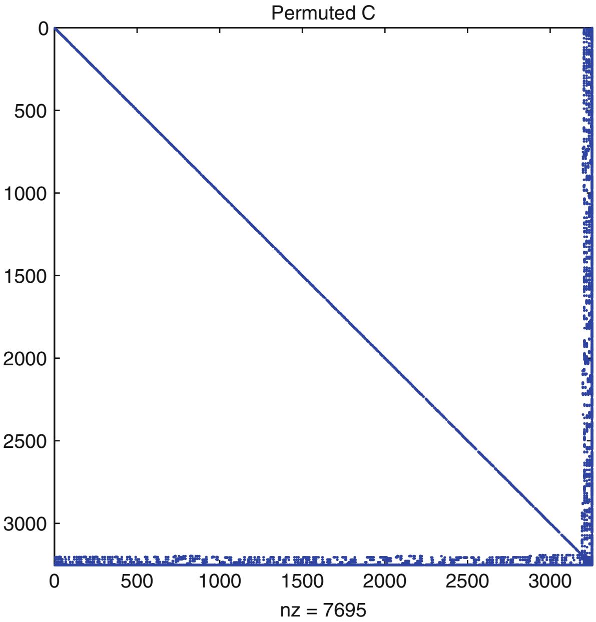


Fig. 6.10 Permuted **C** according to scc(**G**)

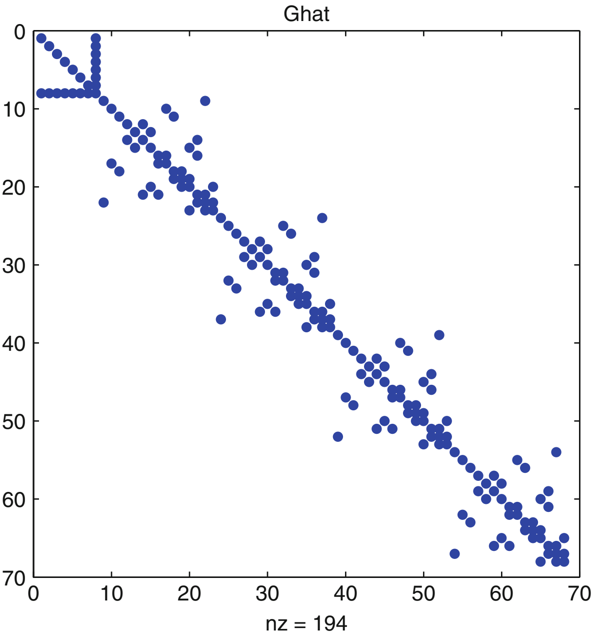


Fig. 6.11 Reduced **G** matrix with Sparse MA

Fig. 6.12 Reduced **C** matrix with Sparse MA

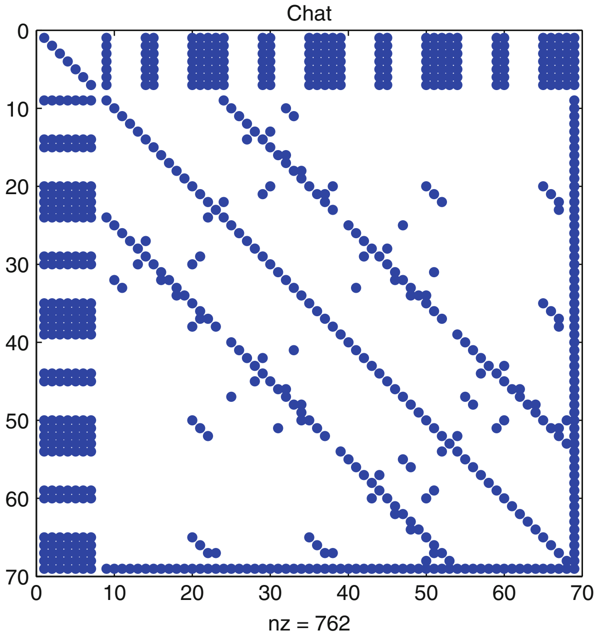
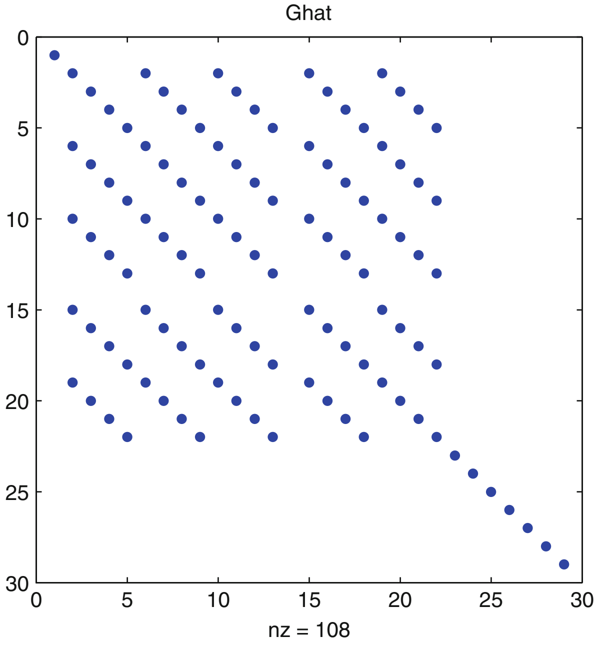


Fig. 6.13 Reduced **G** matrix with PACT



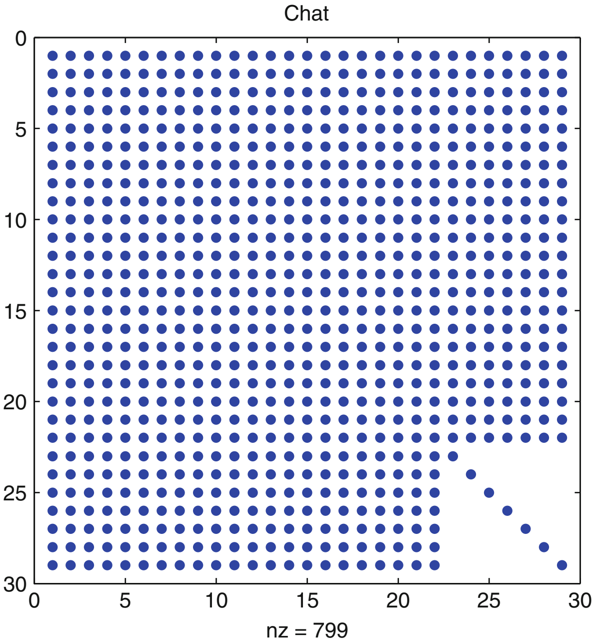


Fig. 6.14 Reduced **C** matrix with PACT

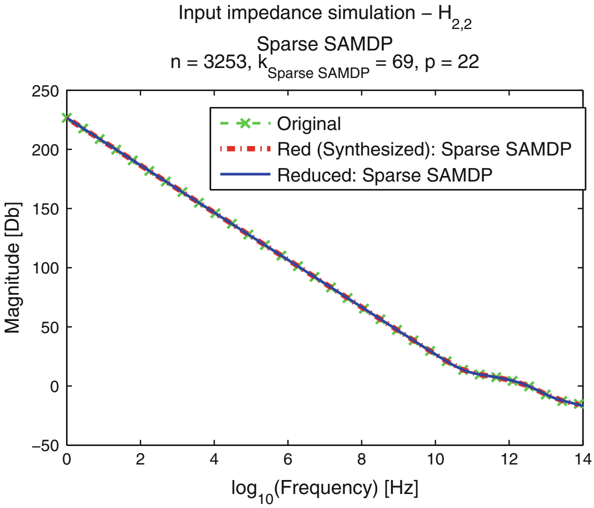


Fig. 6.15 AC simulation 1: original, reduced (Sparse MA) and synthesized model

original model and three reduced models are presented: SparseMA, PACT and the commercial software Jivaro [64]. The SparseMA model is the most accurate for the entire frequency range.

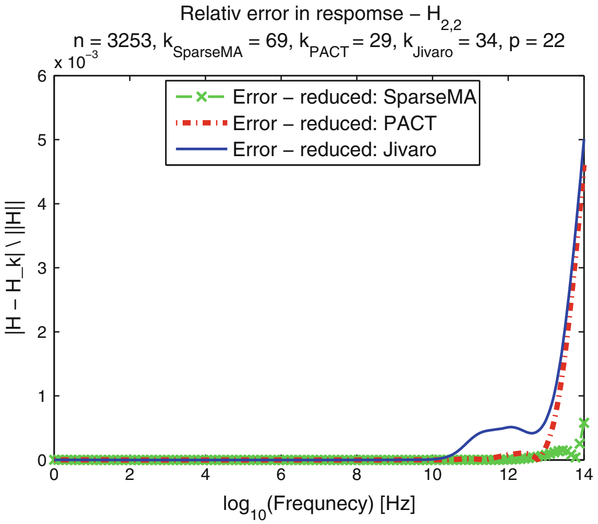


Fig. 6.16 AC simulation 1: relative error between original and reduced models (SparseMA, Pact, Jivaro)

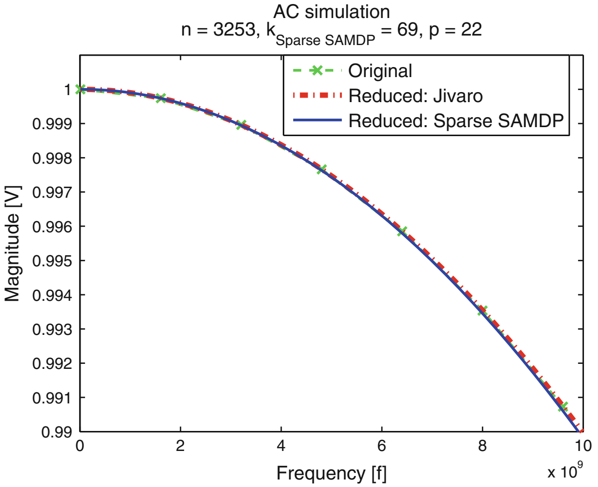


Fig. 6.17 AC simulation 2: original, reduced (Sparse MA) and reduced (Jivaro)

Figure 6.17 shows a different AC circuit simulation, where the SparseMA model performs comparably to the reduced model obtained with the commercial software Jivaro [64]. Finally, the transient simulation in Fig. 6.18 confirms that the SparseMA model is both accurate and stable.

Table 6.3 shows the reduction results for the RC network. For the 3 reduced models: SparseMA, PACT and Jivaro we assess the effect of the reduction by means

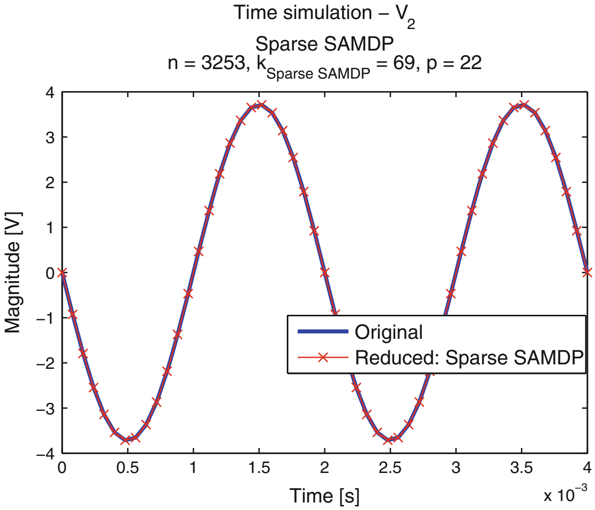


Fig. 6.18 Transient simulation 1: all external nodes grounded and voltage measured at node 2. Original and reduced (Sparse MA – synthesized)

Table 6.3 Results with SparseMA reduction on *RC* netlist

	Original	Red. SparseMA	Red. PACT	Red. Jivaro
#external nodes	22			
#internal nodes	3,231	47	7	12
#unknowns	3,253	69	29	34
#resistors	7,944	78	68	28
#capacitors	3,466	383	414	97
$\frac{\#\text{elements}}{\#\text{int. nodes}}$	3.53	9.8	68.8	10.4
$\frac{\#\text{elements}}{\#\text{unknowns}}$	3.5	6.7	16.6	3.67
CPU time	6.8 s	0.1 s	0.06	0.02 s
Speed up		68×	113×	340×

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

6.2.4 Concluding Remarks

New approaches were presented for reducing R and RC circuits with multi-terminals, using tools from graph theory. It was shown how netlist partitioning and node reordering strategies can be combined with existing model reduction techniques, to improve the sparsity of the reduced RC models and implicitly their simulation time. The proposed sparsity preserving method, SparseMA, performs comparably to the commercial tool Jivaro. Future work will investigate how similar strategies can be applied to RC models with many more terminals [$\sim O(10^3)$] and to *RLCk* netlists.

6.3 Simulation of Mutually Coupled Oscillators Using Nonlinear Phase Macromodels and Model Order Reduction Techniques

The design of modern RF (radio frequency) integrated circuits becomes increasingly more complicated due to the fact that more functionality needs to be integrated on a smaller physical area.[5] In the design process floor planning, i.e., determining the locations for the functional blocks, is one of the most challenging tasks. Modern RF chips for mobile devices, for instance, typically have an FM radio, Bluetooth, and GPS on one chip. These functionalities are implemented with Voltage Controlled Oscillators (VCOs), that are designed to oscillate at certain different frequencies. In the ideal case, the oscillators operate independently, i.e., they are not perturbed by each other or any signal other than their input signal. Practically speaking, however, the oscillators are influenced by unintended (parasitic) signals coming from other blocks (such as Power Amplifiers) or from other oscillators, via for instance (unintended) inductive coupling through the substrate. A possibly undesired consequence of the perturbation is that the oscillators lock to a frequency different than designed for, or show pulling, in which case the oscillators are perturbed from their free running orbit without locking.

The locking effect was first observed by the Dutch scientist Christian Huygens in the seventeenth century. He observed that pendulums of two nearby clocks hanging on the same wall after some time moved in unison [120] (in other words they

[5] Section 6.3 has been written by Davit Harutyunyan, Joost Rommes, E. Jan W. ter Maten and Wil H.A. Schilders.

locked to the same frequency). Similar effects occur also for electrical oscillators. When an oscillator is locked to a different frequency, it physically means that the frequency of the oscillator is changed and as a result the oscillator operates at the new frequency. In this case in the spectrum of the oscillator we will observe a single peak corresponding to the new frequency of the oscillator. Contrary to the locking case, frequency pulling occurs when the interfering frequency source is not strong enough to cause frequency locking (e.g. weak substrate coupling). In this case in the spectrum of the pulled oscillator we will observe several sidebands around the carrier frequency of the oscillator. In Sect. 6.3.9 we will discuss several practical examples of locking and pulling effects.

Oscillators appear in many physical systems and interaction between oscillators has been of interest in many applications. Our main motivation comes from the design of RF systems, where oscillators play an important role [95, 100, 107, 120] in, for instance, high-frequency Phase Locked Loops (PLLs). Oscillators are also used in the modeling of circadian rhythm mechanisms, one of the most fundamental physiological processes [91]. Another application area is the simulation of large-scale biochemical processes [114].

Although the use of oscillators is widely spread over several disciplines, their intrinsic nonlinear behavior is similar, and, moreover, the need for fast and accurate simulation of their dynamics is universal. These dynamics include changes in the frequency spectrum of the oscillator due to small noise signals (an effect known as jitter [100]), which may lead to pulling or locking of the oscillator to a different frequency and may cause the oscillator to malfunction. The main difficulty in simulating these effects is that both phase and amplitude dynamics are strongly nonlinear and spread over separated time scales [113]. Hence, accurate simulation requires very small time steps during time integration, resulting in unacceptable simulation times that block the design flow. Even if computationally feasible, transient simulation only gives limited understanding of the causes and mechanisms of the pulling and locking effects.

To some extent one can describe the relation between the locking range of an oscillator and the amplitude of the injected signal (these terms will be explained in more detail in Sect. 6.3.1). Adler [90] shows that this relation is linear, but it is now well known that this is only the case for small injection levels and that the modeling fails for higher injection levels [111]. Also other linearized modeling techniques [120] suffer, despite their simplicity, from the fact that they cannot model nonlinear effects such as injection locking [111, 127].

In this section we use the nonlinear phase macromodel introduced in [100] and further developed and analyzed in [104–106, 111, 113, 115, 116, 127]. Contrary to linear macromodels, the nonlinear phase macromodel is able to capture nonlinear effects such as injection locking. Moreover, since the macromodel replaces the original oscillator system by a single scalar equation, simulation times are decreased while the nonlinear oscillator effects can still be studied without loss of accuracy. One of the contributions of this paper is that we show how such macromodels can be used in industrial practice to predict the behavior of inductively coupled oscillators.

Returning to our motivation, during floor planning, it is of crucial importance that the blocks are located in such a way that the effects of any perturbing signals are minimized. A practical difficulty here is that transient simulation of the full system is very expensive and usually unfeasible during the early design stages. One way to get insight in the effects of inductive coupling and injected perturbation signals is to apply the phase shift analysis [100]. In this section we will explain how this technique can be used to estimate the effects for perturbed individual and coupled oscillators, and how this can be of help during floor planning. We will consider perturbations caused by oscillators and by other components such as balanced/unbalanced transformers (baluns).

In some applications to reduce clockskew (clocksignals becoming out of phase), for instance, oscillators can be coupled via transmission lines [102]. Since accurate models for transmission lines can be large, this may lead to increased simulation times. We show how model order reduction techniques [94, 96, 97, 124] can be used to decrease simulation times without unacceptable loss of accuracy.

The section is organized as follows. In Sect. 6.3.1 we summarize the phase noise theory. A practical oscillator model and an example application are described in Sect. 6.3.2. Inductively coupled oscillators are discussed in detail in Sect. 6.3.3. In Sect. 6.3.4 we give an overview of existing methods to model injection locking of individual and resistively/capacitively coupled oscillators. In Sect. 6.3.5 we consider small parameter variations for mutually coupled oscillators. In Sects. 6.3.6 and 6.3.7 we show how the phase noise theory can be used to analyze oscillator-balun coupling and oscillator-transmission line coupling, respectively. In Sect. 6.3.8 we give a brief introduction to model order reduction and present a Matlab script used in our implementations. Numerical results are presented in Sect. 6.3.9 and the conclusions are drawn in Sect. 6.3.10.

6.3.1 Phase Noise Analysis of Oscillator

A general free-running oscillator can be expressed as an autonomous system of differential (algebraic) equations:

$$\frac{\mathrm{d}\mathbf{q}(\mathbf{x})}{\mathrm{d}t} + \mathbf{j}(\mathbf{x}) = 0, \tag{6.67a}$$

$$\mathbf{x}(0) = \mathbf{x}(T), \tag{6.67b}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ are the state variables, T is the period of the free running oscillator, which is in general unknown, $\mathbf{q}, \mathbf{j} : \mathbb{R}^n \to \mathbb{R}^n$ are (nonlinear) functions describing the oscillator's behavior and n is the system size. The solution of (6.67) is called Periodic Steady State (PSS) and is denoted by $\mathbf{x}_{pss}$. Although finding the PSS solution can be an challenging task in itself, we will not discuss this in the present paper and refer the interested reader to, for example, [105, 108–110, 122, 123, 126].

A general oscillator under perturbation can be expressed as a system of differential equations

$$\frac{\mathrm{d}\mathbf{q}(\mathbf{x})}{\mathrm{d}t} + \mathbf{j}(\mathbf{x}) = \mathbf{b}(t), \tag{6.68}$$

where $\mathbf{b}(t) \in \mathbb{R}^n$ are perturbations to the free running oscillator. For small perturbations $\mathbf{b}(t)$ it can be shown [100] that the solution of (6.68) can be approximated by

$$\mathbf{x}_p(t) = \mathbf{x}_{pss}(t + \alpha(t)) + \mathbf{y}(t), \tag{6.69}$$

where $\mathbf{y}(t)$ is the orbital deviation and $\alpha(t) \in \mathbb{R}$ is the phase shift, which satisfies the following scalar nonlinear differential equation:

$$\dot{\alpha}(t) = \mathbf{V}^T(t + \alpha(t)) \cdot \mathbf{b}(t), \tag{6.70a}$$

$$\alpha(0) = 0, \tag{6.70b}$$

where $\mathbf{V}(t) \in \mathbb{R}^n$ is called Perturbation Projection Vector (PPV) of (6.68). It is a special projection vector of the perturbations and is computed based on Floquet theory [99, 100, 115]. The PPV is a periodic function with the same period as the oscillator and can efficiently be computed directly from the PPS solution, see for example [101]. Using this simple and numerically cheap method one can do many kinds of analysis for oscillators, e.g. injection locking, pulling, a priori estimate of the locking range [100, 111].

For small perturbations the orbital deviation $\mathbf{y}(t)$ can be ignored [100] and the response of the perturbed oscillator is computed by

$$\mathbf{x}_p(t) = \mathbf{x}_{pss}(t + \alpha(t)). \tag{6.71}$$

6.3.2 LC Oscillator

For many applications oscillators can be modeled as an LC tank with a nonlinear resistor as shown in Fig. 6.19. This circuit is governed by the following differential equations for the unknowns (v, i):

$$C\frac{\mathrm{d}v(t)}{\mathrm{d}t} + \frac{v(t)}{R} + i(t) + S\tanh\left(\frac{G_n}{S}v(t)\right) = b(t), \tag{6.72a}$$

$$L\frac{\mathrm{d}i(t)}{\mathrm{d}t} - v(t) = 0, \tag{6.72b}$$

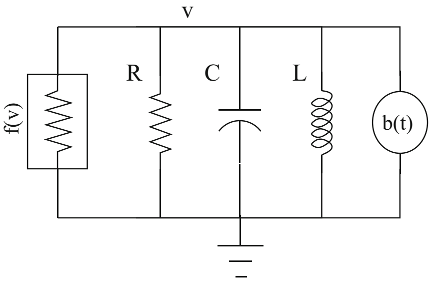


Fig. 6.19 Voltage controlled oscillator: current of the nonlinear resistor is given by $f(v) = S\tanh(\frac{G_n}{S}v(t))$

where C, L and R are the capacitance, inductance and resistance, respectively. The nodal voltage is denoted by v and the branch current of the inductor is denoted by i. The voltage controlled nonlinear resistor is defined by S and G_n parameters, where S has influence on the oscillation amplitude and G_n is the gain [111].

A lot of work [111, 120] has been done for the simulation of this type of oscillators. Here we will give an example that can be of practical use for designers. During the design process, early insight in the behavior of system components is of crucial importance. In particular, for perturbed oscillators it is very convenient to have a direct relationship between the injection amplitude and the side band level.

For the given RLC circuit with the following parameters $L = 930 \cdot 10^{-12}$ H, $C = 1.145 \cdot 10^{-12}$ F, $R = 1{,}000\,\Omega$, $S = 1/R$, $G_n = -1.1/R$ and injected signal $b(t) = A_{\text{inj}}\sin(2\pi f)$, we plot the side band level of the voltage response versus the amplitude A_{inj} of the injected signal for different offset frequencies, see Fig. 6.20. The results in Fig. 6.20 can be seen as a simplified representation of Arnol'd tongues [98], that is helpful in engineering practice. We see, for instance, that the oscillator locks to a perturbation signal with an offset of 10 MHz if the corresponding amplitude is larger than $\sim 10^{-4}$ A (when the signal is locked the sideband level becomes 0 dB). This information is useful when designing the floor plan of a chip, since it may put additional requirements on the placement (and shielding) of components that generate, or are sensitive to, perturbing signals.

As an example, consider the floor plan in Fig. 6.21. The analysis described above and in Fig. 6.20 first helped to identify and quantify the unintended pulling and locking effects due to the coupling of the inductors (note that the potential causes (inductors) of pulling and locking effects first have to be identified; in practice, designers usually have an idea of potential coupling issues, for instance when there are multiple oscillators in a design). The outcome of this analysis indicated that there were unintended pulling effects in the original floorplan and hence some components were relocated (and shielded) to reduce unintended pulling effects. Finally, the same macromodels, but with different coupling factors due to the relocation of components, were used to verify the improved floorplan.

Although the LC tank model is relatively simple, it can be of high value especially in the early stages of the design process (schematic level), since it can be used to estimate the effects of perturbation and (unintended) coupling on the behavior of oscillators. As explained before, this may be of help during floor

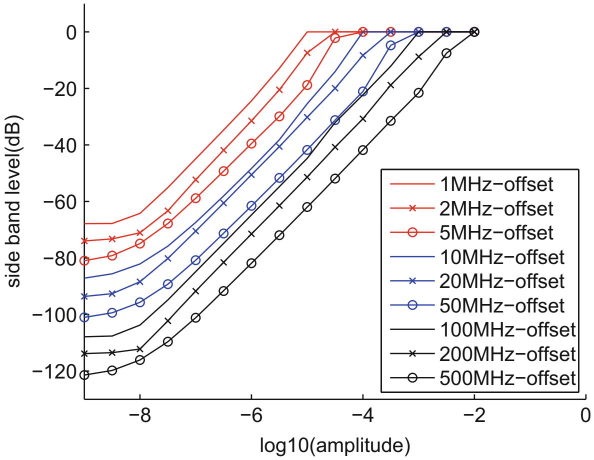


Fig. 6.20 Side band level of the voltage response versus the injected current amplitude for different offset frequencies

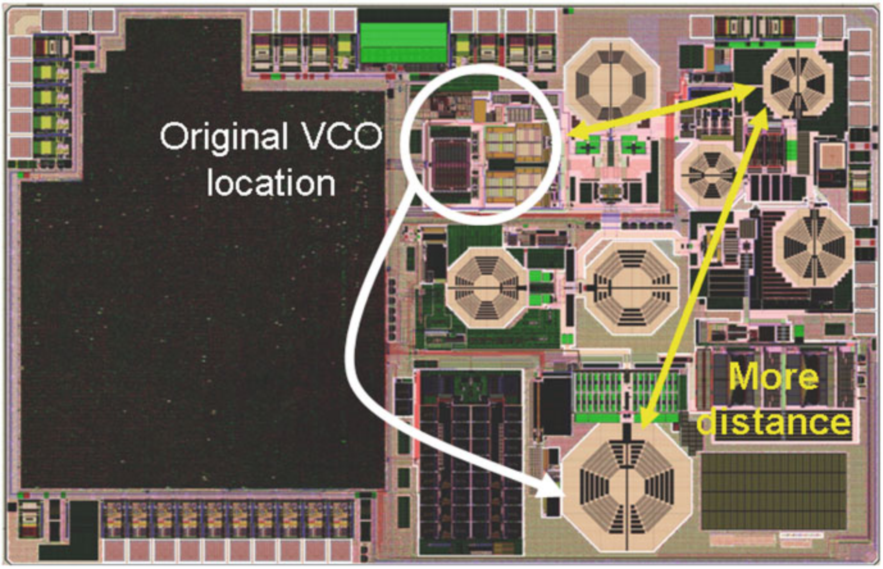


Fig. 6.21 Floor plan with relocation option that was considered after nonlinear phase noise analysis showed an intolerable pulling due to unintended coupling. Additionally, shielding was used to limit coupling effects even further

planning. In later stages, one typically validates the design via layout simulations, which can be much more complex due to the inclusion of parasitic elements. In general one has to deal with larger dynamical systems when parasitics are included, but the phase noise theory still applies. Therefore, in this paper we do not consider

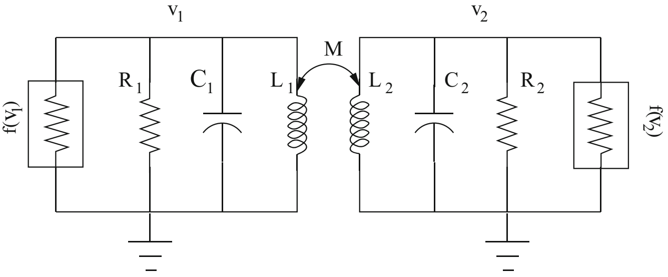


Fig. 6.22 Two inductively coupled LC oscillators

extracted parasitics. However, the values for L, C, R and coupling factors are typically based on measurement data and layout simulations of real designs.

6.3.3 *Mutual Inductive Coupling*

Next we consider the two mutually coupled LC oscillators shown in Fig. 6.22. The inductive coupling between these two oscillators can be modeled as

$$L_1\frac{\mathrm{d}i_1(t)}{\mathrm{d}t}+M\frac{\mathrm{d}i_2(t)}{\mathrm{d}t}=v_1(t), \tag{6.73a}$$

$$L_2\frac{\mathrm{d}i_2(t)}{\mathrm{d}t}+M\frac{\mathrm{d}i_1(t)}{\mathrm{d}t}=v_2(t), \tag{6.73b}$$

where $M = k\sqrt{L_1L_2}$ is the mutual inductance and $|k| < 1$ is the coupling factor. This makes the matrix

$$\begin{pmatrix} L_1 & M \\ M & L_2 \end{pmatrix}$$

positive definite, which ensures that the problem is well posed. In this section all the parameters with a subindex refer to the parameters of the oscillator with the same subindex. If we combine the mathematical model (6.72) of each oscillator with (6.73), then the two inductively coupled oscillators can be described by the following differential equations

$$C_1\frac{\mathrm{d}v_1(t)}{\mathrm{d}t}+\frac{v_1(t)}{R_1}+i_1(t)+S\tanh(\frac{G_n}{S}v_1(t))=0, \tag{6.74a}$$

$$L_1\frac{\mathrm{d}i_1(t)}{\mathrm{d}t}-v_1(t)=-M\frac{\mathrm{d}i_2(t)}{\mathrm{d}t}, \tag{6.74b}$$

$$C_2 \frac{\mathrm{d}v_2(t)}{\mathrm{d}t} + \frac{v_2(t)}{R_2} + i_2(t) + S\tanh(\frac{G_n}{S} v_2(t)) = 0, \tag{6.74c}$$

$$L_2 \frac{\mathrm{d}i_2(t)}{\mathrm{d}t} - v_2(t) = -M \frac{\mathrm{d}i_1(t)}{\mathrm{d}t}. \tag{6.74d}$$

For small values of the coupling factor k the right-hand side of (6.74b) and (6.74d) can be considered as a small perturbation to the corresponding oscillator and we can apply the phase shift theory described in Sect. 6.3.1. Then we obtain the following simple nonlinear equations for the phase shift of each oscillator:

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} 0 \\ -M \dfrac{\mathrm{d}i_2(t)}{\mathrm{d}t} \end{pmatrix}, \tag{6.75a}$$

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} 0 \\ -M \dfrac{\mathrm{d}i_1(t)}{\mathrm{d}t} \end{pmatrix}, \tag{6.75b}$$

where the currents and voltages are evaluated by using (6.71):

$$[v_1(t), i_1(t)]^T = \mathbf{x}^1_{pss}(t + \alpha_1(t)), \tag{6.75c}$$

$$[v_2(t), i_2(t)]^T = \mathbf{x}^2_{pss}(t + \alpha_2(t)). \tag{6.75d}$$

Small parameter variations have also been studied in the literature by Volterra analysis, see e.g. [92, 93].

6.3.3.1 Time Discretization

The system (6.75) is solved by using implicit backward Euler for the time discretization and the Newton method is applied for the solution of the resulting two dimensional nonlinear equations (6.76a) and (6.76b), i.e.

$$\alpha_1^{m+1} = \alpha_1^m + \tau \mathbf{V}_1^T(t^{m+1} + \alpha_1^{m+1}) \cdot \begin{pmatrix} 0 \\ -M \dfrac{i_2(t^{m+1}) - i_2(t^m)}{\tau} \end{pmatrix}, \tag{6.76a}$$

$$\alpha_2^{m+1} = \alpha_2^m + \tau \mathbf{V}_2^T(t^{m+1} + \alpha_2^{m+1}) \cdot \begin{pmatrix} 0 \\ -M \dfrac{i_1(t^{m+1}) - i_1(t^m)}{\tau} \end{pmatrix}, \tag{6.76b}$$

$$[v_1(t^{m+1}), i_1(t^{m+1})]^T = \mathbf{x}^1_{pss}(t^{m+1} + \alpha_1^{m+1}), \tag{6.76c}$$

$$[v_2(t^{m+1}),\, i_2(t^{m+1})]^T = \mathbf{x}^2_{pss}(t^{m+1} + \alpha_2^{m+1}), \tag{6.76d}$$

$$\alpha_1^1 = 0,\ \alpha_2^1 = 0,\ m = 1,\ldots,$$

where $\tau = t^{m+1} - t^m$ denotes the time step. For the Newton iterations in (6.76a) and (6.76b) we take (α_1^m, α_2^m) as initial guess on the time level $(m+1)$. This provides very fast convergence (in our applications within around four Newton iterations). See [123] and references therein for more details on time integration of electric circuits.

6.3.4 *Resistive and Capacitive Coupling*

For completeness in this section we describe how the phase noise theory applies to two oscillators coupled by a resistor or a capacitor.

6.3.4.1 Resistive Coupling

Resistive coupling is modeled by connecting two oscillators by a single resistor, see Fig. 6.23. The current i_{R_0} flowing through the resistor R_0 satisfies the following relation

$$i_{R_0} = \frac{v_1 - v_2}{R_0}, \tag{6.77}$$

where R_0 is the coupling resistance. Then the phase macromodel is given by

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} (v_1 - v_2)/R_0 \\ 0 \end{pmatrix}, \tag{6.78a}$$

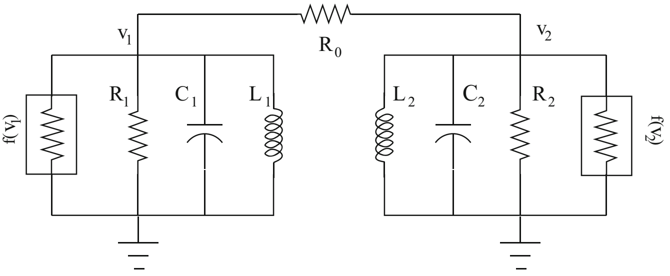


Fig. 6.23 Two resistively coupled LC oscillators

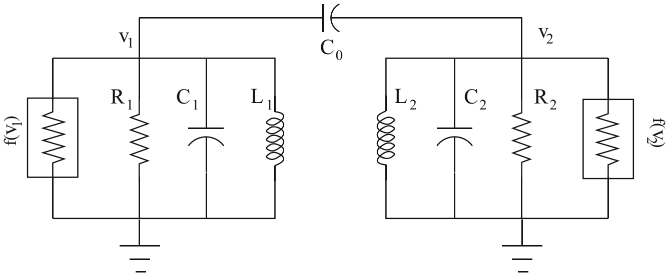


Fig. 6.24 Two capacitively coupled LC oscillators

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} -(v_1 - v_2)/R_0 \\ 0 \end{pmatrix}, \tag{6.78b}$$

where the voltages are updated by using (6.71). More details on resistively coupled oscillators can be found in [113].

6.3.4.2 Capacitive Coupling

When two oscillators are coupled via a single capacitor with a capacitance C_0 (see Fig. 6.24), then the current i_{C_0} through the capacitor C_0 satisfies

$$i_{C_0} = C_0 \frac{d(v_1 - v_2)}{dt}. \tag{6.79}$$

In this case the phase macromodel is given by

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} C_0 \frac{d(v_1 - v_2)}{dt} \\ 0 \end{pmatrix}, \tag{6.80a}$$

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} -C_0 \frac{d(v_1 - v_2)}{dt} \\ 0 \end{pmatrix}, \tag{6.80b}$$

where the voltages are updated by using (6.71).

Time discretization of (6.78) and (6.80) is done according to (6.76).

6.3.5 Small Parameter Variation Model for Oscillators

For many applications performing simulations with nominal design parameters is no longer sufficient and it is necessary to do simulations around the nominal parameters. In practice designers use Monte-Carlo type simulation techniques to get insight about the device performance for small parameter variations. However these methods can be very time consuming and not applicable for large problems. For analyzing small parameter variations one can use polynomial chaos approach described in [119]. But in this paper we apply the technique described in [128] to mutually coupled oscillators. Here we briefly sketch the ideas of the method and for details we refer to [128].

Consider an oscillator under a perturbation $\mathbf{b}(t)$ described by a set of ODE's:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} + f(\mathbf{x}, p) = \mathbf{b}(t), \tag{6.81}$$

where f describes the nonlinearity in the oscillator and it is a function of the state variables $\mathbf{x}$ and the parameter $\mathbf{p}$. Let us consider a parameter variation

$$\mathbf{p} = \mathbf{p}_0 + \varDelta\mathbf{p}, \tag{6.82}$$

where $\mathbf{p}_0$ is the nominal parameter and $\varDelta\mathbf{p}$ is the parameter deviation from $\mathbf{p}_0$. Then for small parameter deviations the phase shift equation for (6.81) reads

$$\dot{\alpha}(t) = \mathbf{V}^T(t + \alpha(t)) \cdot (\mathbf{b}(t) - F_P(t + \alpha(t))\varDelta\mathbf{p}), \tag{6.83a}$$

$$\alpha(0) = 0, \tag{6.83b}$$

where $\mathbf{V}(t)$ is the perturbation projection vector of the oscillator with nominal parameters and

$$F_P(t + \alpha(t)) = \left.\frac{\partial f}{\partial \mathbf{p}}\right|_{\mathbf{x}_{\mathrm{pss}}(t+\alpha(t)),\mathbf{p}_0}, \tag{6.84}$$

where $\mathbf{x}_{\mathrm{pss}}$ is the PSS of (6.81) with nominal parameters.

In Sect. 6.3.9.1 we show numerical experiments of two inductively coupled oscillators using small parameter variations.

6.3.6 Oscillator Coupling with Balun

In this section we analyze inductive coupling effects between an oscillator and a balun. A balun is an electrical transformer that can transform balanced signals to unbalanced signals and vice versa, and they are typically used to change impedance (applications in (RF) radio). The (unintended) coupling between an oscillator and a balun typically occurs on chips that integrate several oscillators for, for instance, FM radio, Bluethooth and GPS, and hence it is important to understand possible

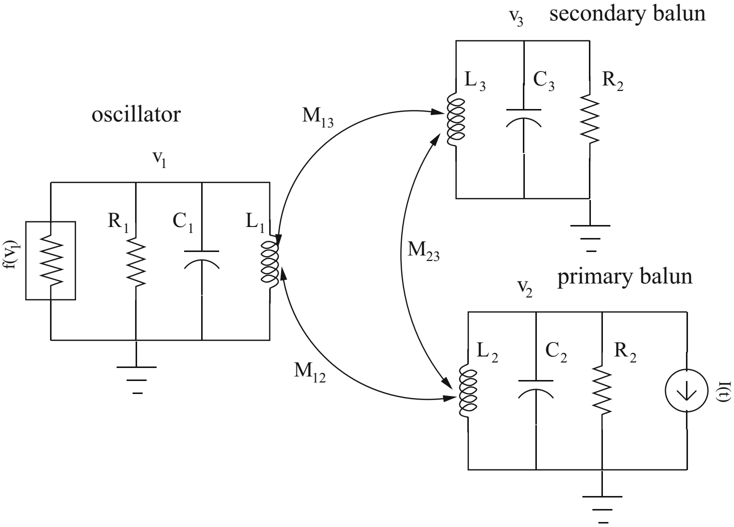


Fig. 6.25 Oscillator coupled with a balun

coupling effects during the design. In Fig. 6.25 a schematic view is given of an oscillator which is coupled with a balun via mutual inductors.

The following mathematical model is used for oscillator and balun coupling (see Fig. 6.25):

$$C_1\frac{dv_1(t)}{dt} + \frac{v_1(t)}{R_1} + i_1(t) + S\tanh(\frac{Gn}{S}v_1(t)) = 0, \tag{6.85a}$$

$$L_1\frac{di_1(t)}{dt} + M_{12}\frac{di_2(t)}{dt} + M_{13}\frac{di_3(t)}{dt} - v_1(t) = 0, \tag{6.85b}$$

$$C_2\frac{dv_2(t)}{dt} + \frac{v_2(t)}{R_2} + i_2(t) + I(t) = 0, \tag{6.85c}$$

$$L_2\frac{di_2(t)}{dt} + M_{12}\frac{di_1(t)}{dt} + M_{23}\frac{di_3(t)}{dt} - v_2(t) = 0, \tag{6.85d}$$

$$C_3\frac{dv_3(t)}{dt} + \frac{v_3(t)}{R_3} + i_3(t) = 0, \tag{6.85e}$$

$$L_3\frac{di_3(t)}{dt} + M_{13}\frac{di_1(t)}{dt} + M_{23}\frac{di_2(t)}{dt} - v_3(t) = 0, \tag{6.85f}$$

where $M_{ij} = k_{ij}\sqrt{L_i L_j}$, $i, j = 1, 2, 3$, $i < j$ is the mutual inductance and k_{ij} is the coupling factor. The parameters of the nonlinear resistor are $S = 1/R_1$ and $G_n = -1.1/R_1$ and the current injection in the primary balun is denoted by $I(t)$.

For small coupling factors we can consider $M_{12}\frac{di_2(t)}{dt} + M_{13}\frac{di_3(t)}{dt}$ in (6.85b) as a small perturbation to the oscillator. Then similar to (6.75), we can apply the phase shift macromodel to (6.85a)–(6.85b). The reduced model corresponding to (6.85a)–(6.85b) is

$$\frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \mathbf{V}^T(t+\alpha(t)) \cdot \begin{pmatrix} 0 \\ -M_{12}\dfrac{di_2(t)}{dt} - M_{13}\dfrac{di_3(t)}{dt} \end{pmatrix}. \tag{6.86}$$

The balun is described by a linear circuit (6.85c)–(6.85f) which can be written in a more compact form:

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} + A\mathbf{x}(t) + B\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} + C = 0, \tag{6.87}$$

where

$$E = \begin{pmatrix} C_2 & 0 & 0 & 0 \\ 0 & L_2 & 0 & M_{23} \\ 0 & 0 & C_3 & 0 \\ 0 & M_{23} & 0 & L_3 \end{pmatrix}, \tag{6.88a}$$

$$A = \begin{pmatrix} 1/R_2 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1/R_3 & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix}, \tag{6.88b}$$

$$B^T = \begin{pmatrix} 0 & M_{12} & 0 & M_{13} \end{pmatrix}, \tag{6.88c}$$

$$C^T = \begin{pmatrix} I(t) & 0 & 0 & 0 \end{pmatrix}, \tag{6.88d}$$

$$\mathbf{x}^T = \begin{pmatrix} v_2(t) & i_2(t) & v_3(t) & i_3(t) \end{pmatrix}. \tag{6.88e}$$

With these notations (6.86) and (6.87) can be written in the following form

$$\frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \mathbf{V}^T(t+\alpha(t)) \cdot \begin{pmatrix} 0 \\ -B^T\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} \end{pmatrix}, \tag{6.89}$$

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} + A\mathbf{x}(t) + B\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} + C = 0, \tag{6.90}$$

where $i_1(t)$ is computed by using (6.71). This system can be solved by using a finite difference method.

6.3.7 Oscillator Coupling to a Transmission Line

In some applications oscillators are coupled via transmission lines. By coupling oscillators via transmission lines, for instance, one can reduce the clock skew in clock distribution networks [102]. Accurate models for transmission lines may contain up to thousands or millions of RLC components [129]. Furthermore, the oscillators or the components that perturb (couple to) the oscillators can consists of many RLC components, for instance when ones takes into account parasitic effects. Since simulation times usually increase with the number of elements, one would like to limit the number of (parasitic) components as much as possible, without losing accuracy.

The schematic view of an oscillator coupled to a transmission line is given in Fig. 6.26. Using phase macromodel for oscillator and by applying Kirchhoff's current law to the transmission line circuit, we obtain the following set of differential equations:

$$\frac{\mathrm{d}\alpha(t)}{\mathrm{d}t} = \mathbf{V}^T(t+\alpha(t)) \cdot \begin{pmatrix} \dfrac{y(t)-v(t)}{R_1} \\ 0 \end{pmatrix} \tag{6.91a}$$

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{6.91b}$$

$$y(t) = \mathcal{C}^T\mathbf{x}, \tag{6.91c}$$

where

$$E = \mathrm{diag}(C_1, C_2, \ldots, C_n),\ A = \mathrm{tridiag}(\frac{1}{R_i}, -\frac{1}{R_i} - \frac{1}{R_{i+1}}, \frac{1}{R_{i+1}}), \tag{6.92a}$$

$$B = \begin{pmatrix} \frac{1}{R_1} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix},\ \mathbf{x} = \begin{pmatrix} v_1(t) \\ v_2(t) \\ \vdots \\ v_n(t) \end{pmatrix},\ \mathbf{u}(t) = \begin{pmatrix} v(t) \\ I(t) \end{pmatrix},\ \mathcal{C} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{6.92b}$$

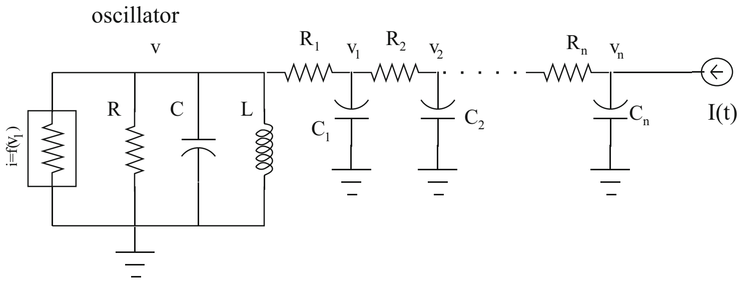


Fig. 6.26 Oscillator coupled to a transmission line

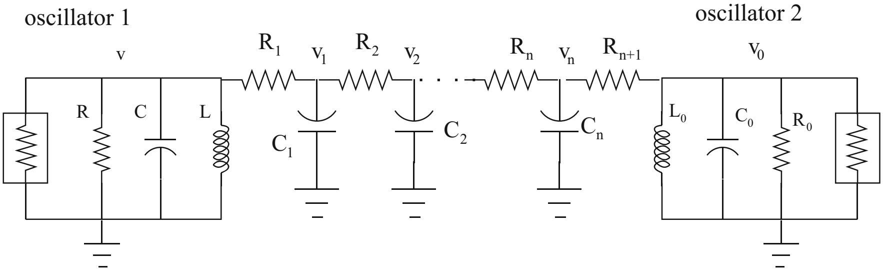


Fig. 6.27 Two oscillators coupled via a transmission line

In a similar way the phase macromodel of two oscillators coupled via a transmission line, see Fig. 6.27, is given by the following equations:

$$\frac{\mathrm{d}\alpha_1(t)}{\mathrm{d}t} = \mathbf{V}_1^T(t+\alpha_1(t)) \cdot \begin{pmatrix} \dfrac{v_1(t)-v(t)}{R_1} \\ 0 \end{pmatrix} \tag{6.93a}$$

$$E\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = A\mathbf{x}(t) + B\mathbf{u}(t), \tag{6.93b}$$

$$\frac{\mathrm{d}\alpha_2(t)}{\mathrm{d}t} = \mathbf{V}_2^T(t+\alpha_2(t)) \cdot \begin{pmatrix} \dfrac{v_n(t)-v_0(t)}{R_{n+1}} \\ 0 \end{pmatrix}, \tag{6.93c}$$

where $\alpha_1(t)$ and $\alpha_2(t)$ ($\mathbf{V}_1$ and $\mathbf{V}_2$) are phase shifts (PPV's) of the corresponding oscillator. The matrices E, A and $\mathbf{x}$ are given by (6.92) and

$$B = \begin{pmatrix} \frac{1}{R_1} & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & \frac{1}{R_{n+1}} \end{pmatrix}, \quad \mathbf{u}(t) = \begin{pmatrix} v(t) \\ v_0(t) \end{pmatrix}. \tag{6.94}$$

6.3.8 Model Order Reduction

Model order reduction (MOR) techniques [94, 96, 97, 124] can be used to reduce the number of elements significantly. Here we show how model order reduction can be used for the analysis of oscillator perturbation effects as well. Since the main focus is to show how MOR techniques can be used (and not which technique is the most suitable), we limit the discussion here to Balanced Truncation [118]. For other methods, see, e.g., [94, 96, 97, 124].

Given a dynamical system (A, B, C) (assume $E = I$), balanced truncation [118] consists of first computing a balancing transformation $V \in \mathbb{R}^{n \times n}$. The balanced system $(V^T AV, V^T B, V^T C)$ has the nice property that the Hankel Singular Values[6] are easily available. A reduced order model can be constructed by selecting the columns of V that correspond to the $k < n$ largest Hankel Singular Values. With $V_k \in \mathbb{R}^{n \times k}$ having as columns these k columns, the reduced order model (of order k) becomes $(V_k^A V_k, V_k^T B, V_k^T C)$. If $E \neq I$ is nonsingular, balanced truncation can be applied to $(E^{-1}A, E^{-1}B, C)$. For more details on balanced truncation, see [96, 97, 118, 124].

In this section we apply model order reduction to linear circuits that are coupled to oscillators, and the relevant equations for each problem describing linear circuits have the form of (6.89b)–(6.89c). For each problem the corresponding matrices A, E, B, and C can be identified readily, see (6.88), (6.92), (6.94) and note $C \equiv \mathcal{C}$. We use Matlab [117] implementation for balanced truncation to obtain reduced order models:

```
sys = ss( -E\A, -E\B, C′, 0 ) ;
[hsv, baldata] = hsvd(sys); % Hankel singular values
mor_dim = nnz((hsv>1e-10)); % choose largest singular
% values where mor_dim is the dimension
% of the reduced system
rsys= balred(sys,mor_dim,’Elimination’,’Truncate’,...
’Balancing’, baldata) ; %truncate
```

Note that we can apply balanced truncation because E is nonsingular. It is well known that in many cases in circuit simulation the system is a descriptor system and hence E is singular. Although generalizations of balanced truncation to descriptor systems exist [124, 125], other MOR techniques such as Krylov subspace methods and modal approximation might be more appropriate. We refer the reader to [94, 96, 97, 124] for a good introduction to such techniques and MOR in general.

6.3.9 Numerical Experiments

It is known that a perturbed oscillator either locks to the injected signal or is pulled, in which case side band frequencies all fall on one side of the injected signal, see, e.g., [111]. We will see that contrary to the single oscillator case, where side band frequencies all fall on one side of the injected signal, for (weakly) coupled oscillators a double-sided spectrum is formed.

In Sects. 6.3.9.1–6.3.9.3 we consider two LC oscillators with different kinds of coupling and injection. The inductance and resistance in both oscillators are

[6] Similar to singular values of matrices, the Hankel singular values and corresponding vectors can be used to identify the dominant subspaces of the system’s statespace: the larger the Hankel singular value, the more dominant.

$L_1 = L_2 = 0.64\,\text{nH}$ and $R_1 = R_2 = 50\,\Omega$, respectively. The first oscillator is designed to have a free running frequency $f_1 = 4.8$ GHz with capacitance $C_1 = 1/(4L_1\pi^2 f_1^2) = 1.7178$ pF. Then the inductor current in the first oscillator is $A_1 = 0.0303$ A and the capacitor voltage is $V_1 = 0.5844$ V. In a similar way the second oscillator is designed to have a free running frequency $f_2 = 4.6$ GHz with the inductor current $A_2 = 0.0316$ A and the capacitor voltage $V_2 = 0.5844$ V. For both oscillators we choose $S_i = 1/R_i$, $G_n = -1.1/R_i$ with $i = 1, 2$. In Sect. 6.3.9.4 we describe experiments for an oscillator coupled to a balun.

The values for L, C, R and (mutual) coupling factors are based on measurement data and layout simulations of real designs.

In all the numerical experiments the simulations are run until $T_{\text{final}} = 6 \cdot 10^{-7}$ s with the fixed time step $\tau = 10^{-11}$. Simulation results with the phase shift macromodel are compared with simulations of the full circuit using the CHORAL[103, 121] one-step time integration algorithm, hereafter referred to as full simulation. All experiments have been carried out in Matlab 7.3. We would like to remark that in all experiments simulations with the macromodels were typically ten times faster than the full circuit simulations.

In all experiments, for a given oscillator or balun we use the response of the nodal voltage to plot the spectrum (spectrum composed of discrete harmonics) of the signal.

6.3.9.1 Inductively Coupled Oscillators

Numerical simulation results of two inductively coupled oscillators, see Fig. 6.22, for different coupling factors k are shown in Fig. 6.28, where the frequency is plotted versus the Power Spectral Density (PSD[7]). In Fig. 6.28 we present results for the first oscillator. Similar results are obtained for the second oscillator around its own carrier frequency. For small values of the coupling factor we observe a very good approximation with the full simulation results. As the coupling factor grows, small deviations in the frequency occur, see Fig. 6.28d. Because of the mutual pulling effects between the two oscillators a double sided spectrum is formed around each oscillator carrier frequency. The additional sidebands are equally spaced by the frequency difference of the two oscillators.

The phase shift $\alpha_1(t)$ of the first oscillator for a certain time interval is given in Fig. 6.29. We note that it has a sinusoidal behavior. For a single oscillator under perturbation a completely different behavior is observed: in locked condition the phase shift changes linearly, whereas in the unlocked case the phase shift has a nonlinear behavior different than a sinusoidal, see for example [112].

[7] Matlab code for plotting the PSD is given in [107].

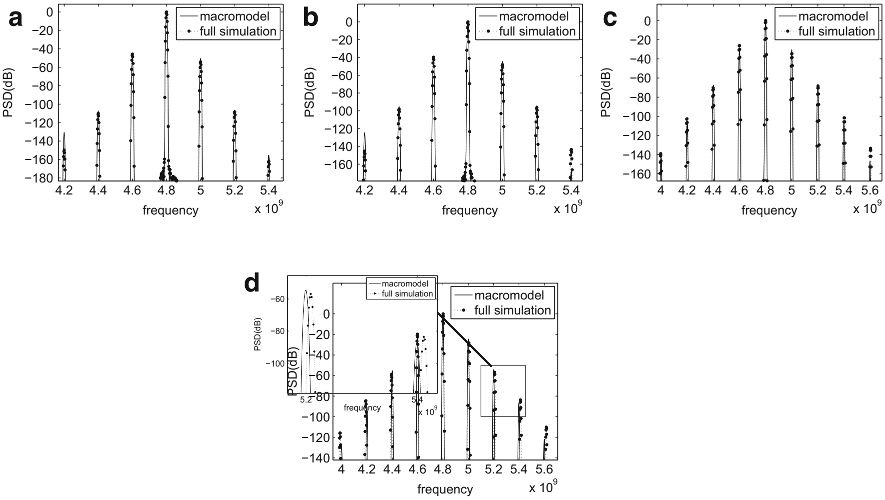


Fig. 6.28 Inductive coupling. Comparison of the output spectrum of the first oscillator obtained by the phase macromodel and by the full simulation for a different coupling factor k. (**a**) $k = 0.0005$. (**b**) $k = 0.001$. (**c**) $k = 0.005$. (**d**) $k = 0.01$

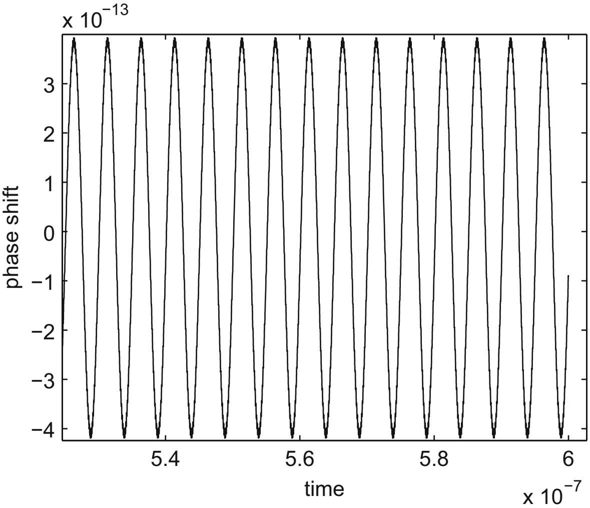


Fig. 6.29 Inductive coupling. Phase shift $\alpha_1(t)$ of the first oscillator with $k = 0.001$

Parameter Variation in Two Inductively Coupled Oscillators

Let us consider two inductively coupled oscillators with the nominal parameters given in Sect. 6.3.9 and a small parameter ΔL variation in the inductance of the second oscillator. Then the corresponding model is:

$$\frac{\mathrm{d}v_1(t)}{\mathrm{d}t} + \frac{v_1(t)}{C_1 R_1} + i_1(t) + \frac{S}{C_1}\tanh(\frac{G_n}{S}v_1(t)) = 0, \tag{6.95a}$$

$$\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} - \frac{v_1(t)}{L_1} = -\frac{M}{L_1}\frac{\mathrm{d}i_2(t)}{\mathrm{d}t}, \tag{6.95b}$$

$$\frac{\mathrm{d}v_2(t)}{\mathrm{d}t} + \frac{v_2(t)}{C_2 R_2} + i_2(t) + \frac{S}{C_2}\tanh(\frac{G_n}{S}v_2(t)) = 0, \tag{6.95c}$$

$$\frac{\mathrm{d}i_2(t)}{\mathrm{d}t} - \frac{v_2(t)}{L_2 + \Delta L} = -\frac{M}{L_2 + \Delta L}\frac{\mathrm{d}i_1(t)}{\mathrm{d}t}. \tag{6.95d}$$

By using the small parameter variation model given in Sect. 6.3.5 we obtain the corresponding phase shift macromodel for (6.95):

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} 0 \\ -\frac{M}{L_1}\frac{\mathrm{d}i_2(t)}{\mathrm{d}t} \end{pmatrix}, \tag{6.96a}$$

$$\dot{\alpha_2}(t) = \mathbf{V}_2^T(t + \alpha_2(t)) \cdot \begin{pmatrix} 0 \\ -\frac{M}{L_2 + \Delta L}\frac{\mathrm{d}i_1(t)}{\mathrm{d}t} - \frac{v_2(t)}{L_2^2}\Delta L \end{pmatrix}, \tag{6.96b}$$

where the currents and voltages are evaluated by using (6.75c)–(6.75d).

For this numerical experiments we consider the coupling factor to be equal to $k = 0.0005$. Furthermore, let us denote by $f_2^{\text{full},\Delta L}$ and $f_2^{\text{phase},\Delta L}$ the new frequency of the second oscillator obtained by full simulation and phase macromodel for the given parameter variation ΔL. Then we define

$$\Delta f = f_2^{\text{full},\Delta L} - f_2^{\text{phase},\Delta L}.$$

In Fig. 6.30 we show the relative frequency difference Δf versus parameter variation ΔL. We note that for small parameter variations ($\Delta L/L_2 \leq 0.01$) the phase macromodel provides a good approximation to the full simulation results.

In Fig. 6.31 we show the output spectrum of the second oscillator for several values of the parameter ΔL.

6.3.9.2 Capacitively Coupled Oscillators

The coupling capacitance in Fig. 6.24 is chosen to be $C_0 = k \cdot C_{\text{mean}}$, where $C_{\text{mean}} = (C_1 + C_2)/2 = 1.794 \cdot 10^{-12}$ and we call k the capacitive coupling factor. Simulation results for the first oscillator for different capacitive coupling factors k are given in Fig. 6.32 (similar results are obtained for the second oscillator around its own carrier frequency).

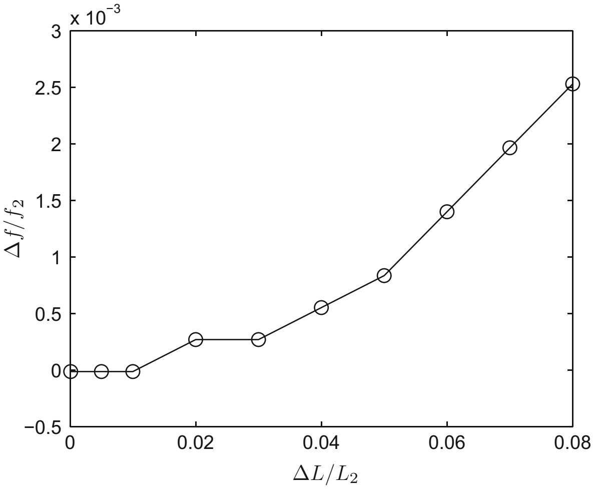


Fig. 6.30 Frequency difference versus parameter variation

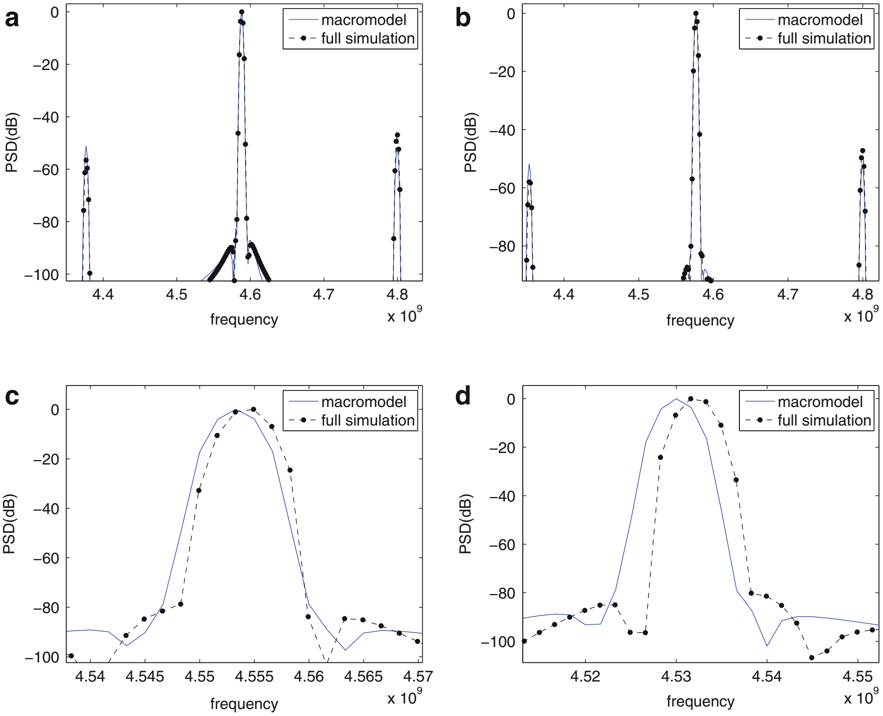


Fig. 6.31 Output spectrum of the second oscillator for several parameter variations ΔL. (**a**) $\Delta L/L_2 = 0.005$. (**b**) $\Delta L/L_2 = 0.01$. (**c**) $\Delta L/L_2 = 0.02$. (**d**) $\Delta L/L_2 = 0.03$

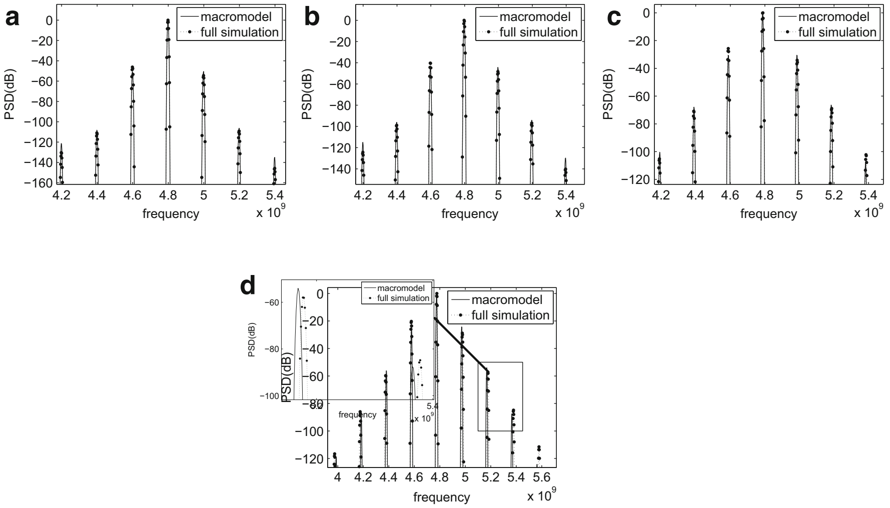


Fig. 6.32 Capacitive coupling. Comparison of the output spectrum of the first oscillator obtained by the phase macromodel and by the full simulation for a different coupling factor k. (**a**) $k = 0.0005$. (**b**) $k = 0.001$. (**c**) $k = 0.005$. (**d**) $k = 0.01$

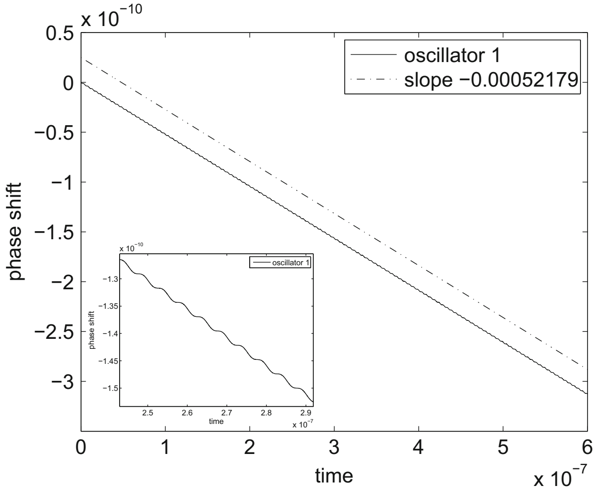


Fig. 6.33 Capacitive coupling. Phase shift of the first oscillator with $k = 0.001$

For a larger coupling factor $k = 0.01$ the phase shift macromodel shows small deviations from the full simulation results Fig. 6.32d.

The phase shift $\alpha_1(t)$ of the first oscillator and a zoomed section for some interval are given in Fig. 6.33. In a long run the phase shift seems to change linearly with a slope of $a = -0.00052179$. The linear change in the phase shift is a clear indication that the frequency of the first oscillator is changed and is locked to a new frequency, which is equal to $(1+a)f_1$. The change of the frequency can be explained as follows: as noted in [114], capacitive coupling may change the free running frequency because this kind of coupling changes the equivalent tank capacitance. From a mathematical point of view it can be explained in the following way. For the capacitively coupled oscillators the governing equations can be written as:

$$(C_1 + C_0)\frac{dv_1(t)}{dt} + \frac{v_1(t)}{R} \tag{6.97a}$$

$$+ i_1(t) + S\tanh(\frac{G_n}{S}v_1(t)) = C_0\frac{dv_2(t)}{dt},$$

$$L_1\frac{di_1(t)}{dt} - v_1(t) = 0, \tag{6.97b}$$

$$(C_2 + C_0)\frac{dv_2(t)}{dt} + \frac{v_2(t)}{R} \tag{6.97c}$$

$$+ i_2(t) + S\tanh(\frac{G_n}{S}v_2(t)) = C_0\frac{dv_1(t)}{dt},$$

$$L_2\frac{di_2(t)}{dt} - v_2(t) = 0. \tag{6.97d}$$

This shows that the capacitance in each oscillator is changed by C_0 and that the new frequency of each oscillator is

$$\tilde{f_i} = \frac{1}{2\pi\sqrt{L_1(C_i + C_0)}}, \ i = 1, 2.$$

In the zoomed figure within Fig. 6.33 we note that the phase shift is not exactly linear but that there are small wiggles. By numerical experiments it can be shown that these small wiggles are caused by a small sinusoidal contribution to the linear part of the phase shift. As in case of mutually coupled inductors, the small sinusoidal contributions are caused by mutual pulling of the oscillators (right-hand side terms in (6.97a) and (6.97c)).

6.3.9.3 Inductively Coupled Oscillators Under Injection

As a next example, let us consider two inductively coupled oscillators where in one of the oscillators an injected current is applied. Let us consider the case where a sinusoidal current of the form

$$I(t) = A_{\text{inj}}\sin(2\pi(f_1 - f_{\text{off}})t) \tag{6.98}$$

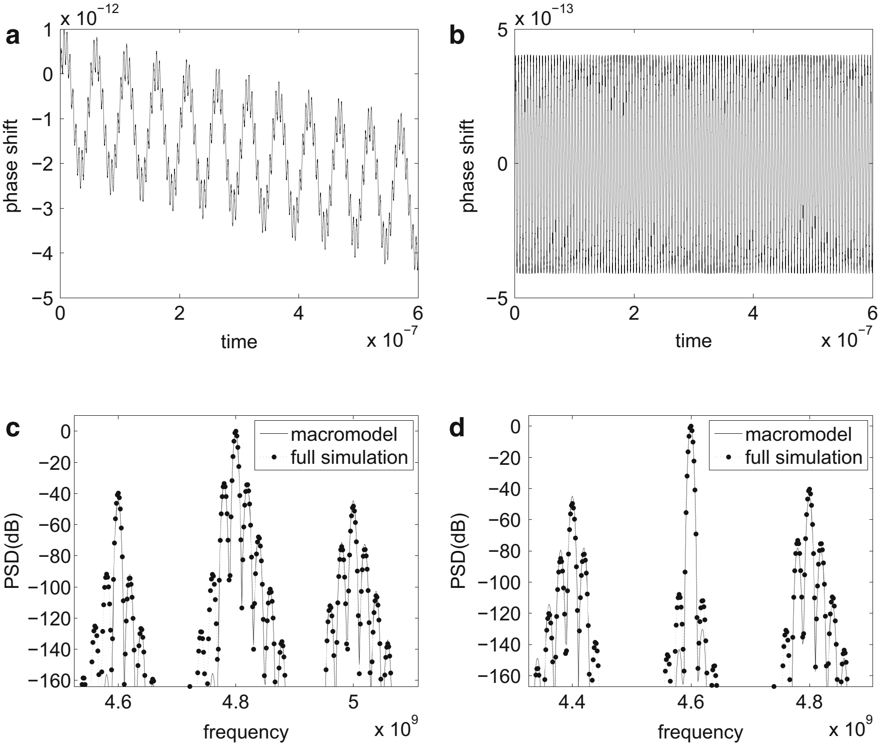


Fig. 6.34 Inductive coupling with injection and $k = 0.001$. *Top*: phase shift. *Bottom*: comparison of the output spectrum obtained by the phase macromodel and by the full simulation with a small current injection. (**a**) Oscillator 1. (**b**) Oscillator 2. (**c**) Oscillator 1. (**d**) Oscillator 2

is injected in the first oscillator. Then (6.75a) is modified to

$$\dot{\alpha_1}(t) = \mathbf{V}_1^T(t + \alpha_1(t)) \cdot \begin{pmatrix} -I(t) \\ -M\dfrac{di_2(t)}{dt} \end{pmatrix}. \tag{6.99}$$

For a small current injection with $A_{\text{inj}} = 10\,\mu\text{A}$ and an offset frequency $f_{\text{off}} =$ 20 MHz the spectra of both oscillators and the phase shift with coupling factor $k = 0.001$ are given in Fig. 6.34. It is clear from Figs. 6.34a, b that the phase shift of both oscillators does not change linearly, which implies that the oscillators are not in the steady state. As a result in Figs. 6.34c, d we observe spectral widening in the spectra of both oscillators. We note that the phase macromodel simulations are good approximations of the full simulation results.

6.3.9.4 Oscillator Coupled to a Balun

Finally, consider an oscillator coupled to a balun as shown in Fig. 6.25 with the following parameters values:

Oscillator	Primary balun	Secondary balun
$L_1 = 0.64\,\text{nH}$	$L_2 = 1.10\,\text{nH}$	$L_3 = 3.60\,\text{nH}$
$C_1 = 1.71\,\text{pF}$	$C_2 = 4.00\,\text{pF}$	$C_3 = 1.22\,\text{pF}$
$R_1 = 50\,\Omega$	$R_2 = 40\,\Omega$	$R_2 = 60\,\Omega$

The coupling factors in (6.85) are chosen to be

$$k_{12} = 10^{-3},\ k_{13} = 5.96 * 10^{-3},\ k_{23} = 9.33 * 10^{-3}. \tag{6.100}$$

The injected current in the primary balun is of the form

$$I(t) = A_{\text{inj}} \sin(2\pi(f_0 - f_{\text{off}})t), \tag{6.101}$$

where $f_0 = 4.8\,\text{GHz}$ is the oscillator's free running frequency and $f_{\text{off}} = 20$ MHz is the offset frequency.

Results of numerical experiments done with the phase macromodel and the full simulations are shown in Fig. 6.35. We note that for a small current injection ($A_{\text{inj}} = 10^{-4} - 10^{-2}$ A) both the oscillator and the balun are pulled by each other. When the injected current is not strong ($A_{\text{inj}} = 10^{-4}$ A) the oscillator is pulled slightly and in the spectrum of the oscillator (Fig. 6.35a) we observe a spectral widening with two spikes around-60 dB (weak "disturbance" of the oscillator). By gradually increasing the injected current, the oscillator becomes more disturbed and in the spectrum we observe widening with higher side band levels, cf. Fig. 6.35c–f. When the injected current is strong enough (with $A_{\text{inj}} = 10^{-1}$ A) to lock the oscillator to the frequency of the injected signal, we observe a single spike at the new frequency. Similar results are also obtained for the secondary balun.

Oscillator Coupled to a Balun

Consider an oscillator coupled to a balun as shown in Fig. 6.25 with the following parameters values:

Oscillator	Primary balun	Secondary balun
$L_1 = 0.64 \cdot 10^{-9}$	$L_2 = 1.10 \cdot 10^{-9}$	$L_3 = 3.60 \cdot 10^{-9}$
$C_1 = 1.71 \cdot 10^{-12}$	$C_2 = 4.00 \cdot 10^{-12}$	$C_3 = 1.22 \cdot 10^{-12}$
$R_1 = 50$	$R_2 = 40$	$R_2 = 60$

The coefficients of the mutual inductive couplings are $k_{12} = 10^{-3}$, $k_{13} = 5.96 * 10^{-3}$, $k_{23} = 9.33 * 10^{-3}$. The injected current in the primary balun is of the form

$$I(t) = A_{\text{inj}} \sin(2\pi(f_0 - f_{\text{off}})t), \tag{6.102}$$

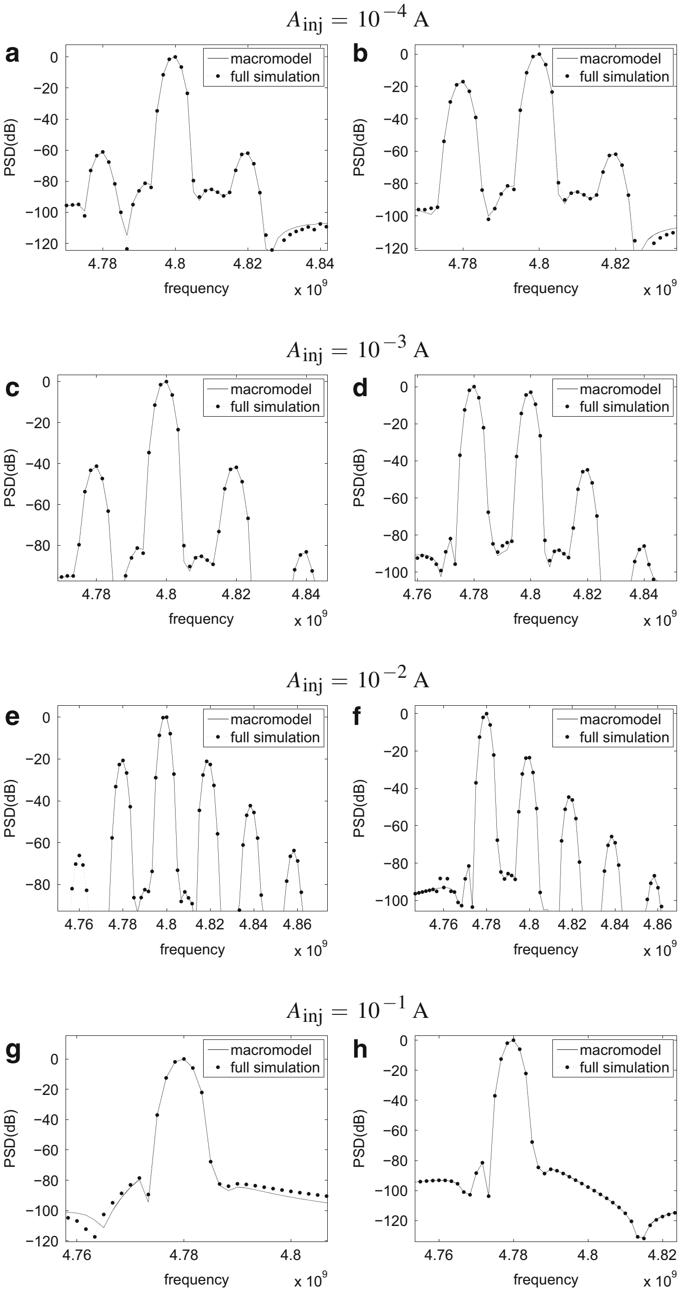


Fig. 6.35 Comparison of the output spectrum of the oscillator coupled to a balun obtained by the phase macromodel and by the full simulation for an increasing injected current amplitude A_{inj} and an offset frequency $f_{\text{off}} = 20$ MHz. (**a**) oscillator. (**b**) primary balun. (**c**) oscillator. (**d**) primary balun. (**e**) oscillator. (**f**) primary balun. (**g**) oscillator. (**h**) primary balun

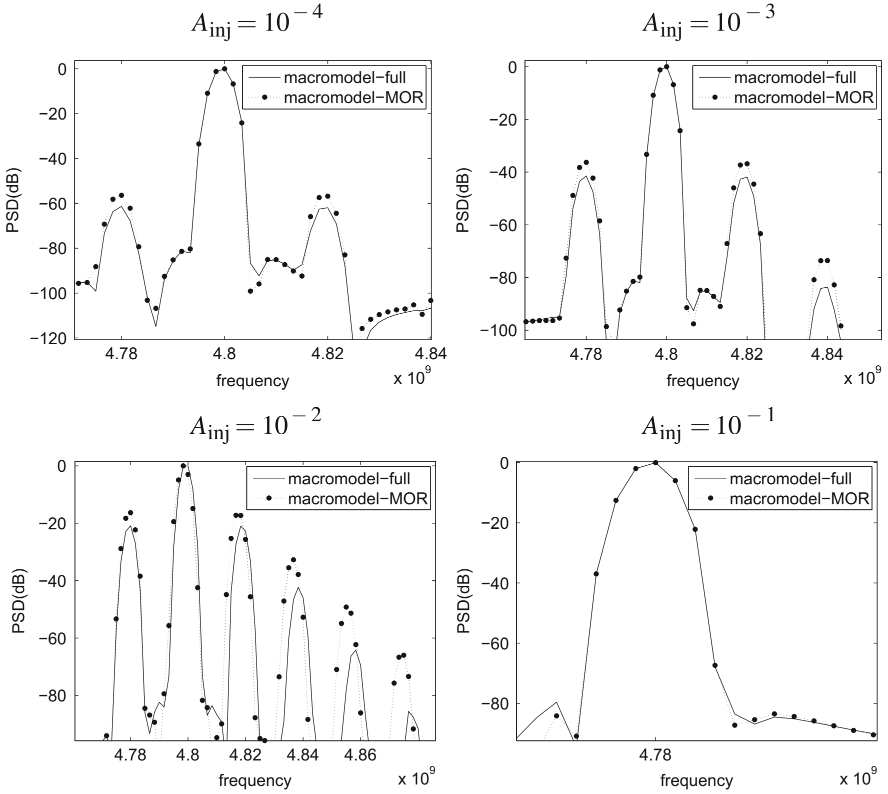


Fig. 6.36 Comparison of the output spectrum of the oscillator coupled to a balun obtained by the macromodel-full and the macromodel-MOR simulations for an increasing injected current amplitude A_{inj} and an offset frequency $f_{\text{off}} = 20$ MHz

where $f_0 = 4.8$ GHz is the oscillator's free running frequency, f_{off} is the offset frequency and A_{inj} is the current amplitude.

Results of the numerical experiments are shown in Fig. 6.36, where the results obtained by the macromodel-MOR technique with mor_dim = 2 provide a good approximation to the full-simulation results. We note that for the injected current with $A_{\text{inj}} = 10^{-1}$ A the oscillator is locked to the injected signal. Similar results are also obtained for the balun.

6.3.9.5 Oscillators Coupled with Transmission Lines

In this section we consider two academic examples, where transmission lines are modeled with RC components.

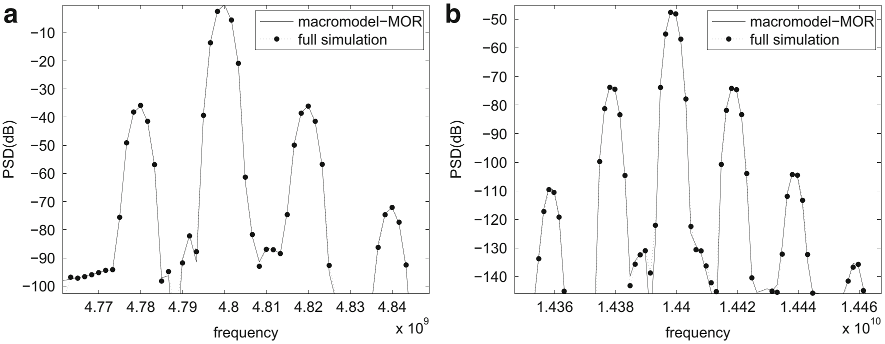


Fig. 6.37 Comparison of the output spectrum around the first and third harmonics of the oscillator coupled to a transmission line, cf. Fig. 6.26. (**a**) first harmonic. (**b**) third harmonic

Single Oscillator Coupled to a Transmission Line

Let us consider the same oscillator as given in the previous section, now coupled to a transmission line, see Fig. 6.26. The size of the transmission line is $n = 100$ with the following parameters: $C_1 = \ldots = C_n = 10^{-2}\,\text{pF}, R_1 = 40\,\text{k}\Omega, R_2 = \ldots = R_n = 1\,\Omega$. The injected current has the form (6.102) with $A_{\text{inj}} = 10^{-2}$ A and $f_{\text{off}} = 20\,\text{MHz}$. Dimension of the reduced system is mor_dim = 18. Simulation results around the first and third harmonics (this oscillator does not have a second harmonic) are shown in Fig. 6.37. The macromodel-MOR method, using techniques described in Sect. 6.3.8, gives a good approximation to the full simulation results.

Two LC Oscillators Coupled via a Transmission Line

For this experiment we consider two LC oscillators coupled via a transmission line with the mathematical model given by (6.93). The first oscillator has a free running frequency $f_1 = 4.8\,\text{GHz}$ and is described in Sect. 6.3.9.4. The second LC oscillator has the following parameter values: $R_0 = 50\,\Omega$, $L_0 = 0.64\,\text{nH}$, $C_0 = 1.87\,\text{pF}$ and a free running frequency $f_2 = 4.6\,\text{GHz}$. The size of the transmission line is $n = 100$ with the following parameters: $C_1 = \ldots = C_n = 10^{-2}\,\text{pF}, R_1 = R_{n+1} = 4\,\text{k}\Omega, R_2 = \ldots = R_n = 0.001\,\Omega$. Dimension of the reduced system is mor_dim = 16. Numerical simulation results are given in Fig. 6.38. We note that macromodel-MOR approach gives a very good approximation to the full-simulation results.

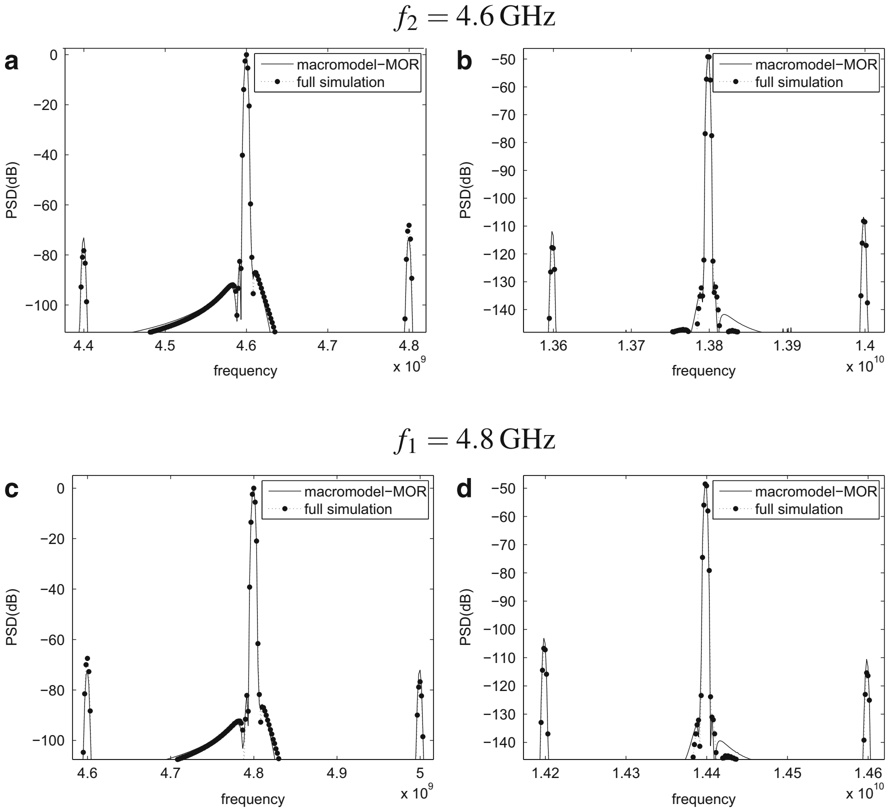


Fig. 6.38 Comparison of the output spectrum around the first and third harmonics of two oscillators coupled via a transmission line. (**a**) first harmonic. (**b**) third harmonic. (**c**) first harmonic. (**d**) third harmonic

6.3.10 Conclusion

In this section we have shown how nonlinear phase macromodels can be used to accurately predict the behavior of individual or mutually coupled voltage controlled oscillators under perturbation, and how they can be used during the design process. Several types of coupling (resistive, capacitive, and inductive) have been described and for small perturbations, the nonlinear phase macromodels produce results with accuracy comparable to full circuit simulations, but at much lower computational costs. Furthermore, we have studied the (unintended) coupling between an oscillator and a balun, a case which typically arises during design and floor planning of RF circuits. For the coupling of oscillators with transmission lines we showed how the phase macromodel can be used with model order reduction techniques to provide an accurate and efficient method.

References

References for Section 6.1

1. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
2. Astrid, P.: Reduction of process simulation models: a proper orthogonal decomposition approach. Ph.D.-thesis, Technische Universiteit Eindhoven (2004)
3. Astrid, P., Verhoeven, A.: Application of least squares mpe technique in the reduced order modeling of electrical circuits. In: Proceedings of the 17th International Symposium on MTNS, Kyoto, pp. 1980–1986 (2006)
4. Astrid, P., Weiland, S., Willcox, K., Backx, T.: Missing point estimation in models described by proper orthogonal decomposition. IEEE Trans. Autom. Control **53**(10), 2237–2251 (2008)
5. Bai, Z., Skoogh, D.: Krylov subspace techniques for reduced-order modeling of nonlinear dynamical systems. Appl. Numer. Math. **43**, 9–44 (2002)
6. Bai, Z., Skoogh, D.: A projection method for model reduction of bilinear dynamical systems. Linear Algebra Appl. 415(2–3), 406–425 (2006)
7. Bechtold, T., Striebel, M., Mohaghegh, K., ter Maten, E.J.W.: Nonlinear model order reduction in nanoelectronics: combination of POD and TPWL. PAMM Proc. Appl. Maths Mech. **8**(1), 10057–10060 (2009). doi:10.1002/pamm.200810057. (Special Issue on Proceedings GAMM Annual Meeting 2008)
8. Benner, P., Breiten, T.: Krylov-subspace based model reduction of nonlinear circuit models using bilinear and quadratic-linear approximations. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, vol. 17, pp. 153–159. Springer, Berlin/New York (2012)
9. Benner, P., Damm, T.: Lyapunov equations, energy functionals and model order reduction of bilinear and stochastic systems. SIAM J. Control Optim. **49**(2), 686–711 (2011)
10. Bittner, K., Urban, K.: Adaptive wavelet methods using semiorthogonal spline wavelets: sparse evaluation of nonlinear functions. Appl. Comput. Harmon. Anal. **24**, 91–119 (2008)
11. Cai, W., Wang, J.: Adaptive multiresolution collocation methods for initial-boundary value problems of nonlinear PDEs. SIAM J. Numer. Anal. **33**(3), 937–970 (1996)
12. Chaturantabut, C., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput. **32**(5), 2737–2764 (2010)
13. Chaturantabut, C., Sorensen, D.C.: A state space error estimate for POD-DEIM nonlinear model reduction. SIAM J. Numer. Anal. **50**(1), 46–63 (2012)
14. Condon, M., Ivanov, R.: Nonlinear systems – algebraic gramians and model reduction. COMPEL Int. J. Comput. Math. Electr. Electron. Eng. **24**(1), 202–219 (2005)
15. Condon, M., Ivanov, R.: Krylov subspaces from bilinear representations of nonlinear systems. COMPEL Int. J. Comput. Math. Electr. Electron. Eng. **26**(2), 399–406 (2007)
16. Dautbegović, E., Condon, M., Brennan, C.: An efficient nonlinear circuit simulation technique. IEEE. Trans. Microw. Theory Tech. **53**(2), 548–555 (2005)
17. Dautbegović, E.: Wavelets in ciruit simulation. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 131–141. Springer, Berlin/Heidelberg (2010)
18. Dong, N., Roychowdhury, J.: General-purpose nonlinear model-order reduction using piecewise-polynomial representations. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(2), 249–264 (2008)
19. Freund, R.W.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**(1–2), 395–421 (2000)
20. Fujimoto, K., Scherpen, J.M.A.: Singular value analysis and balanced realizations for nonlinear systems. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications, pp. 251–272. Springer, Berlin/Heidelberg (2008)

21. Gad, E., Nakhla, M.: Efficient model reduction of linear periodically time-varying systems via compressed transient system function. IEEE Trans. Circuit Syst. 1 **52**(6), 1188–1204 (2005)
22. Gu, C.: Model order reduction of nonlinear dynamical systems. Ph.D.-thesis, University of California, Berkeley (2012). http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-217.pdf
23. Günther, M.: Simulating digital circuits numerically – a charge-oriented ROW approach. Numer. Math. **79**, 203–212 (1998)
24. Holmes, P., Lumley, J., Berkooz, G.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambrige University Press, Cambrige (1996)
25. Ionescu, T.C.: Balanced truncation for dissipative and symmetric nonlinear systems. Ph.D.-thesis, Rijksuniversiteit Groningen (2009)
26. Ionescu, T.C., Scherpen, J.M.A.: Nonlinear cross gramians. In: Korytowski, A., Malanowski, K., Mitkowski, W., Szymkat, M. (eds.) System Modeling and Optimization. Series: IFIP Advances in Information and Communication Technology, vol. 312, pp. 293–306. Springer, Berlin/Heidelberg (2009)
27. Loève, M.: Probability Theory. Van Nostrand, New York (1955)
28. Martinez, J.A.: Model order reduction of nonlinear dynamic systems using multiple projection bases and optimized state-space sampling. Ph.D.-thesis, University of Pittsburgh (2009). http://d-scholarship.pitt.edu/6685/
29. Mohaghegh, K., Striebel, M., ter Maten, J., Pulch, R.: Nonlinear model order reduction based on trajectory piecewise linear approach: comparing different linear cores. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 563–570. Springer, Berlin/Heidelberg (2010)
30. Moore, B.: Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans. Autom. Control **26**(1), 17–32 (1981)
31. Pinnau, R.: Model reduction via proper orthogonal decomposition. In: Schilders, W., van der Vorst, H., Rommes, J. (eds.) Model Order Reduction: Theory, Applications, and Research Aspects, pp. 95–109. Springer, Berlin (2008)
32. Rathinam, M., Petzold, L.R.: A new look at proper orthogonal decomposition. SIAM J. Numer. Anal. **41**(5), 1893–1925 (2003)
33. Rewieński, M.J., White, J.: A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. IEEE Trans. CAD Int. Circuit. Syst. **22**(2), 155–170 (2003)
34. Rewieński, M.J.: A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems. Ph.D.-thesis, Massachusetts Institute of Technology (2003)
35. Striebel, M., Rommes, J.: Model order reduction of nonlinear systems in circuit simulation: status, open issues, and applications. CSC Preprint 08-07, Chemnitz University of Technology (2008). http://www.tu-chemnitz.de/mathematik/csc/index.php
36. Striebel, M., Rommes, J.: Model order reduction of nonlinear systems in circuit simulation: status and applications. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 289–301. Springer, Dordrecht (2011)
37. Striebel, M., Rommes, J.: Model order reduction of parameterized nonlinear systems by interpolating input-output behavior. In: Michielsen, B., Poirier, J.-R. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 16, pp. 405–413. Springer, Heidelberg (2012)
38. Striebel, M., Rommes, J.: Model order reduction of nonlinear systems by interpolating input-output behavior. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, vol. 17, pp. 145–151. Springer, Berlin/New York (2012)
39. Tiwary, S.K., Rutenbar, R.A.: Scalable trajectory methods for on-demand analog macromodel extraction. In: DAC '05: Proceedings of the 42nd Annual Conference on Design Automation, San Diego, pp. 403–408. ACM, New York (2005)

40. Vasilyev, D., Rewieński, M., White, J.: A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS. In: Proceedings of the Design Automation Conference, Anaheim, pp. 490–495 (2003)
41. Verhoeven, A.: Redundancy reduction of IC models by multirate time-integration and model order reduction. Ph.D.-thesis, TU Eindhoven (2008). http://alexandria.tue.nl/extra2/200712281.pdf
42. Verhoeven, A., Voss, T., Astrid, P., ter Maten, E.J.W., Bechtold, T.: Model order reduction for nonlinear problems in circuit simulation. PAMM Proc. Appl. Math. Mech. **7**(1), 1021603–1021604 (2008). doi:10.1002/pamm.200700537. (Special Issue Proceedings ICIAM-2007)
43. Verhoeven, A., ter Maten, J., Striebel, M., Mattheij, R.: Model order reduction for nonlinear IC models. In: Korytowski, A., Malanowski, K., Mitkowski, W., Szymkat, M. (eds.) System Modeling and Optimization. Series: IFIP Advances in Information and Communication Technology, vol. 312, pp. 476–491. Springer, Berlin/Heidelberg (2009)
44. Verhoeven, A., Striebel, M., ter Maten, E.J.W.: Model order reduction for nonlinear IC models with POD. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 571–578. Springer, Berlin/Heidelberg (2010)
45. Verhoeven, A., Striebel, M., Rommes, J., ter Maten, E.J.W., Bechtold, T.: Proper orthogonal decomposition model order reduction of nonlinear IC models. In: Fitt, A.D., Norbury, J., Ockendon, H., Wilson, E. (eds.) Progress in Industrial Mathematics at ECMI 2008, Mathematics in Industry, vol. 15, pp. 441–446. Springer, Berlin/Heidelberg (2010)
46. Verriest, E.I.: Time variant balancing and nonlinear balanced realizations. In: Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.) Model Order Reduction: Theory, Research Aspects and Applications, pp.213–250. Springer, Berlin/Heidelberg (2008)
47. Volkwein, S.: Model reduction using proper orthogonal decomposition . TU Graz (2008). http://www.uni-graz.at/imawww/volkwein/POD.pdf
48. Voß, T.: Model reduction for nonlinear differential algebraic equations. MSc.-thesis, Bergische Universität Wuppertal. Also published as Nat.Lab. Unclassified Report PR-TN-2005/00919, Philips Research (2005)
49. Voß, T., Pulch, R., ter Maten, J., El Guennouni, A.: Trajector piecewise linear aproach for nonlinear differential-algebraic equations in circuit simulation. In: Ciuprina, G., Ioan, D. (eds.) Scientific Computing in Electrical Engineering. Mathematics in Industry, vol. 11, pp. 167–173. Springer, Berlin/New York (2007)
50. Willcox, K., Peraire, J.: Balanced model reduction via the proper orthogonal decomposition. AIAA J. **40**(11), 2323–2330 (2002)
51. Zhou, D., Cai, W.: A fast wavelet collocation method for highspeed circuit simulation. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **46**(8), 920–930 (1999)
52. Zhou. D., Cai, W., Zhang, W.: An adaptive wavelet method for nonlinear circuit simulation. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **46**(8), 930–938 (1999)
53. Zong, K., Yang, F., Zeng, X.: A wavelet-collocation-based trajectory piecewise-linear algorithm for time-domain model-order reduction of nonlinear circuits. IEEE. Trans. Circuits Syst. I Regular Papers **57**(11), 2981–2990 (2010)

References for Section 6.2

54. Amestoy, P.R., Davis, T.A., Duff, I.S.: An approximate minimum degree ordering algorithm. SIAM J. Matrix Anal. Appl. **17**(4), 886–905 (1996)
55. Benner, P., Schneider, A.: Model order and terminal reduction approaches via matrix decomposition and low rank approximation. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008. Mathematics in Industry, vol. 14, pp. 523–530. Springer, Berlin/Heidelberg (2010)

56. Benner, P., Schneider, A.: On stability, passivity and reciprocity preservation of ESVDMOR. In: [58], pp. 277–287 (2011)
57. Benner, P., Schneider, A.: Model reduction for linear descriptor systems with many ports. In: Günther, M., Bartel, A., Brunk, M., Schöps, S., Striebel, M. (eds.) Progress in Industrial Mathematics at ECMI 2010. Mathematics in Industry, vol. 17, pp. 137–144. Springer, Berlin/New York (2012)
58. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
59. Cadence: AssuraRCX. http://www.cadence.com
60. Chua, L.O., Lin, P.: Computer Aided Analysis of Electric Circuits: Algorithms and Computational Techniques, 1st edn. Prentice Hall, Englewood Cliffs (1975)
61. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms, 1st edn. MIT, Cambridge (1990)
62. Davis, T.A.: Suite sparse: a suite of sparse matrix packages. http://www.cise.ufl.edu/research/sparse/SuiteSparse/
63. Duff, I.S., Erisman, A.M., Reid, J.K.: Direct Methods for Sparse Matrices. Clarendon Press, Oxford (1986)
64. Jivaro: available from EdXact SA, Voiron. http://www.edxact.com
65. Electrostatic discharge association. http://www.esda.org
66. Freund, R.W.: SPRIM: Structure-preserving reduced-order interconnect macromodeling. In: Technical Digest of the 2004 IEEE/ACM International Conference on CAD, Los Alamitos, pp. 80–87 (2004)
67. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. John Hopkins University Press, Baltimore (1996)
68. Ionutiu, R.: Model order reduction for multi-terminal systems – with applications to circuit simulation. Ph.D.-thesis, TU Eindhoven (2011). http://alexandria.tue.nl/extra2/716352.pdf
69. Ionutiu, R., Rommes, J.: Circuit synthesis of reduced order models. Technical report 2008/00316, NXP Semiconductors (2009)
70. Ionutiu, R., Rommes, J.: A framework for synthesis of reduced order models. In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 4, Section 4.5, this volume (2015)
71. Ionutiu, R., Rommes, J., Antoulas, A.C.: Passivity Preserving Model Reduction using the Dominant Spectral Zero Method. In: Coupled Multiscale Simulation and Optimization in Nanoelectronics, Ch. 4, Section 4.4, this volume (2015)
72. Karypis, G., Kumar, V.: METIS, A software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. http://glaros.dtc.umn.edu/gkhome/metis/
73. Kerns, K.J., Yang, A.T.: Stable and efficient reduction of large, multiport networks by pole analysis via congruence transformations. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **16**(7), 734–744 (1997)
74. Kerns, K.J., Yang, A.T.: Preservation of passivity during RLC network reduction via split congruence transformations. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **17**(7), 582–591 (1998)
75. Kolyer, J.M., Watson, D.: ESD: From A to Z. Springer, Boston (1996)
76. McCalla, W.J.: Fundamentals of Computer Aided Circuit Simulation, 1st edn. Kluwer Academic, Boston (1988)
77. Miettinen, P., Honkala, M., Roos, J.: Using METIS and hMETIS Algorithms in Circuit Partitioning. Circuit Theory Laboratory Report Series CT-49. Helsinki University of Technology, Espoo (2006)
78. Pstar: In-house industrial circuit simulator. NXP Semiconductors, Eindhoven. http://www.nxp.com
79. Phillips, J.R., Silveira, L.M.: Poor man's tbr: a simple model reduction scheme. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **24**(1), 283–288 (2005)

80. Rommes, J., Martins, N.: Efficient computation of transfer function dominant poles using subspace acceleration IEEE Trans. Power Syst. **21**(3), 1218–1226 (2006)
81. Rommes, J., Martins, N.: Efficient computation of multivariable transfer function dominant poles using subspace acceleration. IEEE Trans. Power Syst. **21**(4), 1471–1483 (2006)
82. Rommes, J.: Methods for eigenvalue problems with applications in model order reduction. Ph.D.-thesis, Utrecht University (2007). http://igitur-archive.library.uu.nl/dissertations/2007-0626-202553/index.htm
83. Rommes, J., Lenaers, P., Schilders, W.H.A.: Reduction of large resistor networks. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering 2008. Series Mathematics in Industry, vol. 14, pp. 555–562. Springer, Berlin/Heidelberg (2010)
84. Rommes, J., Schilders, W.H.A.: Efficient methods for large resistor networks. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **29**(1), 28–39 (2010)
85. Rommes, J., Martins, N.: Model reduction using modal approximation. This COMSON Handbook (2013)
86. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
87. Yang, F., Zeng, X., Su, Y., Zhou, D.: RLC equivalent circuit synthesis method for structure-preserved reduced-order model of interconnect in VLSI. Commun. Comput. Phys. **3**(2), 376–396 (2008)
88. Zečević, A.I., Šiljak, D.D.: Balanced decompositions of sparse systems for multilevel prallel processing. IEEE Trans. Circuit Syst. I Fund. Theory Appl. **41**(3), 220–233 (1994)
89. Zhou, Q., Sun, K., Mohanram, K., Sorensen, D.C.: Large power grid analysis using domain decomposition. In: Proceedings of the Design Automation and Test in Europe (DATE), Munich, pp. 27–32 (2006)

References for Section 6.3

90. Adler, R.: A study of locking phenomena in oscillators. Proc. I.R.E. Waves Electron. **34**, 351–357 (1946)
91. Agarwal, S., Roychowdhury, J.: Efficient multiscale simulation of circadian rhythms using automated phase macromodelling techniques. In: Proceedingsof the Pacific Symposium on Biocomputing, Kohala Coast, vol. 13, pp. 402–413 (2008)
92. Aikio, J.P.: Frequency domain model fitting and Volterra analysis implemented on top of harmonic balance simulation. Ph.D.-thesis, Faculty of Technology of the University of Oulu (2007). http://jultika.oulu.fi/Record/isbn978-951-42-8420-5
93. Aikio, J.P., Makitalo, M., Rahkonen, T.: Harmonic load-pull technique based on Volterra analysis. In: European Microwave Conference, Rome, pp. 1696–1699 (2009)
94. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
95. Banai, A., Farzaneh, F.: Locked and unlocked behaviour of mutually coupled microwave oscillators. IEE Proc. Antennas Propag. **147**, 13–18 (2000)
96. Benner, P., Mehrmann, V., Sorensen, D. (eds.): Dimension Reduction of Large-Scale Systems. Lecture Notes in Computational Science and Engineering, vol. 45. Springer, Berlin/New York (2005)
97. Benner, P., Hinze, M., ter Maten, E.J.W. (eds.): Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74. Springer, Dordrecht (2011)
98. Boyland, P.L.: Bifurcations of circle maps: Arnol'd tongues, bistability and rotation intervals. Commun. Math. Phys. **106**(3), pp. 353–381 (1986)
99. Brachtendorf, H.G.: Theorie und Analyse von autonomen und quasiperiodisch angeregten elektrischen Netzwerken. Habilitationsschrift, Universität Bremen (2001)

100. Demir, A., Mehrotra, A., Roychowdhury, J.: Phase noise in oscillators: a unifying theory and numerical methods for characterization. IEEE Trans. Circuit Syst. I **47**(5), 655–674 (2000)
101. Demir, A., Long, D., Roychowdhury, J.: Computing phase noise eigenfunctions directly from steady-state jacobian matrices. IEEE/ACM International Conference on Computer Aided Design, ICCAD-2000, San Jose, pp. 283–288 (2000)
102. Galton, I., Towne, D.A., Rosenberg, J.J., Jensen, H.T.: Clock distribution using coupled oscillators. In: IEEE International Symposium on Circuits and Systems, Atlanta, vol. 3, pp. 217–220 (1996)
103. Günther, M.: Simulating digital circuits numerically – a charge-oriented ROW approach. Numer. Math. **79**, 203–212 (1998)
104. Harutyunyan, D., Rommes, J., ter Maten, J., Schilders, W.: Simulation of mutually coupled oscillators using nonlinear phase macromodels. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. (TCAD) **28**(10), 1456–1466 (2009)
105. Günther, M., Feldmann, U., ter Maten, J.: Modelling and discretization of circuit problems. In: [123], pp. 523–659 (2005)
106. Harutyunyan, D., Rommes, J.: Simulation of coupled oscillators using nonlinear phase macromodels and model order reduction. In: [97], pp. 163–175 (2011)
107. Heidari, M.E., Abidi, A.A.: Behavioral models of frequency pulling in oscillators. In: IEEE International Behavioral Modeling and Simulation Workshop, San Jose, pp. 100–104 (2007)
108. Houben, S.H.J.M.: Circuits in motion: the numerical simulation of electrical oscillators. Ph.D.-thesis, Technische Universiteit Eindhoven (2003). http://alexandria.tue.nl/extra2/200310849.pdf
109. Kevenaar, T.A.M.: Periodic steady state analysis using shooting and wave-form-Newton. Int. J. Circuit Theory Appl. **22**(1), 51–60 (1994)
110. Kundert, K., White, J., Sangiovanni-Vincentelli, A.: An envelope-following method for the efficient transient simulation of switching power and filter circuits. In: IEEE International Conference on Computer-Aided Design, ICCAD-88. Digest of Technical Papers, Santa Clara, pp. 446–449 (1988)
111. Lai, X., Roychowdhury, J.: Capturing oscillator injection locking via nonlinear phase-domain macromodels. IEEE Trans. Micro. Theory Tech. **52**(9), 2251–2261 (2004)
112. Lai, X., Roychowdhury, J.: Automated oscillator macromodelling techniques for capturing amplitude variations and injection locking. In: IEEE/ACM International Conference onComputer Aided Design, ICCAD-2004, San Jose, pp. 687–694 (2004)
113. Lai, X., Roychowdhury, J.: Fast and accurate simulation of coupled oscillators using nonlinear phase macromodels. In: 2005 IEEE MTT-S International Microwave Symposium Digest, Long Beach, pp. 871–874 (2005)
114. Lai, X., Roychowdhury, J.: Fast simulation of large networks of nanotechnological and biochemical oscillators for investigating self-organization phenomena. In: Proceedings of the IEEE Asia South-Pacific Design Automation Conference, Yokohama, pp. 273–278 (2006)
115. Maffezzoni, P.: Unified computation of parameter sensitivity and signal-injection sensitivity in nonlinear oscillators. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(5), 781–790 (2008)
116. Maffezzoni, P., D'Amore, D.: Evaluating pulling effects in oscillators due to small-signal injection. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **28**(1), 22–31 (2009)
117. Mathworks: Matlab 7 (2009). http://www.mathworks.com/
118. Moore, B.C.: Principal component analysis in linear systems: controllability, observability and model reduction. IEEE Trans. Autom. Control **26**(1), pp. 17–32 (1981)
119. Pulch, R.: Polynomial chaos expansions for analysing oscillators with uncertainties. In: Troch, I., Breitenecker, F. (eds.) Proceedings MATHMOD 09 Vienna, ARGESIM Report 35 (Full Papers), TU Vienna (2009)
120. Razavi, B.: A study of injection locking and pulling in oscillators. IEEE J. Solid-State Circuit **39**(9), 1415–1424 (2004)
121. Rentrop, P., Günther, M., Hoschek, M., Feldmann, U.: CHORAL—a charge-oriented algorithm for the numerical integration of electrical circuits. In: Jäger, W., Krebs, H.-J. (eds.) Mathematics—Key Technology for the Future, pp. 429–438. Springer, Berlin (2003)

122. Semlyen, A., Medina, A.: Computation of the periodic steady state in systems with nonlinear components using a hybrid time and frequency domain method. IEEE Trans. Power Syst. **10**(3), 1498–1504 (1995)
123. Schilders, W.H.A., ter Maten, E.J.W. (eds.) Numerical Methods in Electromagnetics. Handbook of Numerical Analysis, vol. 13. Elsevier, Amsterdam/Oxford (2005)
124. Schilders, W.H.A., van der Vorst, H.A., Rommes, J. (eds.): Model Order Reduction: Theory, Research Aspects and Applications. Mathematics in Industry, vol. 13. Springer, Berlin (2008)
125. Stykel, T.: Gramian based model reduction for descriptor systems. Math. Control Signals Syst. **16**, 297–319 (2004)
126. ter Maten, E.J.W., Fijnvandraat, J.G., Lin, C., Peters, J.M.F.: Periodic AC and periodic noise in RF simulation for electronic circuit design. In: Antreich, K., Bulirsch, R., Gilg, A., Rentrop, P. (eds.) Modeling, Simulation and Optimization of Integrated Circuits. International Series of Numerical Mathematics, vol. 146, pp. 121–134. Birkhäuser Verlag, Basel (2003)
127. Wan, Y., Lai, X., Roychowdhury, J.: Understanding injection locking in negative resistance lc oscillators intuitively using nonlinear feedback analysis. In: Proceedings of the IEEE Custom Integrated Circuits Conference, San Jose, pp. 729–732 (2005)
128. Wang, Z., Lai, X., Roychowdhury, J.: PV-PPV: parameter variability aware, automatically extracted, nonlinear time-shifted oscillator macromodels. In: Proceedings of the Design Automation Conference DAC'07, Yokohama, pp. 142–147 (2007)
129. Yanzhu, Z., Dingyu, X.: Modeling and simulating transmission lines using fractional calculus. In: International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007), Shanghai, pp. 3115–3118 (2007). doi:10.1109/WICOM.2007.773

Part IV
Optimization

This section is devoted to the optimization of the hot spot benchmark example introduced by STMicroelectronics, which couples thermal and electrical effects. The PDAE theory for electro-thermal coupled systems has been introduced in Sect. 2.2.2. Corresponding simulation paradigms based on the Demonstrater Platform methodology can be found in Sect. 8.3. Chap. 7 discusses now how to embedd an optimization flow in an industrial environment to optimize Power-Mos circuits with respect to the peak current.

Chapter 7
Optimization Methods and Applications to Microelectronics CAD

Salvatore Rinaudo, Valeria Cinnera Martino, Franco Fiorante, Giovanni Stracquadanio, and Giuseppe Nicosia

Abstract In many areas of research and design, simulators are a crucial tool for optimizing the relevant features of devices and for determining the effect of parameter variations on the output of a given system. Many commercially available simulation tools in the microelectronics industry are endowed with optimization programs, usually based on Least Squares Methods coupled with a numerical solver for minimizing, such as the normal equations or gradient methods. However these optimization codes are strictly linked to the whole commercial simulation package and cannot be easily adapted to the various requirements of an industrial environment. This chapter aims at introducing the most important algorithms and methods which could be of interest to CAD engineers working in universities or in several microelectronics companies. The examples which will be illustrated are real industrial cases and the results obtained with the cascade of simulators used within STMicroelectronics will be presented.

7.1 Motivation

In many areas of research and design, simulators are a crucial tool for optimizing the relevant features of devices and for determining the effect of parameter variations on the output of a given system.

The simulators are used to replace a large amount of experiments and measurements which are necessary to take into consideration the deterministic and stochastic behaviour of the manufacturing processes of electronic devices.

S. Rinaudo (✉) • V. Cinnera Martino • F. Fiorante
STMicroelectronics, Stradale Primosole 50, 95121, Catania, Italy
e-mail: salvatore.rinaudo@st.com; valeria.cinnera@st.com; franco.fiorante@st.com

G. Nicosia • G. Stracquadanio
Department of Mathematics and Computer Science, University of Catania, V.le A. Doria 6, 95125, Catania, Italy
e-mail: nicosia@dmi.unict.it; stracquadanio@dmi.unict.it

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_7

A very important application of optimization is the *parameter extraction*, by which the parameters characterizing a given device within a given mathematical model can be obtained from the measurements of some characteristics of the device.

For instance the parameters required to model the behaviour of a device are related to physical quantities such as mobility, recombination and so on. When a device is described by an equivalent circuit, as for example the Gummel-Poon model, used by many circuit simulators (e.g. SPICE [36]), to perform the characterization of the electrical behaviour of bipolar transistors, the parameters are related to the electrical components of the circuit.

Many commercially available simulation tools in the EDA field of microelectronics industry are endowed with optimization programs, usually based on Least Squares Methods coupled with a numerical solver for minimizing, such as the normal equations or gradient methods. *However these optimization codes are strictly linked to the whole commercial simulation package and cannot be easily adapted to the various requirements of an industrial environment.*

For instance, in some industrial applications, a designer would like to have a global optimizer, which, being computationally expensive, is usually not available in the commercial package. However the greater computational cost of global optimization could be tolerable in an industrial context if efficient use be made of the computing power of a given design unit (e.g. by an appropriate use of a cluster of multivendors workstations). Another important example of great interest is the optimization of a cost function which is computed by using several simulation codes which are not integrated in a single software package and have been provided by different software vendors. Still another example would be the need to add more functions to the optimization (or parameter extraction) which are not usually found in commercial optimization software.

Because of these various demands for customized optimization and tolerance analysis in a CAD/CAM unit, several research groups in the microelectronics industry have developed their own optimization packages.

In particular the TCAD unit of STMicroelectronics has developed a post processing package called *EXEMPLAR* [33, 34] which encompasses global optimization and advanced statistical sensitivity analysis for the cascade of commercial and in house simulators, which are widely used by the design engineers throughout the company. The activity, started two decades ago in ST with the development of Exemplar, has been extended in the COMSON project.

This chapter aims at introducing for the most important algorithms and methods which are part of the optimization framework Exemplar which could be of interest to CAD engineers working in universities (graduate students) or in several microelectronics companies.

The examples which will be illustrated are *real industrial cases* and the results obtained with the cascade of simulators used within STMicroelectronics will be presented. This textbook is addressed to:

- Researchers in the microelectronics industry working in the CAD area. In particular those who must keep and update the commercial simulation software

used by CAD engineers; and must also integrate different commercial simulation software within optimization environments.
- PhD or advanced students in electrical engineering, computer science and applied mathematics.

7.2 The Optimization Problem

Optimization is the process by which one finds that value x that maximizes or minimizes a given function $f(x)$. The function$f$ is called *objective function.*

Except in linear case, optimization proceeds by iteration, that is, starting from an approximate trial solution, a good algorithm gradually refines the research space until a predetermined level of precision has been reached. An extremum of f (maximum or minimum point) can be either global or local.

Generally, the global extremum is required, even if to distinguish a local extremum from a global extremum is not so simple. A technique to determine the global minimum could be to vary the initial point and take as extremum the one among all that results to be the minimum or maximum in absolute (if they are not all equal). If necessary, a high number of initial points can be generated in a random way. Another technique could be to perturb a local extremum to verify if the algorithm gives again the same extremum. Relatively recent techniques such Simulated Annealing and Genetic Algorithm are designed to minimize functions that are not smooth and that may have many local minima. Simulated Annealing algorithms introduce a random element into the iteration process, giving the algorithm a change to escape from a local extremum. Genetic Algorithms carry information about multiple candidates for the global extremum that are simultaneously refined as iteration proceeds.

In the optimization field it is necessary to make another difference between **constrained optimization** and **unconstrained** optimization; we talk about constrained optimization when the x value which minimizes or maximizes f has to satisfy a priori one or more constraints.

Having established the optimization as unconstrained, we must choose an optimization method. First of all, we must choose between methods that need only evaluations of the function to be minimized and methods that also require evaluations of the derivatives of f. In the multidimensional case, this derivative is the gradient (∇f), that is the vector whose components are the partial derivatives of$f$ with respect to x_i for i $= 1, \ldots,$ n.

Algorithms using the derivative are somewhat more powerful than those using only the function, but not always enough so as to compensate for the additional calculations of derivatives. Another criterion to be taken into account, to choose an optimization method, is related to the quantity of memory it requires. We must choose between methods that require storage of order n^2 and those that require only of order n, where n is the number of dimensions. For moderate values of n and reasonable memory sizes this is not a serious constraint. There will be, however, the occasional application where storage may be critical. Among the methods which does not require the derivatives calculation and require a storage of order n^2, we have to focus more attention on **Nelder and Mead's Simplex Method** and on the **Powell's Method**. For as much as regards the algorithms which require the first derivatives calculation we can consider two major families of algorithms: **Conjugate Gradient Method** and **Quasi-Newton Methods**. The former requires only of order n storage, while the Quasi-Newton Methods require of order n^2 storage. Both families require a one-dimensional minimization sub-algorithm, which can itself either use, or not use, the derivatives calculation.

The *EXEMPLAR* optimizer widely used by ST design engineers, is mainly based on optimization methods without derivatives. Some of these methods are: the Simplex method [28], the Powell [31], the CRS (*Controlled Random Search*) [7, 32] and the Direct method [26].

In the present chapter we describe the integration of ST PAN modeling flow [4, 5] with the EXEMPLAR framework in which the innovative *Discretized Immune Algorithm* (DIA) is encapsulated (Fig. 7.4).

7.3 Parameter Extraction for Compact Circuit Models

High-Voltage discrete *power MOSFETs* robustness is hardly tied to the topology of the layout device. Weakness in layout designs could produce, for example, during a classic UIS (Unclamped Inductive Switching), current focusing known as "hot spots" [12, 27] that could compromise the integrity of the entire device. It will be shown that an automatic optimization of a *power MOSFET*. layout can reduce the peak currents focused in hot spot areas during an UIS. The device is modeled using the innovative Power Analyzer (*PAN*) technique [4, 5] based on an accurate extraction of a spice-like model of the power device starting from its physical layout representation. The optimization is based on a framework fully integrated within the PAN modeling flow, which utilizes many of the most used and effective *state-of-art* optimization algorithms [23].

7.3.1 Automatic Physical Layout Optimization of Discrete Power MOSFETs for Reducing the Effects of Current Density

Discrete power MOSFET device represents nowadays a class of power devices highly requested in the field of SMPS (Switched-Mode Power Supply) for Servers, Solar & Desktop, AC/DC Converters, Battery Chargers, etc. due to their minimized gate charge, high speed switching and lowest $R_{DS}(ON)$ (Static drain-source on resistance).

The basic internal structure of a power MOSFET is made up of several elementary transistor cells connected in parallel rows in order to achieve the current handling capability required by the design application. Each row, and hence each single cell, is powered by a gate metal path that branches from the gate pad and stretching itself across the entire device. In the layout, there are cells displaced in areas far and very near to the gate pad; the result is that the gate impedance of each cell seen towards the gate pad could vary in relation to its distance to the gate pad [2, 6]. As a consequence of this non uniformity, it is possible to observe during the turn-on or turn-off switching that some cells will receive or loose the gate signal at different times causing a different behavior in terms of current carried. For example, at high switching frequencies the time required by the gate signal to reach the farthest elementary cell may be comparable to the switching times of the input signal. Therefore, this provides fast switching times for cells near to the gate pad and increasing delays for those located furthest away. The fast turn off for only portions of the device forces the remainder to drive large amounts of current during switching [2]. The result is a dramatic current density increase in the slowest parts of the device causing what we call a "hot spot" . Hot spots are restricted areas where probable thermal failures can make devices less robust. Also, in a UIS turn-off, peaks of current forced by the inductor load will be carried only by those slower cells, therefore, a better distribution of the cells is necessary to minimize the peak current which is a constraint that the designer should consider in order to develop a more robust devices [8, 24, 25]. This target could be reached by modifying the distribution of the gate metal across the device but paying attention not to vary the gate impedance of the whole device. This is strongly tied to the displacement of the metal path which is often a given parameter requested by the customer.

A new optimization framework flow will be demonstrated using new optimization algorithms that will modify the geometry of the layout. Moving element positions in the layout can cause an increase or decrease of the current density in the device depending on where elements are placed. Power Mos is made up of elements such as metal gate, fingers, gate pad, source windows, etc. which is shown in Fig. 7.1.

For example, moving the metal fingers closer together can cause current to reach some elementary MOS gates quicker and slower for others because the path traveled may be decreased or increased. The number of relative placements for elements can be enormous, therefore, a new optimization tool is designed to aid this process.

7.3.2 *Modeling Approach*

In order to better understand how the whole flow works, we will focus our attention on a spice-like netlist model used in the flow. This model is produced by an extraction starting from a CAD *(Computer Aided Design)* layout view of the device. Since, the layout usually has many different CAD layers that are not useful during

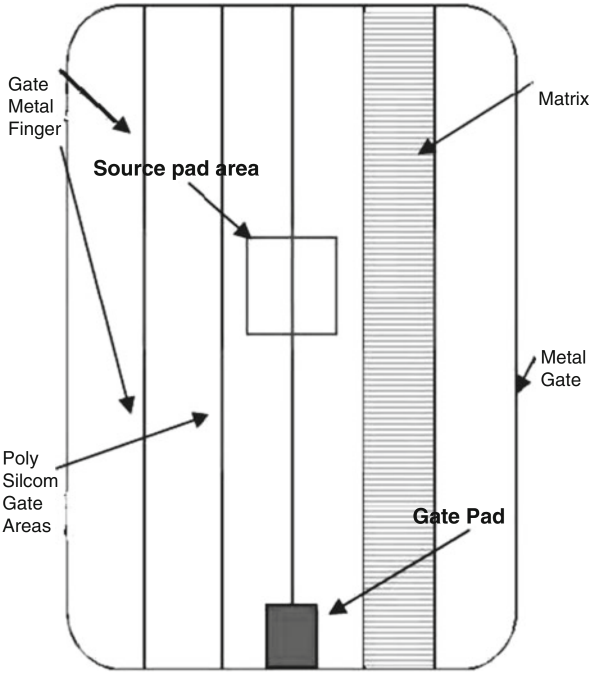


Fig. 7.1 Simplified layout of the Device under investigation

the final extraction. A simplified CAD view of the device is effectively used by the modeling tool as input.

The extracting tool, PAN is used to extract the netlist of the device in two steps from the simplified view.

The first step is to extract a numerical model which is strictly related to the layout topology. Each layer is mapped to a number, for example, an elementary cell is represented by the number 2. The second step starts from the numerical model which automatically produces the spice-like netlist model through indexes interpretation. The cell named as “MOS” corresponds to the model of a single elementary cell which must be supplied by the user and it could also be extracted with the aid of a TCAD *(Technology CAD tool)* simulation tool or by measurements on a wafer.

The peculiarity of this spice-like modeling flow is once the numerical model is produced, then the final netlist could be easily extracted starting from this data. This very important possibility releases the designer from the original layout so that many more analysis can be performed simply by modify the numerical model written in ASCII format which implies modification of the physical layout structure (es. number of fingers, pad size, position, etc.).

Since the position of elements has been discretized such that they are put on a 2D lattice with integer coordinates, the algorithms have to find the x and y coordinates for each element of the circuit under observation.

It is important to outline that any algorithm that is able to work with discrete variables are suitable for this problem, because it can be tackled as a *black box* *optimization* problem. In our work, we use, through the Exemplar optimizer, four algorithms that are the *state-of-art* in black box and circuit design optimization: in particular, we use Controlled Random Search (CRS), and Controlled Random Search Enhanched (CRS-E) [7]. The modelling methodology has been evaluated in several of its aspect and a U.S. Patent has also been deposited [5].

7.4 The Optimization Algorithm

Designing micro-electronic devices is a complex process, which takes into account increasing frequency and bandwidth ranges, small size factor, high reliability and low power consumption [3, 20, 29]. From a mathematical point of view, there are three major aspects to take into account; the formulation of a formal model of the system, the performance optimization and the robustness analysis. In this chapter, we focus on finding an optimal design for the power MOSFET. The space of solutions defined by the parameters of the power MOSFET is enormous and it is highly rugged. Moreover, the power MOSFET is a complex device which simulation requires ∼5 min; due to this expensive computational cost, an optimization algorithm has to find good solution using tight budget of simulations [3, 20, 29]. In order to tackle effectively this optimization problem, we design a new OPTIMIZATION IMMUNOLOGICAL ALGORITHM, called OPTIA [13, 18]. The OPTIA is a stochastic optimization algorithm based on the Human Clonal Selection Principle of the Immune System (IS) [1, 16, 17].

The IS is an excellent example of bottom-up optimization strategy through which adaptation operates at the local level of cells and molecules, and useful behavior emerges at the global level [15, 19]. In particular, the Clonal Selection theory shows that B and T lymphocytes, that are able to recognize the antigen, will start to proliferate by cloning upon recognition of such antigen. When a B cell is activated by binding an antigen, many clones are produced in response via the so called *clonal expansion*. The newly created cells can undergo to somatic hypermutation, creating offspring B cells with mutated receptors: the higher the affinity of a B-cell to the antigens, the more likely it will clone. This results in a Darwinian process of variation and selection, called *affinity maturation*. The increase in size of these populations couples with the production of cells with longer than expected lifetimes, assuring the organism a higher specific responsiveness to that antigenic attack in the future: the so called *immunological memory of the system*.

OPTIA tries to mimic the clonal selection principle for optimization: a problem is an antigen and a B-cell is a candidate solution. The affinity between an antigen and a B-cell is given by the objective function of the optimization problem [13, 14, 35].

Each B-cell is a vector of real values of dimension n, where n is the dimension of the problem; moreover each candidate solution has associated an age τ: it indicates the number of iterations since the last successful mutation [9, 13, 21]. Initially the age is set to zero.

Fig. 7.2 Pseudo-code of the Optimization Immune Algorithm (optIA)

```
procedure OPTIA($d; dup; \tau_B; \rho; \beta; s_a$)
    $t \leftarrow 0$
    $BC_{arch} \leftarrow Create\_Archive(s_a)$
    $P^{(t)} \leftarrow Initialize(d)$
    $Evaluate\,(P^t)$
    while $\neg Termination\_Condition()$ do
        $P^{(clo)} \leftarrow Cloning(P^{(t)}; dup)$
        $P^{(hyp)} \leftarrow Hypermutation(P^{(clo)}; \rho)$
        $P^{(macro)} \leftarrow Macromutation(P^{(hyp)}; \beta)$
        $Evaluate(P^{(macro)})$
        $Aging(P^{(t)}; P^{(macro)}; \tau_B)$
        $P^{(t+1)} \leftarrow Selection(P^{(t)}; P^{(macro)}; BC_{arch})$
        $t \leftarrow t+1$
    end while
end procedure
```

An initial population $P^{(0)}$ of dimension d is generated randomly, with each variable constrained into its lower and upper bounds. However, it could be useful to use an ad-hoc population to start the optimization process: optIA can take in input a *starting point* p_{st}, and it use this point to initializes one B-cell of the population and the remaining $d-1$ candidate solutions are initialized with vectors obtained as a perturbation of p_{st}.

The algorithm is iterative: each iteration is made of a cloning, mutation and selection phase [9, 10, 30]. The algorithm stops when a given stopping criterion is verified: in particular, it ends when a maximum number of fitness function evaluations is reached.The pseudo-code of the algorithm is shown in Fig. 7.2.

The cloning phase is responsible for the production of new B-cell. Each B-cell is cloned *dup* times producing a population $P_{N_c}^{(clo)}$ of size $d \cdot dup = N_c$, where each cloned B-cell takes the same age of its parent. At the same time, the age of the parent is increased by one.

After the $P^{(clo)}$ population is created, it undergoes to the mutation phase in order to find better solutions. In the mutation phase, the hypermutation and hypermacromutation are applied to each candidate solutions [11]. These operators are the principle responsible of the exploring and exploiting ability of the algorithm.

The hypermutation operator is based on the *self-adaptive gaussian mutation* that is computed by:

$$\sigma_i' = \sigma_i * exp((\tau * N(0,1)) + (\tau' * N_i(0,1))) \tag{7.1}$$

$$x_i^{new} = x_i + \sigma * N(0,1). \tag{7.2}$$

The hypermacromutation applies a convex mutation to a given solution according to the following equation:

$$x_i^{new} = (1-\beta) * x_i + \beta * x_k; \tag{7.3}$$

where $x_i \neq x_k, \beta \in [0.1]$ is a random number obtained with uniform distribution. Since variables x_i and x_k typically have different ranges, the value x_k is normalized within the range of x_i using the following equation:

$$x_k^{norm} = L_i + \frac{(x_k - L_k)}{(U_k - L_k)} \times (L_i - U_i) \tag{7.4}$$

where L_i, U_i are the lower and upper bounds of x_i and L_k, U_k are lower and upper bounds of x_k. The value used to mutate the variables x_i is then x_k^{norm}.

The mutation operators are controlled by a mutation rate α that is differently defined according to the type of operator: for the hypermutation operator, it is defined as

$$\alpha = e^{(-\rho \times f)} \tag{7.5}$$

instead for the macromutation operator is defined as

$$\alpha = (\frac{1}{\beta}) \times e^{(-f)} \tag{7.6}$$

where f is the fitness function value normalized in [0, 1].

These operators are applied sequentially: the hypermutation operator acts on the $P^{(clo)}$ and it produces a new population $P^{(hyp)}$ and the hypermacromutation mutate the $P^{(hyp)}$ generating the $P^{(macro)}$ population.

The population $P^{(macro)}$ is evaluated: if a B-cell achieves a better objective function value, its age is set to zero otherwise it is increased by one.

The *Aging* operator is executed on $P^{(t)}$ and $P^{(macro)}$: it drops all the B-cells with age greater than $\tau_b + 1$, where τ_b is a parameter of the algorithm.

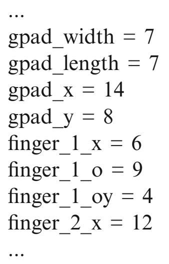
The B-cells deleted by the AGING operator are not discarded but they are saved into an archive BC_{arch} of size s_a: if there is enough space into the archive, the B-cell is saved into the first available location, otherwise a random location of the archive is selected and it is substituted by this new B-cell.

The selection is then performed and the new $P^{(t+1)}$ is created by picking the best individuals from the parents and the mutated B-cells: if $|P^{(t+1)}| < d$, $d - |P^{(t+1)}|$ B-cells are randomly taken from the archive and added to the new population.

Many real world problems associate to each variable of an optimization problem a fineness parameter in addiction to lower and upper bounds.

OPTIA is able to handle this kind of variable using a *grid based model*: a grid can be defined as an n-dimensional space which fineness is specified by a parameter δ.

When OPTIA evaluate the fitness of a B-cell, it projects the solution on the grid and successively it evaluates the fitness of the projected point. It is possible to define different strategies to project the point on the grid. For each variable of the problem,



```
...
gpad_width = 7
gpad_length = 7
gpad_x = 14
gpad_y = 8
finger_1_x = 6
finger_1_o = 9
finger_1_oy = 4
finger_2_x = 12
...
```

Fig. 7.3 Extraction of the description file

OPTIA uses the following projection equation:

$$\Pi(x, l, u, \delta) = min(|x - x_{\delta}^{1}|, |x_{\delta}^{2} - x|) \tag{7.7}$$

$$x_{\delta}^{1} = \frac{x - l}{\delta} \tag{7.8}$$

$$x_{\delta}^{2} = \frac{u - x}{\delta} \tag{7.9}$$

where x is a variable of the problem, l, u are the relative lower and upper bounds and δ is the mesh fineness parameter.

7.4.1 The Optimization Flow

The methodology proposed is based on the possibility of modifying the device layout structure by simply modifying the contents of the numerical model stored in an ASCII file. Before using the numerical model in the optimization flow, it must be translated into a text description format using a custom language named DES [1] which is based on a series of parameters that exhaustively describes the content of the numerical model.

The DES file automatically created by custom software starting from the numerical model allows for a better definition of constraints that the optimizer must take into consideration. The DES parameters are geometric locations of elements within the power Mos layout, for example, the location of the finger 1 is given by the x,y coordinates and the width parameters such as finger_1_x, finger_1_oy and finger_1_o respectively. These parameters are setup as variables which are inputs to the optimizer that can be modified by the optimizer algorithm to find the optimal location. The same is true for each of the other parameters in Fig. 7.3.

The optimization flow works accordingly as shown in Fig. 7.4. Firstly, a reference starting geometric layout must be provided. This geometric layout is simplified into

[1] DES name used to describe the description of the numerical model matrix.

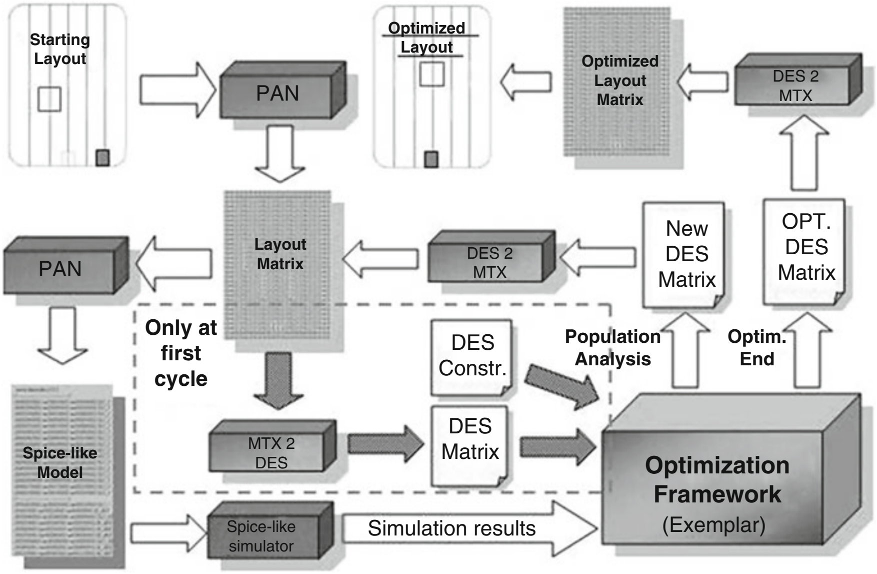


Fig. 7.4 IO Optimization Flow

a numerical mapped model of the layout which is extracted by the PAN tool so that it can be converted using DES2MTX[2] software into a detailed text description for each of the important elements. The text description contains the positional coordinates and sizes of each element which describes the make up of the discrete power MOS. These elements positions and sizes become the parameters which may be optimized in order to resolve the current density problem which has been previously described. Secondly, the detailed text description is provided as input to the optimization framework as shown in Fig. 7.3 along with constrained bounds for each elements parameters under evaluation which is provided on the first iteration cycle and used through out the entire optimization process. Also, at the same time the PAN tool also converts the numerical mapped layout into a spice like netlist which is simulated using a third party tool that is a circuit simulator such as Mentor Graphics Eldo [22] or a fast spice simulator; the results of the performances or simulation results are passed to the optimization framework as inputs. Next, the optimization algorithm goes to work by first reading these inputs and lastly providing new elements parameters values under evaluation. If the new values under optimization are not within the given constrained bounds (not yet optimal), a new text description file provided by the optimization framework is converted back to a numerical model by the DES2MTX software. Then it is converted to a new spice like netlist again,

[2] DES2MTX is software to convert description language to numerical mapped data of the PowerMos layout. At the same manner, also the Mtx2Des software has been developed.

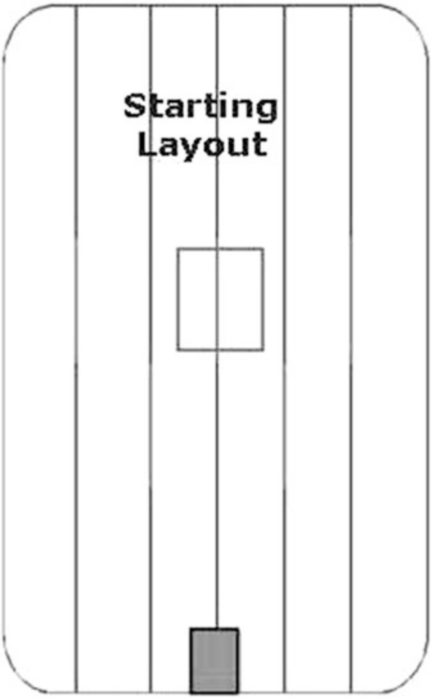


Fig. 7.5 Starting Layout

which is then simulated, and delivered to the optimization frame work. This flow is cycled until optimal solutions are found or the maximum number of iterations has been achieved by the optimizer, even if a solution cannot be obtained. However, if an optimal solution has been found, the text description file provided by the optimization framework is converted back to a numerical model by the DES2MTX. Finally, the PAN tool converts the new numerical model to its original geometric layout form as the optimized physical layout view.

7.4.2 Simulation Results

In this work, we used the optimization flow described in order to improve the robustness of power MOSFET devices in terms of maximum current density allowed in a UIS turn-off.

Taken as reference the starting layout of Fig. 7.5, we obtained as result of the optimization a better new layout where that maximum currentpeak is lower than the one in the starting layout. In the built spice-like model of the device, each single cell represents a portion of area of the real device; the size of that area depends on how dense the mesh has been chosen for the discretezation necessary to extract the matrix. An evaluation of the drain current referred to the elementary

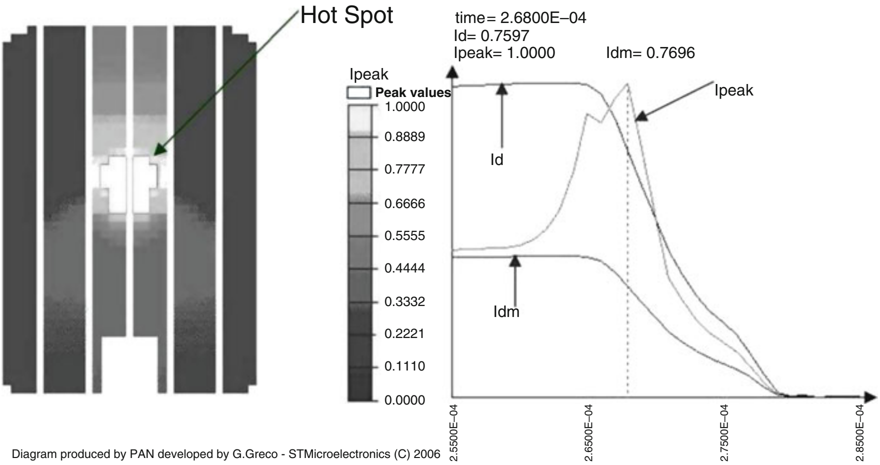


Fig. 7.6 Ipeak current 2D map and waveforms in a UIS turn-off for the starting reference layout

cell and, hence, to a given area, will give us very useful current density information. For technologist, such as parameter is known and represents a physical limit for the technology even if other causes of failure are to be searched in an excess of the silicon temperature reached due to the switching power and of some parasitic bipolars triggered by the high slopes of the drain currents flowing across the elementary cells [6]. All these aspects could be easily investigated using the PAN tool together with the optimization flow but in this work, only matters tied to peak currents are investigated and result presented. In figure 2D map and waveforms referred to the Ipeak currents across the reference starting layout is reported in Fig. 7.6 . The light area on dark background represents regions of the devices where currents are higher.

That 2D map gives information useful from the qualitatively point of view. Quantitatively information, instead, are given by the waveforms on the right part for Fig. 7.6 which reports the whole Idrain current across the drain terminal of the devices I_d, the average current I_{dm} that should have been if each cell of the devices would have switched ideally without gate/source delay and, at end, the maximum peak current found across all the cells of the devices. The difference between I_{peak} and I_{dm} represents an index of current unbalancing of the device. To study the effective of immune algorithm and two investigate the hardness of the problem, we have conducted a long series of simulation using various optimization algorithm: we have used the SIMPLEX, CRS and CRS-E methods. In our experimental protocol, when it is possible, we use as a starting point the circuit proposed by the designers or we generate one randomly. In particular, OPTIA can work completely *from scratch* or take an initial point, that is assigned to an individual and the other individuals of the population are small perturbations of this starting point.

The stopping criterion adopted is the attainment of a fixed number of simulations (in our case 10^4) or, for numerical methods, the achievement of the convergence.

Table 7.1 PowerMosfet optimization results using evolutionary and numerical algorithms

Algorithm	Initial point	Ipeak
Designer's point	–	9.948×10^{-3}
optIA	**Random point**	$\mathbf{7.281 \times 10^{-3}}$
Crs	Random point	7.296×10^{-3}
optIA	Designer's point	7.394×10^{-3}
Crs-E	Random point	7.397×10^{-3}
Simplex	Designer's point	7.894×10^{-3}

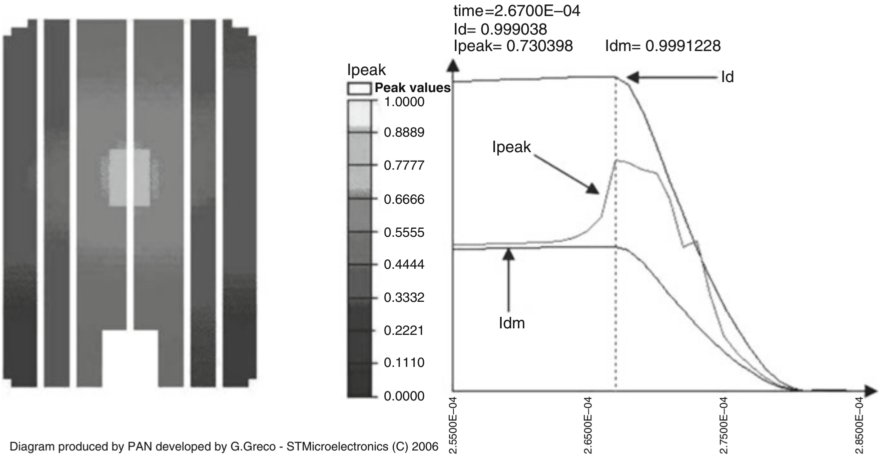


Diagram produced by PAN developed by G.Greco - STMicroelectronics (C) 2006

Fig. 7.7 Ipeak current 2D map and waveforms in a UIS turn-off for the optimized layout

In Table 7.1, we report for each algorithm the best solution found in terms of IPEAK by the various algorithms: by inspecting the results, OPTIA found the best solution using random points. From an optimization point of view, this is not surprising because the designer's circuit can be a local optimum that prevent the algorithm to find new good solutions. The 2D final current map and relative waveforms have been shown in Fig. 7.7. As it is possible to observe, during the analyzed range of time, where the current peak occurs, the final value for that parameter is normalized decreasing from 1.0000 to 0.73098 obtaining an improvement of approximately 27 %. In order to obtain this new result the optimizer has modified the geometry of the original layout by repositioning and modifying elements in the layout. The optimizer has decreased the size of poly silicon gate areas and repositioned them along the gate metal finger. It also repositioned the gate metal fingers in horizontal fashion. The final result is a new layout geometrically adjusted and optimized for the better (Fig. 7.8).

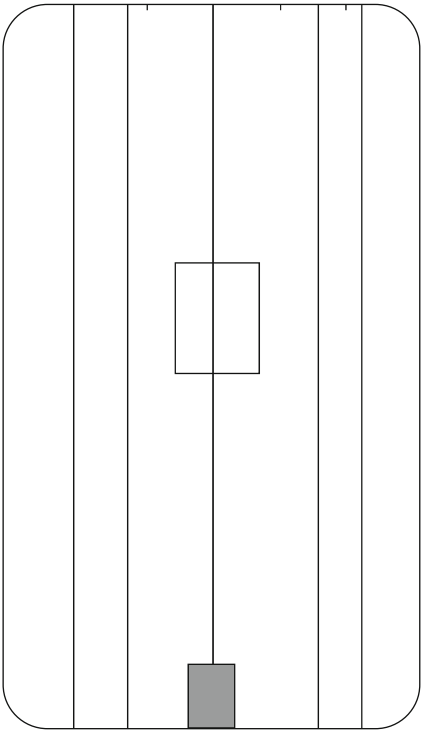


Fig. 7.8 Optimized Layout

7.5 Conclusion

In this work we have shown an innovative methodology for the power MOSFET design aimed to improve robustness and performances. This methodology opens new spaces in power device designing giving to the designers new innovative CAD tools that allows investigating problematic until now little afforded due to the lack of means. The work, yet in the preliminary phase has shown enormous potential for investigation in future work, and of course, will be treated. The designed evolutionary algorithm was shown to produce acceptable solutions in most cases, where classical techniques failed.

References

1. Anile, A., Cutello, V., Nicosia, G., Rascuna, R., Spinella, S.: Comparison among evolutionary algorithms and classical optimization methods for circuit design problems. In: The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, vol. 1, pp. 765–772 (2005)
2. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S.: Analysis of the internal current distribution in power mosfets operated at high switching frequency. In: Proceedings of MSED,Workshop on Modeling and Simulation of Electron Devices, Pisa, vol. 15, pp. 4–5 (2005)

3. Biondi, T., Ciccazzo, A., Cutello, V., D'Antona, S., Nicosia, G., Spinella, S.: Multi-objective evolutionary algorithms and pattern search methods for circuit design problems. J. Univers. Comput. Sci. **12**(4), 432–449 (2006)
4. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S.: Effect of layout parasitics on the current distribution of power mosfets operated at high switching frequency. J. Comput. Electron. **5**(2–3), 149–153 (2006)
5. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S.: Method for modeling large-area transistor devices, and computer program product therefore. U.S. Patent N. 11/770,578 deposited in (2007)
6. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S., Allia, M., Liotta, S.: Distributed modeling of layout parasites in large-area high-speed silicon power devices. IEEE Trans. Power Electron. **22**(5) (2007)
7. Brachetti, P., De Felice Ciccoli, M., Di Pillo, G., Lucidi, S.: A new version of the Price's algorithm for global optimization. J. Global Optim. **10**(2), 165–184 (1997)
8. Budihardjo, I., Lauritzen, P., Mantooth, H.: Performance requirements for power MOSFET models. In: 25th Annual IEEE Power Electronics Specialists Conference, PESC'94 Record, Taipei, pp. 69–76 (1994)
9. Castrogiovanni, M., Nicosia, G., Rascuná, R.: Experimental analysis of the aging operator for static and dynamic optimisation problems. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 804–811. Springer-Verlag Berlin, Heidelberg (2007)
10. Ciccazzo, A., Halfmann, T., Marotta, A., Nicosia, G., Rinaudo, S., Stracquadanio, G., Venturi, A.: New coupled EM and circuit simulation flow for integrated spiral inductor by introducing symbolic simplified expressions. In: IEEE International Symposium on Industrial Electronics, ISIE 2008, Cambridge, pp. 1203–1208 (2008)
11. Conca, P., Nicosia, G., Stracquadanio, G., Timmis, J.: Nominal-Yield-Area Tradeoff in Automatic Synthesis of Analog Circuits: A Genetic Programming Approach Using Immune-Inspired Operators. In: 2009 NASA/ESA Conference on Adaptive Hardware and Systems, San Francisco, pp. 399–406. IEEE (2009)
12. Consoli, A., Gennaro, F., Testa, A., Consentino, G., Frisina, F., Letor, R., Magri, A.: Thermal instability of low voltage power-mosfets. IEEE Trans.Power Electron. **15**, 575–581 (2000)
13. Cutello, V., Nicosia, G.: An immunological approach to combinatorial optimization problems. In: Advances in Artificial Intelligence – IBERAMIA, Seville, pp. 361–370 (2002)
14. Cutello, V., Krasnogor, N., Nicosia, G., Pavone, M.: Immune Algorithm Versus Differential Evolution: A Comparative Case Study Using High Dimensional Function Optimization. Adaptive and Natural Computing Algorithms, ICANNGA 2007, April 11-14, 2007, Warsaw, Poland. Springer, Lecture Notes in Computer Science 4431:93–101 (2007)
15. Cutello, V., Lee, D., Leone, S., Nicosia, G., Pavone, M.: Clonal Selection Algorithm with Dynamic Population Size for Bimodal Search Spaces. Advances in Natural Computation, ICNC 2006, September 24-28, 2006, Xi'an, China. Springer, Lecture Notes in Computer Science 4221:949–958 (2006)
16. Cutello, V., Narzisi, G., Nicosia, G., Pavone, M.: An immunological algorithm for global numerical optimization. In: Artificial Evolution, pp. 284–295. Springer-Verlag Berlin, Heidelberg (2006)
17. Cutello, V., Nicosia, G., Pavia, E.: A parallel immune algorithm for global optimization. In: Intelligent Information Processing and Web Mining, IIS 2006, June 19-22, 2006, Ustron, Poland. Springer, Series on Advances in Soft Computing, pp. 467–475 (2006)
18. Cutello, V., Nicosia, G., Pavone, M.: Exploring the capability of immune algorithms: A characterization of hypermutation operators. In: Artificial Immune Systems, ICARIS 2004, September 13-16, 2004, Catania, Italy. Springer, Lecture Notes in Computer Science 3239:263–276 (2004)
19. Cutello, V., Narzisi, G., Nicosia, G., Pavone, M.: Real coded clonal selection algorithm for global numerical optimization using a new inversely proportional hypermutation operator. In: SAC 2006, Dijon, vol. 2, pp. 950–954. ACM (2006)

20. Cutello, V., Nicosia, G., Rascuna, R., Spinella, S.: Optimising an inductor circuit and a two-stage operational transconductance amplifier using evolutionary and classical algorithms. International J. Comput. Sci. Eng. **2**(3), 158–169 (2006)
21. Cutello, V., Nicosia, G., Romeo, M., Oliveto, P.: On the convergence of immune algorithms. In: IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, Honolulu, pp. 409–415 (2007)
22. Eldo user's manual. Mentor Graphics Corporation (2007)
23. Graeb, H.: Analog Design Centering and Sizing. Springer Netherlands, Dordrecht (2007)
24. Hohl, J., Galloway, K.: Analytical model for the single event burnout of power MOSFETs. IEEE Trans. Nucl. Sci. **34**, 1275–1280 (1987)
25. Hu, C., Chi, M., Patel, V.: Optimum design of power MOSFET's. IEEE Trans. Electron Devices **31**(12), 1693 (1984)
26. Jones, D., Perttunen, C., Stuckman, B.: Lipschitzian optimization without the Lipschitz constant. J. Optimiz. Theory Appl. **79**(1), 157–181 (1993)
27. Kraus, R., Mattausch, H.: Status and trends of power semiconductor device models for circuit simulation. IEEE Trans. Power Electron **13**, 452–465 (1998)
28. Nelder, J., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308 (1965)
29. Nicosia, G., Rinaudo, S., Sciacca, E.: An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization. In: Research and Development in Intelligent Systems XXIV, 10-12 December 2007, Cambridge, England, UK. Springer, pp. 7–20 (2007)
30. Nicosia, G., Rinaudo, S., Sciacca, E.: An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization. In: Research and Development in Intelligent Systems XXIV, 10-12 December 2007, Cambridge, England, UK. Springer, pp. 7–20 (2007)
31. Powell, M.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput. J. **7**(2), 155 (1964)
32. Price, W.: A controlled random search procedure for global optimisation. Comput. J. **20**(4), 367 (1977)
33. Rinaudo, S., Moschella, F., Anile, A.M., O.Muscato: Controlled random search parallel algorithm for global optimization with distributed processes on multivendor cpus. In: Arkeryd, L., et al. (eds.) Progress in Industrial Mathematics – ECMI 98, Gothenburg. B.G. Teubner, Stuttgart/Leipzig (1999)
34. Rinaudo, S., Moschella, F. & Anile, A., M.: Parallel implementation in an industrial framework of statistical tolerancing analysis in microelectronics. In: EURO-PAR'99 Parallel Processing. Lecture Notes in computer Science, vol. 1685. Springer, Berlin/Heidelberg (1999)
35. Tarakanov, A., Nicosia, G.: Foundations of immunocomputing. In: IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, Honolulu, pp. 503–508 (2007)
36. Vladimirescu, A.: The SPICE book. Wiley, New York (1994)

Part V
COMSON Methodology

The COMSON methodology is based in the linkage of a Demonstrator Platform (Chap. 8) with an e-learning environment (Chap. 9). It is used for both testing mathematical modells and methods derived in Chaps. 2–7 and educating young researchers.

Chapter 8
COMSON Demonstrator Platform

Georg Denk, Tamara Bechtold, Massimiliano Culpo, Carlo de Falco, and Alexander Rusakov

Abstract This chapter describes the *Demonstrator Platform* (DP), a framework for simulation of devices, interconnects, circuits, electromagnetic fields, and thermal effects. This framework is used to develop and test new mathematical methods and algorithms. Section 8.1 describes the design of the DP and gives an overview of the available modules. Section 8.2 is a tutorial on how to use the DP focusing on model-order reduction. It shows for the example of a micro-hotplate model all the steps needed to apply model-order reduction, including postprocessing and error estimation. In a second part, a coupled simulation of a circuit combined with a reduced model of a transmission line is presented. Section 8.3 emphasizes the aspect of the DP as a development framework. After introducing the benchmark example of an n-channel power MOS-FET, it is shown how to combine and extend different modules of the DP to a fully coupled electro-thermal simulation of the device.

G. Denk (✉)
Infineon Technologies AG, 81726 München, Germany
e-mail: georg.denk@infineon.com

T. Bechtold
IMTEK – University of Freiburg, Georges-Koehler-Allee 103, 79110 Freiburg, Germany
e-mail: tamara.bechtold@imtek.uni-freiburg.de

M. Culpo
Chair of Applied Mathematics/Numerical Analysis, Bergische Universität Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany
e-mail: m.culpo@cineca.it

C. de Falco
MOX – Modeling and Scientific Computing, Dipartimento di Matematica, Politecnico di Milano, P.zza L. da Vinci 32, 20133 Milano, Italy

CEN – Centro Europeo di Nanomedicina, P.zza L. da Vinci 32, 201333 Milano, Italy
e-mail: carlo.defalco@polimi.it

A. Rusakov
Institute for Design Problems in Microelectronics of Russian Academy of Sciences (IPPM RAS), 3, Sovetskaya Street, Moscow 124365, Russian Federation
e-mail: rusakov@inm.ras.ru

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_8

8.1 Introduction

The purpose of the COMSON project is to develop algorithms for coupled multiscale simulation and optimization in nanoelectronics. As several nodes are involved, a common platform for this development is needed. The main objective of the consortium is therefore to realize an experimental *Demonstrator Platform* in software code, which comprises simulation of devices, interconnects, circuits, electromagnetic fields, and thermal effects in one single framework. It connects each individual achievement, and offers an adequate simulation tool for optimization in a compound design space.

The *Demonstrator Platform* is used as a framework to test mathematical methods and approaches, so as to assess whether they are capable of addressing the industry's problems, and to adequately educate young researchers by hands-on experience on state-of-the-art problems, and beyond. The *Demonstrator Platform* does not aim at replacing existing industrial or commercial codes. However, it will be capable of analyzing medium sized coupled problems of industrial relevance, thus offering a chance to develop advanced mathematics for realistic problems. The second benefit of such a platform is to collect the knowledge of models and methods, which is widespread distributed over the different partners, giving a good opportunity for transfer of knowledge.

This section gives an introduction to the ideas and concepts of the *Demonstrator Platform*, followed by a tutorial section on how this platform can be used in the context of model-order reduction (Sect. 8.2). In Sect. 8.3, a research study is presented which uses the *Demonstrator Platform* for development of the coupled electro-thermal simulation of an n-channel power MOSFET. The *Demonstrator Platform* was also used for developing a coupled circuit-device simulation, see [21].

8.1.1 Design of the Demonstrator Platform

In order to allow an easy installation and application within different environments, the design of the *Demonstrator Platform* was influenced by the following assumptions:

- Provides a fast prototyping environment,
- Is not restricted to a particular operating system,
- Can easily be extended,
- Can easily be distributed to others,
- Allows different license conditions for different parts.

These conditions are especially important, if the *Demonstrator Platform* should be used both in an academic environment and in an industrial environment, as they are present within COMSON. To foster cooperations with other research groups, it is essential to share the *Demonstrator Platform* as a common development platform.

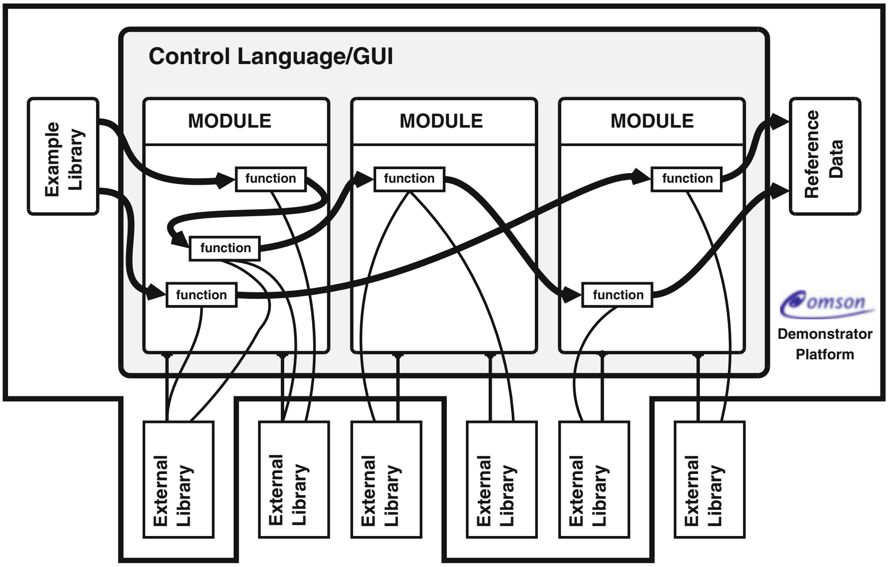


Fig. 8.1 Layout of the *Demonstrator Platform*

For fast prototyping of mathematical algorithms, interpreted languages like Matlab or Octave are widely used. To avoid license problems, we decided to use the free tool Octave [38] which is available for many operating systems and can be distributed without license issues. Octave allows both an interactive approach for development and a batch-oriented usage for long-running computations. It offers additionally a free API for integrating software written in other programming languages like C or Fortran. Octave builds the controlling language of the *Demonstrator Platform*.

The *Demonstrator Platform* uses a modular structure consisting of so-called *modules* and *external libraries*. Modules provide some functionality to the user of the *Demonstrator Platform*, e.g. model-order reduction techniques. To enable the re-use of already existing software, the modules may interface to external libraries, e.g. numerical libraries like Slicot. In Fig. 8.1, the structure of the *Demonstrator Platform* is depicted.

This flexible concept of external libraries provides a solution for incorporating software released under different license conditions. Libraries with a free license can be completely integrated to the *Demonstrator Platform* which is distributed as free software. Other libraries with a more restrictive license have to be kept separated, only the interfacing routines are part of the *Demonstrator Platform*. With this approach it is possible to call confidential software from the *Demonstrator Platform*, without making it a part of it. Especially in an industrial environment, this is of great importance. This is indicated in Fig. 8.1, where the boundary of the *Demonstrator Platform* includes some of the external libraries, while others are located outside.

For quality software, it is not enough to provide a set of functions. Also the documentation is an essential part of the *Demonstrator Platform*. This documentation contains research papers where the theoretic background of the routines is provided. In addition, a description of the available functions is automatically generated out of the sources.

The *Demonstrator Platform* contains an integrated self test to ensure the correctness of the modules. For this purpose every module is provided with some test examples for which a reference solution is known. During the self test, the computed solution is compared with the reference solution and an according message is printed. This feature is especially important during the joint development of new modules, as they might interfere with existing modules.

All examples, including the test examples, are categorized in so-called *class A*, *class B*, or *class C* examples. While *class A* examples are basic unit tests, *class B* examples are (simple) academic examples which already require a full-fledged algorithm to be solved. *Class C* examples are real-life examples which need the proper coupling of algorithms, advanced approaches, and in most cases longer computing times to be solved.

8.1.2 Modules

This section gives a short overview of the available modules of the *Demonstrator Platform*.

8.1.2.1 Generic Numerical Methods

DAEN Differential-Algebraic Equations Solver
DAEN is a BDF implementation of a differential-algebraic equations solver for problems with index 0, 1, and 2. It uses a variable step size, variable order.

RADAU Differential Algebraic Equation Solver
RADAU is a differential-algebraic equation solver for equations of index 0, 1 and 2. This code is a variable step size, variable order implementation of the RADAU methods.

GLIMDA Differential Algebraic Equations Solver
GLIMDA is for the numerical solution of differential equations of index 0, 1 and 2. It is a variable step size, variable order implementation of general linear methods.

BIM Finite-Element Box Integration Method
BIM is a PDE solver using a finite element/finite volume approach. It solves diffusion-advection-reaction (DAR) partial differential equations based on the finite volume Scharfetter-Gummel (FVSG) method also known as Box Integration Method (BIM).

8.1.2.2 Model-Order Reduction

AMOR Interface to MOR4ANSYS
AMOR provides an interface to the software package MOR for ANSYS which is part of the module MOR4ANSYS

MOR4ANSYS MOR for ANSYS
MOR4ANSYS is an adapted version of the software package MOR for ANSYS [44] which reduces dynamic systems provided in Matrix Market format.

ROM-WB Tools for migration between RomWork and OCS
ROM-WB provides some tools for converting input files to make them usuable for the module OCS and RomWork.

SLICOTINT Interface to SLICOT Model Reduction Routines
SLICOTINT provides an interface to the SLICOT library [49]. It supports balance and truncate model reduction (routine AB09AD), singular perturbation approximation based model reduction (routine AB09BD), Hankel norm approximation based model reduction (routine AB09CD). It also allows the efficient computation of Hankel Singular Values of the dynamical system.

MOR Collection of simple MOR tools
MOR provides a collection of simple projector matrix builders for DAE systems.

8.1.2.3 Circuit and Device Simulation

OCS Octave Circuit Simulator
OCS is a module for solving DC and transient MNA equations stemming from electrical circuits.

SKIF Interface to the SiMKit library
SKIF provides an interface to SiMKit library [37], a library of compact transistor models.

D4MEKAI Device Simulator based on Maximum Entropy Principle
D4MEKAI provides a device simulation of the hydrodynamical model of semiconductors in 1D in case of the Kane dispersion relation.

ETMEP1D Device Simulator of the MEP energy-transport model in 1D
ETMEP1D provides the 1D simulation of the MEP energy-transport model for semiconductors by using a Scharfetter-Gummel like scheme while the Poisson equation is solved with a false transient method.

ET_MEP_MOSFET Device Simulator for the 2D MEP energy-balanced model.
ET_MEP_MOSFET provides the 2D numerical simulation of the MEP energy-transport model for semiconductors by using the Scharfetter-Gummel scheme while the Poisson equation is solved with a false transient method [41–43].

FIDES Field Device Simulator
FIDES provides a package for monolithic or co-simulation of field-circuit coupled problems.

ROMI Reduced Order Model of the multiconductor interconnects
ROMI provides routines for extraction of the reduced order model of the multiconductor interconnects.

SECS1D, SECS2D, SECS3D Drift-Diffusion simulator for 1d, 2d, and 3d semiconductor devices
SECS1D, SECS2D, and SECS3D provides a device simulator for 1d, 2d, and 3d, resp., semiconductor devices based on drift-diffusion [22, 24, 25].

8.1.2.4 Optimization

CRS Optimization routine based on Controlled Random Search
CRS provides an optimization algorithm based on controlled random search plus some test functions.

OPTBOOK Optimization procedures
OPTBOOK provides a set of optimization procedures, described in the book "Optimization of the EM devices".

8.1.2.5 Auxiliary Routines

MSH Meshing Software Package
MSH is a package for creating and managing triangular and tetrahedral meshes for finite-element or finite-volume PDE solvers.

FPL FEM Plotting
FPL provides a collection of routines to plot data on unstructured triangular and tetrahedral meshes.

8.2 Tutorial on Working with the *Demonstrator Platform* with Emphasis on MOR

Principle and methods of model order reduction (MOR) are explained in Chaps. 4–6. Here we would like to demonstrate how the *Demonstrator Platform* can be used for testing new MOR algorithms and for coupling reduced order models with the surrounding/driving circuitry. Section 8.2.1 lists the implemented MOR modules and describes in more details those, which are relevant for this tutorial. Section 8.2.2 introduces the chosen nanoelectronic case studies. Section 8.2.3 demonstrates in a step-by-step-manner how to run model order reduction within the *Demonstrator Platform* by using two external libraries MOR for ANSYS [44] and SLICOT [49]. It also describes a provided framework for prototyping and testing new model reduction algorithms. Section 8.2.4 demonstrates how to use reduced order models (created by the mentioned libraries) within a circuit simulation with the OCS.

8.2.1 Overview and Structure of MOR Modules

At present, there are the following MOR related modules within the *Demonstrator Platform*:

- AMOR,
- MOR,
- MOR4ANSYS,
- MOR_UTILITIES,
- ROMI,
- ROM_WB,
- SLICOTINT,

each located in the subdirectory with the same name. For example, the module AMOR is located in *DP_DIR*/`AMOR`, where *DP_DIR* denotes the directory in which the demonstrator platform has been installed. In the following, we will shortly describe the structure and functionality of AMOR, MOR4ANSYS, MOR_Utilities and SLICOTINT modules, which are used in the tutorials in Sect. 8.2.3.

8.2.1.1 Interface Modules AMOR and SLICOTINT

AMOR and SLICOTINT are interfaces between the *Demonstrator Platform* and two external libraries, MOR for ANSYS (described below) and SLICOT (stands for **S**ubroutine **L**ibrary **i**n **Co**ntrol **T**heory, [49]). Both libraries depend on subroutines from BLAS (**B**asic **L**inear **A**lgebra **S**ubroutines) and LAPACK (**L**inear **A**lgebra **Pack**age) [2] for numerical linear algebra and are interfaced to the *Demonstrator Platform* using dynamically linked C++ functions, contained in `amor.cc` and `slicot.cc`, as schematically represented in Fig. 8.2. The compilation of those two functions with `mkoctfile` results in `amor.oct` and `slicot.oct` which provide the Octave functions `amor` and `slicot`. The Fortran subroutines `AB09AD`, `AB09BD` and `AB09CD` from SLICOT implement three different MOR algorithms for first order linear dynamical systems, namely Balanced Truncation Approximation (BTA) [52], Singular Perturbation Approximation (SPA) [36], and Hankel-Norm Approximation (HNA) [27], respectively. Note that the SLICOT library also contains other mathematical tools such as discrete sine/cosine and Fourier transformations, which are however not available within the *Demonstrator Platform* at present. A full list of SLICOT subroutines, the documentation and examples are available at [49]. Input arguments for all functions displayed in Fig. 8.2 will be discussed in Sect. 8.2.3.

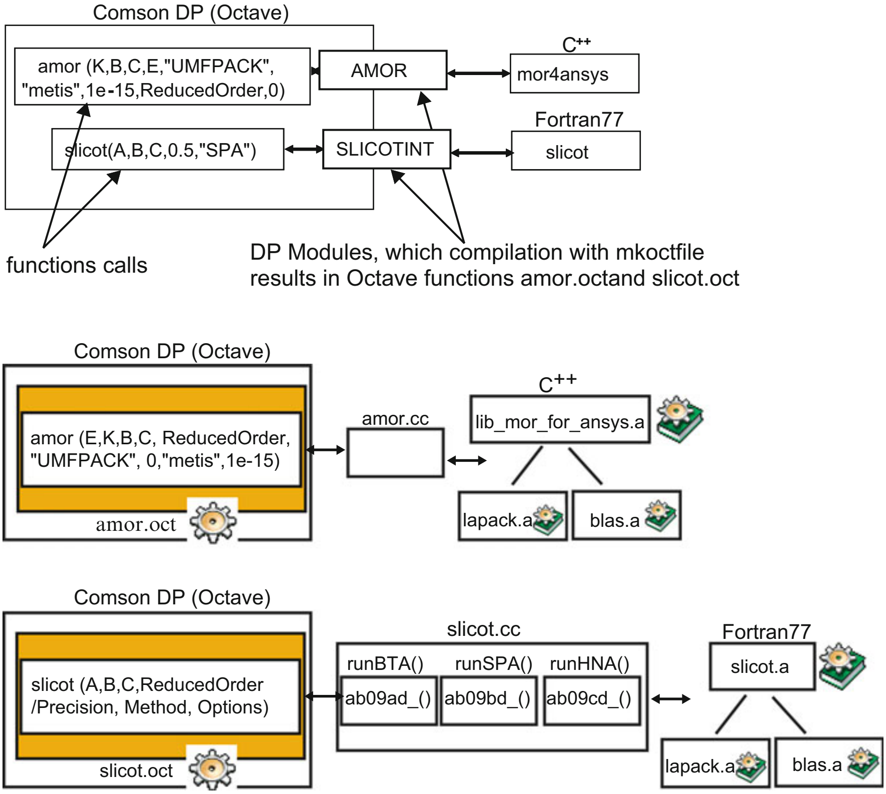


Fig. 8.2 *Demonstrator Platform* interfaces to MOR for ANSYS and SLICOT libraries

8.2.1.2 MOR4ANSYS

The *Demonstrator Platform* module MOR4ANSYS incorporates the C++ library MOR for ANSYS (stands for **mo**del **o**rder **r**eduction **for ANSYS**) [45]. In its original form, MOR for ANSYS is meant to build compact models directly from finite element models implemented in the ANSYS simulator.[1] It implements the first and second order Arnoldi algorithm [5, 26] for model reduction of the linear dynamical systems. In the current implementation, as a *Demonstrator Platform* module, ANSYS support has been disabled and only the reduction of the first order linear dynamical systems is possible (see Fig. 8.3), provided the system matrices are available in the MatrixMarket format [10].

[1] The last GPL licensed version, which has been adapted for the *Demonstrator Platform* is MOR for ANSYS 1.8. Current commercial version is MOR for ANSYS 2.5 (see http://modelreduction.com).

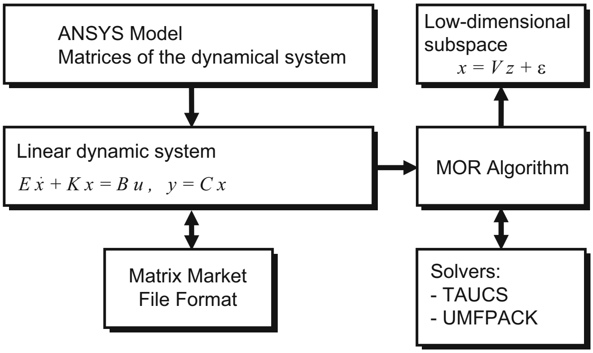


Fig. 8.3 MOR for ANSYS structure. ANSYS support has been disabled within the COMSON *Demonstrator Platform* and only system matrices of the first order linear dynamical system, in the Matrix Market format, can be passed

8.2.1.3 MOR_UTILITIES

The *Demonstrator Platform* module MOR_UTILITIES contains two Octave scripts, namely `MOR4ANSYS_Tutorial.m` and `SLICOT_Tutorial.m`, which demonstrate in a step-by-step manner how to perform model order reduction by interfacing the two libraries MOR for ANSYS and SLICOT. It further contains a script `Post4MOR.m` with a number of Octave functions for postprocessing a reduced order model, i.e. analyzing it in time and frequency domain, visualizing results, etc. The goal of this environment is to provide a framework for prototyping new model reduction algorithms. Our experience shows that it is more convenient to first perform research and prototyping in an interpreter environment, like Octave or Matlab, and only then to perform a compiled language implementation in e.g. C++.

8.2.2 MOR Case Studies

In this section we introduce two case studies to demonstrate the usage of linear model order reduction routines within the *Demonstrator Platform*. The first one is a model of a micro-machined silicon nitride membrane which is, mathematically speaking, a large-scale linear ordinary differential equation system (ODE). The second one is an academic model of a transmission line which is, mathematically speaking, a large-scale linear differential algebraic equation system (DAE).

8.2.2.1 Micro-hotplate

The first test model is a model of a MEMS (micro-electro-mechanical system) device, known as a micro-hotplate. This class of structures is employed in a variety of other microfabricated devices such as gas sensors [31] and infrared sources [50]. The device features sub-millimeter dimensions and is fabricated using technology originating from the semiconductor industry. Silicon is used as a substrate material

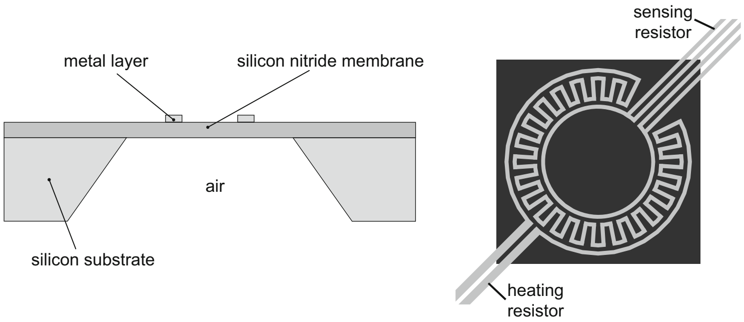


Fig. 8.4 A silicon-nitride membrane with integrated heater and sensing elements was fabricated by low-frequency plasma enhanced chemical vapor deposition. The square membrane is 500 nm thick with a side length of 550 μm, and the metal layer is made from 150 nm platinum with a 50 nm titanium adhesion layer

with a thickness of 525 μm. Silicon nitride with a thickness of 500 nm is deposited onto this substrate. A metal layer, consisting of 150 nm platinum with a 50 nm titanium adhesion layer is fabricated on top of the silicon nitride and shaped to form resistor structures for heating and temperature sensing. In order to achieve suitably high temperatures, the silicon substrate right below the resistor structures is removed by means of wet chemical etching. In this way, a tiny square membrane with 550 μm side length has been created (see Fig. 8.4 left). Such membrane configuration is favorable, as it increases the thermal isolation between the heat source (the heating resistor) and the heat sink (remaining silicon).

Different applications rely on the same working principle, of Joule heating a thin-film membrane. The membrane is being heated by applying an electrical voltage signal to the heating resistor. Temperature measurement on the membrane (necessary for the control) is done with a second resistor, called sensor. Both thin-film metal resistors are arranged as concentric circles (see Fig. 8.4 right) in order to achieve a homogeneous temperature distribution over the membrane.

The target temperature of the membrane is defined by the specific application. In thermo-optical application [33], the membrane temperature determines which wavelength of the focused light will be transmitted. The device works in the temperature range between room temperature and 400 °C. At gas sensing applications [54], the temperature determines the chemical reaction between the unknown gas and the membrane material. The working temperature is between 200 and 400 °C.

In all applications, the simulation goal is to consider important thermal issues such as, which electrical power should be applied in order to reach the target temperature at the membrane or to ensure the homogeneous temperature distribution over the membrane. The original model is the heat-transfer partial differential equation:

$$\nabla \bullet (\kappa(\mathbf{r})\nabla T(\mathbf{r},t)) + Q(\mathbf{r},t) - \rho(\mathbf{r})C_p(\mathbf{r})\frac{\partial T(\mathbf{r},t)}{\partial t} = 0 \tag{8.1}$$

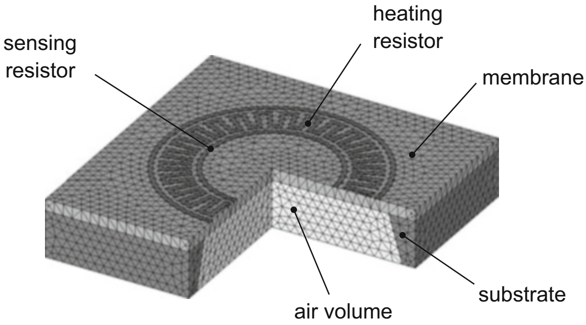


Fig. 8.5 FE mesh of three-dimensional model with 60020 nodes

where **r** is the position, t is the time, κ is the thermal conductivity of the material, C_p is the specific heat capacity, ρ is the mass density, Q is the heat generation rate that is different from zero only within the heater volume ($Q = \mathbf{j}^2 R(T)$, where **j** is the current density within the heater), and T is the unknown temperature distribution. Assuming that the heat distribution is uniformly distributed within the heater and that both material properties κ and C_p are temperature independent around the working point (such assumption can be made in some applications, see e.g. [6]), the finite element (FE) based spatial discretization of (8.1) leads to a large linear ODE system of the form:

$$E \cdot \dot{\mathbf{T}} + K \cdot \mathbf{T} = B \cdot \frac{U^2(t)}{R(T)} \tag{8.2}$$

where t is the time, $\mathbf{T}(t) \in \mathbf{R}^n$ is the state vector of unknown temperatures. E, $K \in \mathbf{R}^{n \times n}$ are heat capacity and heat conductivity which are symmetric, sparse and positive definite matrices.[2] $B \in \mathbf{R}^n$ is the input distribution array and n is the dimension of the system. $U(t)$ is the electrical voltage applied to the heating resistor with temperature-dependent resistance $R(T)$. In case that other sources than $U(t)$ (in total m sources) would be applied to the model, B would be a matrix, $B \in \mathbf{R}^{m \times n}$. Figure 8.5 shows the finite element model (FEM) of the three dimensional geometry, which consists of 60020 nodes. In terms of model (8.24), this means that the dimension of the ODE system is 60020. The solution of (8.24) can be done with a transient analysis within ANSYS, which results in temperature values at specified times in all nodes of the finite element mesh. However, in engineering applications, it is often not necessary to determine the complete temperature field. Mostly, temperature curves in a few nodes are of interest, as specified by the following output equation:

$$\mathbf{y} = C \cdot \mathbf{T} \tag{8.3}$$

[2] In engineering applications the heat capacity matrix is usually denoted with C and heat conductivity matrix with K. However, in the following we use the internal notation of MOR for ANSYS tool, which is used for reduction.

Table 8.1 Outputs for the micro-hotplate model

Name	Number	Comment
Sense_min	37577	Sensor node with minimal temperature
Sense_max	37179	Sensor node with maximal temperature

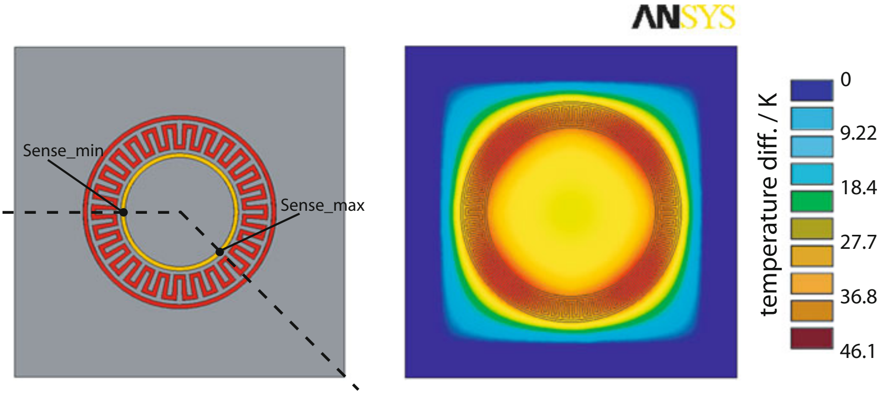


Fig. 8.6 Schematic position of the chosen output nodes (*left*). Temperature distribution after 0.025 s of heating with 2.49 mW (*right*)

where $C \in \mathbf{R}^{p \times n}$ is the user-defined output distribution array and p is the number of outputs of interest. In case that one seeks the complete temperature field, C is the unity matrix of dimension $n \times n$.

For the micro-hotplate, the output nodes of interest (defined by the matrix $C \in \mathbf{R}^{2 \times 60020}$), are described in Table 8.1 and schematically displayed in Fig. 8.6 (left).

The output nodes have been selected in order to capture the average temperature of the sensing resistor. Experimentally, this temperature is obtained by measuring the resistance value of the sensor. Although local variations in the temperature result in local changes in the material's resistivity, this is not resolved by measuring the total resistance. Hence, in order to extract an equivalent temperature value from the FEM results we observe minimum and maximum temperature values in the sensor resistor. The arithmetic mean of these two temperatures is used as an equivalent to the measured temperature.

Equation (8.24) is the starting point for model order reduction, leading to a system of the same form but with smaller dimension. In Sect. 8.2.3 it is demonstrated how to apply MOR to this case study within the framework of the COSMON *Demonstrator Platform*.

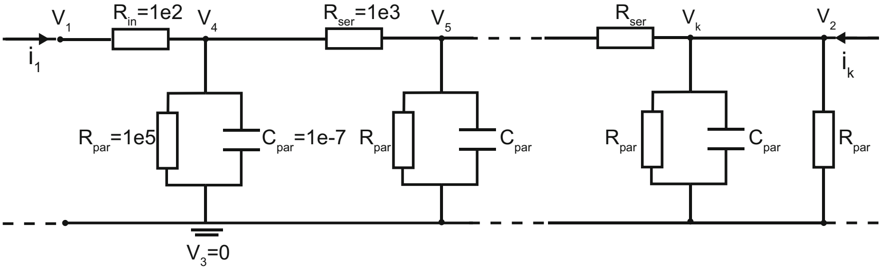


Fig. 8.7 Structure of a transmission line case study. The node numbering is according to the Intermediate File Format (IFF) of the *Demonstrator Platform*, in which external nodes must be numbered first

8.2.2.2 Transmission Line: An Academic Example

Figure 8.7 shows an academic model of a transmission line. This very simple model has been chosen, because it resembles the interconnect modeling and can also be effectively used for testing new MOR algorithms applicable to DAE systems. It consists of a scalable number of RC ladders and after performing a charge-oriented Modified Nodal Analysis (MNA) a linear DAE system of the form:

$$E \cdot \dot{\mathbf{x}} + K \cdot \mathbf{x} = B \cdot \mathbf{u}(t) \tag{8.4}$$

is obtained. In (8.4), t denotes the time, $\mathbf{x}(t) \in \mathbf{R}^n$ is the state vector containing in case of MNA nodal voltages and branch currents, i.e. in the *Demonstrator Platform* implementation also the charges of the capacitors, n is the dimension of the system and $\mathbf{u}(t) \in \mathbf{R}^m$ is the input excitation vector. $E, K \in \mathbf{R}^{n \times n}$ are constant and sparse system matrices,[3] which represent the contribution of capacitors and resistors, respectively. For the transmission line model E is singular as it contains only zero-valued entries in the rows corresponding to the resistor branches. $B \in \mathbf{R}^{n \times m}$ is the input distribution array and m is the number of inputs, i.e. sources. It is further possible to define the output measurement vector $\mathbf{y}(t) \in \mathbf{R}^p$, where p is the number of outputs of interest as:

$$\mathbf{y} = C \cdot \mathbf{x} + D \cdot \mathbf{u}(t) \tag{8.5}$$

with $C \in \mathbf{R}^{p \times n}$ and $D \in \mathbf{R}^{p \times m}$. Contrary to the previously described finite element model, in case of transmission line, it is necessary to specify a second term in the output equation, $D \cdot \mathbf{u}(t)$, which allows for the coupling of the transmission line with surrounding circuitry.

[3] In engineering applications the circuit matrices are usually denoted with C and G. However, here we follow the internal notation of MOR for ANSYS tool, which is used for reduction.

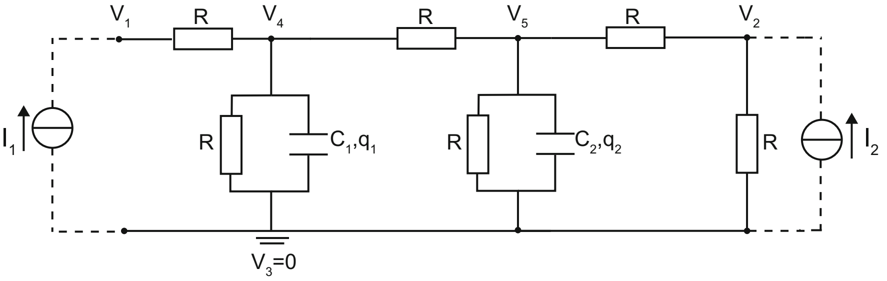


Fig. 8.8 Transmission line with six resistances and two capacitors. Coupling to the surrounding is modeled through two fictive current sources

In order to explain how the system matrices are assembled within the *Demonstrator Platform*, we observe the test model with six resistors and two capacitors, shown in Fig. 8.8, and write down the MNA equations. We model the coupling to the surrounding through two virtual current sources, I_1 and I_2. MNA for nodes N_1, N_2, N_4 and N_5 and for both charges C_1 and C_2, leads to:

$$\text{node } N_1 : \frac{V_1 - V_4}{R} = I_1 \tag{8.6}$$

$$\text{node } N_2 : \frac{V_2 - V_5}{R} + \frac{V_2}{R} = I_2 \tag{8.7}$$

$$\text{node } N_4 : \frac{V_4 - V_1}{R} + \frac{V_4}{R} + \dot{q}_1 + \frac{V_4 - V_5}{R} = 0 \tag{8.8}$$

$$\text{node } N_5 : \frac{V_5 - V_4}{R} + \frac{V_5}{R} + \dot{q}_2 + \frac{V_5 - V_2}{R} = 0 \tag{8.9}$$

$$\text{charge } C_1 : C_1 \cdot V_4 - q_1 = 0 \tag{8.10}$$

$$\text{charge } C_2 : C_2 \cdot V_5 - q_2 = 0 \tag{8.11}$$

or in matrix form:

$$\begin{bmatrix} 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0 \end{bmatrix} \cdot \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_4 \\ \dot{V}_5 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{R} & 0 & -\frac{1}{R} & 0 & 0 & 0 \\ 0 & \frac{2}{R} & 0 & -\frac{1}{R} & 0 & 0 \\ -\frac{1}{R} & 0 & \frac{3}{R} & -\frac{1}{R} & 0 & 0 \\ 0 & -\frac{1}{R} & -\frac{1}{R} & \frac{3}{R} & 0 & 0 \\ 0 & 0 & C_1 & 0 & -1 & 0 \\ 0 & 0 & 0 & C_2 & 0 & -1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_4 \\ V_5 \\ q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{8.12}$$

This resembles (8.4). If we consider (8.12) as a stand-alone system, we can define C in such a way that an arbitrary nodal voltage V_k is considered to be an output of interest. In such case $D = 0$ and the output equation has the form (8.3).

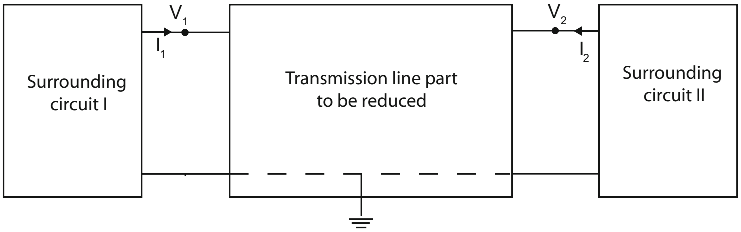


Fig. 8.9 Inputs and outputs of the transmission line, when coupled to the surrounding circuitry

However, if the transmission line is a part of a broader circuit, as shown in Fig. 8.9, it is necessary to introduce terminal-voltages and terminal-currents (V_1, V_2, I_1 and I_2) as inputs/outputs of the model. If we define e.g. terminal voltages as inputs, $u(t) = [V_1\ V_2]^T$, terminal currents as outputs, $y = [I_1\ I_2]^T$ and $x = [V_4\ V_5\ q_1\ q_2]^T$ as the new state-vector, (8.12) can be interpreted as:

$$\left.\begin{array}{r} E \cdot \dot{\mathbf{x}} + K \cdot \mathbf{x} = B \cdot \mathbf{u}(t) \\ \mathbf{y} = C \cdot \mathbf{x} + D \cdot \mathbf{u}(t) \end{array}\right\} \quad \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \cdot \begin{bmatrix} \dot{u} \\ \dot{x} \end{bmatrix} + \begin{bmatrix} D & C \\ -B & K \end{bmatrix} \cdot \begin{bmatrix} u \\ x \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{8.13}$$

with matrices E, K, B, C and D defined as schematically displayed in (8.14).

$$\left[\begin{array}{cc|cccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \underbrace{0 \quad 0 \quad 0 \quad 0}_{E} \end{array}\right] \cdot \overset{\dot{\mathbf{u}}}{\underset{\dot{\mathbf{x}}}{\begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{V}_4 \\ \dot{V}_5 \\ \dot{q}_1 \\ \dot{q}_2 \end{bmatrix}}} + \left[\begin{array}{cc|cccc} \multicolumn{2}{c|}{D} & \multicolumn{4}{c}{C} \\ \frac{1}{R} & 0 & -\frac{1}{R} & 0 & 0 & 0 \\ 0 & \frac{2}{R} & 0 & -\frac{1}{R} & 0 & 0 \\ \hline -\frac{1}{R} & 0 & \frac{3}{R} & -\frac{1}{R} & 0 & 0 \\ 0 & -\frac{1}{R} & -\frac{1}{R} & \frac{3}{R} & 0 & 0 \\ 0 & 0 & c_1 & 0 & -1 & 0 \\ 0 & 0 & 0 & c_2 & 0 & -1 \\ \multicolumn{2}{c|}{\text{-}B} & \multicolumn{4}{c}{K} \end{array}\right] \cdot \overset{\mathbf{u}}{\underset{\mathbf{x}}{\begin{bmatrix} V_1 \\ V_2 \\ V_4 \\ V_5 \\ q_1 \\ q_2 \end{bmatrix}}} = \overset{\mathbf{y}}{\begin{bmatrix} I_1 \\ I_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}} \tag{8.14}$$

System (8.13) can be subjected to model order reduction, which leads to a system of the same form, but with reduced system matrices, as follows:

$$\begin{bmatrix} 0 & 0 \\ 0 & E_r \end{bmatrix} \cdot \begin{bmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{z}} \end{bmatrix} + \begin{bmatrix} D & C_r \\ -B_r & K_r \end{bmatrix} \cdot \begin{bmatrix} \mathbf{u} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{y_r} \\ 0 \end{bmatrix} \tag{8.15}$$

(8.15) can be coupled to the surrounding circuitry. Note that the dimension of D is the same before and after reduction (2×2 in case of transmission line), because the inputs/outputs must stay preserved. The reduction of the transmission line case study is demonstrated in Sect. 8.2.2.2.

8.2.3 A Step by Step Model Reduction Tutorial

In the following we demonstrate model order reduction of the micro-hotplate model in a step-by-step tutorial. We use the following Octave scripts from the module MOR_Utilities:

- `MOR4ANSYS_Tutorial.m`
- `SLICOT_Tutorial.m`
- `Post4MOR.m`

8.2.3.1 Preparing the Model

It is the responsibility of the user to supply the system in the form:

$$\begin{aligned} E \cdot \dot{\mathbf{x}} &= A \cdot \mathbf{x} + B \cdot \mathbf{u} \\ \mathbf{y} &= C \cdot \mathbf{x} \end{aligned} \tag{8.16}$$

where the system matrices E, A, B and C are written in the Matrix Market Format [10]. The naming convection is *ModelName*`.E`, *ModelName*`.A`, *ModelName*`.B` and *ModelName*`.C` where *ModelName* is a user-defined string. It is further convenient to prepare a text file, which contains names of the output nodes, as listed in Table 8.1. These names are used by the postprocessing functions in the `Post4MOR.m` script to e.g. label the plots for the specific outputs of the full-scale and reduced order models. The microhotplate model (Sect. 8.2.2.1) is defined through the following files, all located within the module MOR_Utilities:

- `MicroHotplate.E`,
- `MicroHotplate.A`,
- `MicroHotplate.B`,
- `MicroHotplate.C`,
- `MicroHotplate.C.names`,

where the first four are are written in Matrix Market Format and the last one is a text file in which each line contains the name of the output node, as:

```
Sense_min
Sense_max
```

Prior to actual reduction, it is necessary to load the functions from `Post4MOR.m` and the test model into the Octave environment, as follows:

```
Post4MOR;
[E,A,B,C] = ReadInTestModel("MicroHotplate");
```

`Post4MOR.m` contains Octave functions for integrating the model (8.24), computing the transfer functions of the full and reduced models, computing and plotting different reduction errors in either time or frequency domain, etc. The function `ReadInTestModel` loads the system matrices of the micro-hotplate model from the above files into Octave.

8.2.3.2 Model Reduction with MOR for ANSYS via AMOR

After all four matrices from (8.16) are available in Matrix Market Format and the text file with outputs names is prepared, we can proceed to model order reduction of the dynamical system (8.16). For reduction of the large-scale systems (a few thousand degrees of freedom) within the *Demonstrator Platform*, we propose to use the C++ library, MOR for ANSYS. From the *Demonstrator Platform* it can be called via its Octave interface AMOR.

The implemented Arnoldi reduction algorithm [26] can be applied for ODE and DAE systems equally, i.e. regardless if E in (8.16) is regular or singular. The passivity and stability of the reduced model are granted, as long as both system matrices E and A are positive and semi-definite, as shown in [48]. The basic idea of model reduction with the Arnoldi algorithm is to find a low-dimensional subspace that approximates the transient behavior of the state vector $\mathbf{x}$:

$$\mathbf{x} = V \cdot \mathbf{z} + \epsilon \tag{8.17}$$

and the approximation error ϵ is assumed to be small even though the number of columns of projection matrix V (i.e. the dimension of $\mathbf{z}$) is much less than the number of rows (i.e. the dimension of $\mathbf{x}$). The compact model of the linear first order dynamical system is obtained by the projection of (8.16) as follows:

$$\begin{aligned} V^T E V \cdot \mathbf{z} &= V^T A V \cdot \mathbf{z} + V^T B \cdot \mathbf{u} \\ \mathbf{y_r} &= C V \cdot \mathbf{z} \end{aligned} \tag{8.18}$$

The projection matrix V is iteratively obtained by the MOR algorithm, i.e. column by column. This means that if we produced the reduced model of order r, we can obtain all reduced models of any lower order just by discarding the last columns in the project matrix. This can be used in the recommended strategy [7] to find an optimum dimension of the reduced model, by observing the convergence of the relative error between two "neighbored" reduced order models, with orders r and $r + 1$. The reduced order model is valid for an arbitrary input function. Furthermore, the transient and harmonic simulation of (8.18) is much faster than those of the original high-order system (8.16). However, the physical meaning of the original state vector x is lost in (8.18). The new state vector z can be considered as a vector of generalized coordinates, which requires a certain level of abstraction in the engineering applications. Even so, the inputs and outputs defined by the matrices B and C, will stay preserved after the reduction, i.e. y_r from (8.18) approximates y from (8.16) with high accuracy. It is also possible to recover the complete original state vector x by back-projecting z, as indicated in (8.17), while neglecting ϵ. The functionality of the Arnoldi algorithm is that the transfer function of the full-scale model, defined as:

$$H(s) = C(sE - A)^{-1}B \tag{8.19}$$

when developed into a Taylor series around some value of the Laplace variable $s = s_0$:

$$H(s) = \sum_{i=0}^{\infty} m_i(s_0)(s - s_0)^i \tag{8.20}$$

where $m_i(s_0) = C(-(s_0E - A)^{-1}E)^i \cdot (s_0E - A)^{-1}B$ is called the i-th moment around s_0 will have the same moments as the transfer function $H_r(s)$ of the reduced model, up to the degree r. With other words, it approximates the input/output behavior.

The AMOR interface between MOR for ANSYS and the *Demonstrator Platform* should be called with

```
V = amor(E, K, B, C, r , solver , s_0, \
         ReorderingScheme , tol );
```

where, E, $K = -A$, B and C are the system matrices[4] of the dynamic system, as defined in (8.16) and other parameters (which are optional) are meant to control the model reduction process, as follows:

- `r` specifies the dimension of the reduced model. By default it is 30.
- `solver` is to choose a solver. By default it is `TAUCSlltmf` (suitable for positive definite matrices).
- `s_0` specifies which expansion point should be used for the transfer function. By default it is 0.
- `ReorderingScheme` is the solver parameter. By default it is `metis`.
- `tol` sets up the tolerance to deflate the next column vector of V. By default it is 10^{-15}.

The default values have been chosen based on experience with electro-thermal models of MEMS devices [8]. The output of AMOR is a projection matrix V from (8.17), which is constructed vector by vector. When a new vector is generated, MOR for ANSYS checks its norm. If it is less then the tolerance specified with `tol` option,

[4] Note that the internal system representation from of MOR for ANSYS V. 1.8 differs from (8.16), as it uses matrix $K = -A$.

the vector is deflated (removed), as it is assumed to represent a zero vector within rounding errors.

There is a vast number of solvers to solve a system of linear algebraic equations. MOR for ANSYS 1.8 uses the TAUCS [51] and UMFPACK [19] libraries with following solver choices:

- `TAUCSlltmf`: Multifrontal supernodal Cholesky decomposition.
- `TAUCSlltll`: Left-looking supernodal Cholesky decomposition.
- `TAUCSlltooc`: Out-of-core sparse Cholesky decomposition.
- `TAUCSllt`: Cholesky decomposition column by column (slow).
- `TAUCSldlt`: LDLT factorization.
- `TAUCSlu`: Out-of-core sparse pivoting LU decomposition.
- `UMFPACK`: Multifrontal LU decomposition.

TAUCS solvers for symmetric matrices can take a reordering-scheme parameter, which specifies the reordering method. Allowable values for `ReorderingScheme` parameter are:

- `metis`: hybrid nested-dissection minimum degree ordering.
- `genmmd`: multiple minimum degree ordering.
- `md`: minimum degree ordering.
- `mmd`: multiple minimum degree ordering.
- `amd`: approximate minimum degree ordering.
- `treeorder`: no-fill ordering code for matrices whose graphs are trees.

Our recommendations are as follows. For symmetric and positive definite matrices `TAUCSlltmf` with `metis` is the best choice. If the matrix is symmetric but indefinite `TAUCSldlt` with `metis` is a good choice, although `UMFPACK` may be faster in this case. For non-symmetric matrices one must use `UMFPACK`.

By default, MOR for ANSYS uses zero as an expansion point of the transfer function. This means that the reduced order model will approximate the original model accurately at low frequencies. If, however, the expansion point is different from zero, the reduced model will not preserve the stationary state. The choice of the expansion point depends on the application. For more information about methods and more details on input parameters, please consult the MOR for ANSYS 1.8 manual within the *Demonstrator Platform* documentation.

We run the model order reduction of the micro-hotplate model as follows:

```
K = -A;
V = amor(E, K, B,C, 30, "UMFPACK", 0, "metis", 1e-15);
```

As the output of the call to `amor` is a projection matrix, it is necessary to actually construct the reduced order model (8.18). This can be done by projection:

```
Er = V' * E * V;
Kr = V' * K * V;
Br = V' * B;
Cr = C * V;
```

or in the more compact form, by calling the `build_reduced_system` routine from `Post4MOR.m`:

```
[Er,Kr,Br,Cr] = build_reduced_system(E, K, B, C, V);
Ar = -Kr;
```

It is further possible to save the reduced system in the Matrix Market form by setting a base name for the reduced model. The base name can be same or different than for the original model. In either case, an extention `.MOR` will be attached. For example:

```
write_reduced_system("MicroHotplate", Er, Ar, Br, Cr);
```

will produce the following files:

- `MicroHotplate.MOR.E`,
- `MicroHotplate.MOR.A`,
- `MicroHotplate.MOR.B`,
- `MicroHotplate.MOR.C`.

8.2.3.3 Postprocessing of the Reduced Model and Error Estimation

Once the reduced order model has been created, it is necessary to integrate it, compare it with the full-scale model or with measurements, plot it in time and/or frequency domain etc. For the micro-hotplate we define the integration time of 0.04 s and the constant input function $u(t) = 1$, which corresponds to the constant input power of $Q = 2.49$ mW. Note, that it is also possible to introduce the non-linearity of the input function, as indicated in (8.24) directly on the level of the reduced model, by defining u to be a function of the reduced state-vector z. The following code sequence performs the time integration of the full-scale and reduced order models:

Define the input function and the time range for the time integration:

```
global u = 1;
t = linspace(0, 0.04, 100);
```

Now we need the initial values of the full and reduced state vector:

```
x(:,1) = zeros(rows(K), 1);
z(:,1) = zeros(rows(Kr), 1);
```

The time integration of the full-scale model might take quite some time:

```
y = TimeIntegration(E, K, B, C, u, x, t);
```

The time integration of the reduced model should be fast:

```
yr = TimeIntegration(Er, Kr, Br, Cr, u, z, t);
```

It is advisable to save solutions for further use:

```
save("FullSolution", "y");
save("ReducedSolution", "yr");
```

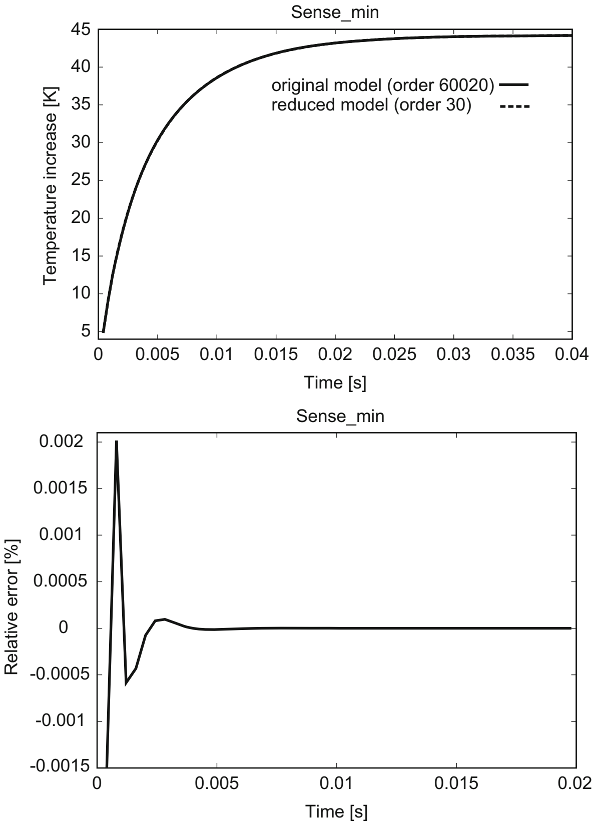


Fig. 8.10 Time response of the full model (order 60000) and reduced model (order 30) (*top*) and relative error between the both (*bottom*)

One can further plot all outputs of the full and reduced system in time domain with:

```
PlotAllOutputs(y, yr, t, "MicroHotplate.C.names");
```

and plot the relative and absolute errors between the full and the reduced model in each node:

```
PlotRelativeErrorTimeDomain(y, yr, t, \
                            "MicroHotplate.C.names");
PlotAbsoluteErrorTimeDomain(y, yr, t, \
                            "MicroHotplate.C.names");
```

Figure 8.10 shows the temperature response in time and the relative error between the full-scale and the reduced order model in `Sense_min` node.

It is further possible to compare the dynamics of the reduced and original system in frequency domain as well. The following functions compute and plot the transfer

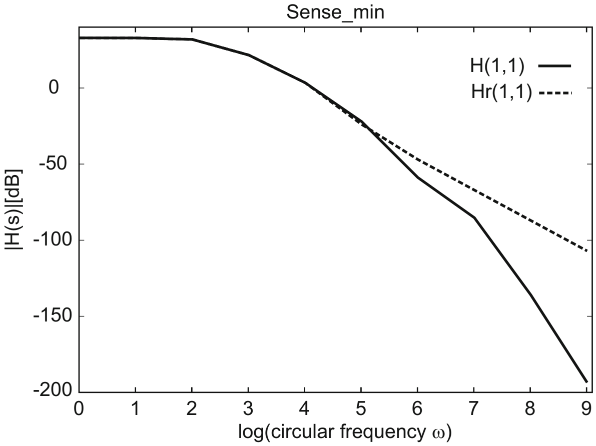


Fig. 8.11 Frequency response H(1,1) of the full model (order 60000) and frequency response Hr(1,1) of the reduced model (order 30)

functions of the full and reduced models for 10 frequency values in the range from 1 to 10^9 rad/s:

Specify the number of frequencies (the frequency range is from 10^0 rad/s to $10^{\mathrm{NrOfFreq}-1}$ rad/s):

```
NrOfFreq = 10;
```

Now compute the transfer function of the full-scale system:

```
G = ComputeTransferFunction(A, B, C, E, NrOfFreq);
```

Compute the transfer function of the reduced system:

```
Gr = ComputeTransferFunction(Ar, Br, Cr, Er, NrOfFreq);
```

Plot the transfer functions between each input/output:

```
PlotAllTransferFunctions(G, Gr, NrOfFreq, \
                         "MicroHotplate.C.names");
```

Figure 8.11 shows both transfer functions in output node Sense_min. The transfer functions match well for low frequencies, which is due to the fact that we have chosen zero as an expansion point for the Taylor series in (8.20).

The main draw-back of the Arnoldi algorithm is that there is no mathematical theory to estimate the reduction error. Hence, for the user it is difficult to predict which dimension of the reduced model will provide the required accuracy. In [7] we have shown that it is possible to estimate the reduction error by observing the relative error in frequency domain between two "neighbored" reduced models of order r and $r + 1$. If we define a relative frequency response error as:

$$E_r(s) = \frac{|H(s) - H_r(s)|}{|H(s)|} \tag{8.21}$$

where $H(s)$ and $H_r(s)$ are the transfer functions of the original and of the reduced order model (as defined in (8.19)), respectively, and a relative frequency-response error between two successive reduced order models as:

$$\hat{E}_r(s) = \frac{|H_r(s) - H_{r+1}(s)|}{|H_r(s)|} \tag{8.22}$$

it turns out that for the micro-hotplate case studies it holds:

$$E_r(s) \approx \hat{E}_r(s) \tag{8.23}$$

For benchmarks from [35] (8.23) holds for a wide range of frequencies around the expansion point $s_0 = 0$. To observe the convergence of the relative error within the *Demonstrator Platform* run:

```
ErrorEstimate(E, A, B, C, V, Freq);
```

where `Freq` is the circular frequency of interest. Figure 8.12 shows the convergence of relative error at 10^3 rad/s and at 10^6 rad/s. We observe that E_r and $\hat{E}_r$ match well at lower frequency 10^3 rad/s, which is near the expansion point $s_0 = 0$. The system order necessary to reach the convergence at $\omega = 10^3$ rad/s is 16, which means that it is not possible to approximate the system better with higher order. The convergence occurs presumably because the machine's numerical precision has been reached. At high frequencies convergence disappears. Instead, we observe fluctuations, due to being too far away from the expansion point.

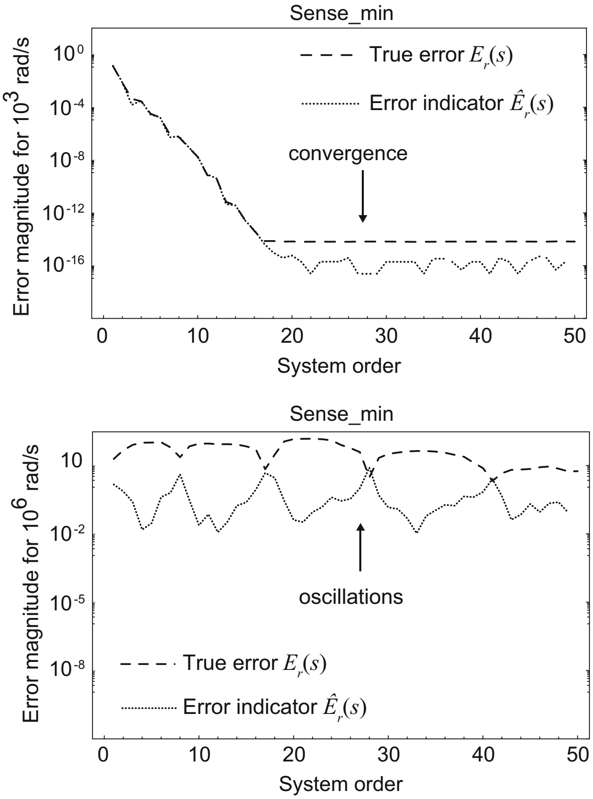
8.2.3.4 Running SLICOT via SLICOTINT for Further Reduction of the Compact Model

The SLICOT subroutines can be used for reduction of moderate size models (order few thousands) and for ODE systems of the form:

$$\begin{aligned} \dot{\mathbf{x}} &= A \cdot \mathbf{x} + B \cdot \mathbf{u} \\ \mathbf{y} &= C \cdot \mathbf{x} \end{aligned} \tag{8.24}$$

only. It is possible to use them to further compact the model that has been obtained by MOR for ANSYS. We have developed the Octave interface SLICOTINT to the model reduction subroutines of the SLICOT library. The provided script `SLICOT_Tutorial.m` gives the guidelines on how to use it and demonstrates how the reduction is performed with the micro-hotplate example.

Fig. 8.12 Error indicators in the frequency domain in output node Sense_min of the micro-hotplate model at $\omega = 10^3$ rad/s (*top*) and at $\omega = 10^6$ rad/s (*bottom*)



We invoke Octave, load the reduced-order model (it is of order 30) into the Octave environment and convert it to the state-space (single-matrix) form, which is necessary for MOR methods implemented in SLICOT:

Load the functions for the MOR postprocessing and the test model:

```
Post4MOR;
[E1,A1,B1,C1] = ReadInTestModel("MicroHotplate.MOR");
```

Convert the model into the state-space (single matrix) form, which is necessary for control-theory routines implemented in SLICOT:

```
A = E1\A1;
B = E1\B1;
C = C1;
```

We call the Balanced Truncation Approximation algorithm, implemented in SLICOT with:

```
[Ar,Br,Cr,HSV] = slicot(A, B, C, 5, "BTA");
```

or with:

```
[Ar,Br,Cr,HSV] = slicot(A, B, C, 0.01, "BTA");
```

The first call invokes reduction with the BTA method to order five (resulting ODE system will have five equations) and the second one, a reduction to the smaller order system with the accuracy of 1 %. The latest is possible, because the MOR methods implemented in SLICOT (also known as control theory methods), provide a global error bound between the transfer function $H(s)$ of the original and $H_r(s)$ of the reduced system as follows:

$$\|H(s) - H_r(s)\|_\infty \leq 2(\sigma_{r+1} + \ldots + \sigma_n) \tag{8.25}$$

where the infinity norm $\|.\|_\infty$ denotes the largest magnitude of the difference of the transfer functions and $\sigma_{r+1}, \ldots, \sigma_n$ are the smallest Hankel singular values (HSV) of the dynamical system under consideration. Hankel singular values are the property of the dynamical system, which reflect the contributions of the different entries of the state vector to system responses (see [3] for theoretical explanation).

For calling one of the other two MOR algorithms for linear ODE systems implemented in SLICOT, that is Hankel Norm Approximation or Singular Perturbation Approximation, it is necessary to replace `BTA` with `HNA` or `SPA`. Each function returns the system matrices of the reduced ODE system, A_r, B_r and C_r, as well as the list of Hankel singular values sorted in descending order.

After reduction, it is possible to display the responses of the full and the reduced order systems in time and/or frequency domain and to plot the relative and absolute errors, by using the same commands as described in Sect. 8.2.3.3. Figure 8.13 shows the responses of two different reduced models produced with the BTA algorithm for the same input function, initial conditions and time- and frequency ranges, as in the previous step.

It is further interesting to observe (see Fig. 8.14), how the target order five model can be reached with smaller error, in transient phase, if sequential MOR is used (reduction of the original model with order 60020 down to order 30 with Arnoldi algorithm and further to order five with BTA) than the Arnoldi algorithm alone. This is due to the fact that the resulting reduced model of order five (gained by sequential MOR) includes information of 30 Arnoldi vectors. The steady-state error increases however, which is typical for the BTA. Better approximation of the steady-state can be reached by using SPA (see [18] for more explanation). With

```
plot(log(HSV),".");
```

we can plot the Hankel singular values of the system. The following commands compute and plot the error bound as defined by (8.25). First, we compute the frequency domain error of reduction (infinity norm):

```
Eps_jw = InfinityNormError(G, Gr, NrOfFreq);
```

Then, we plot the frequency domain error of reduction (infinity norm):

```
PlotInfinityNormError(Eps_jw, NrOfFreq);
```

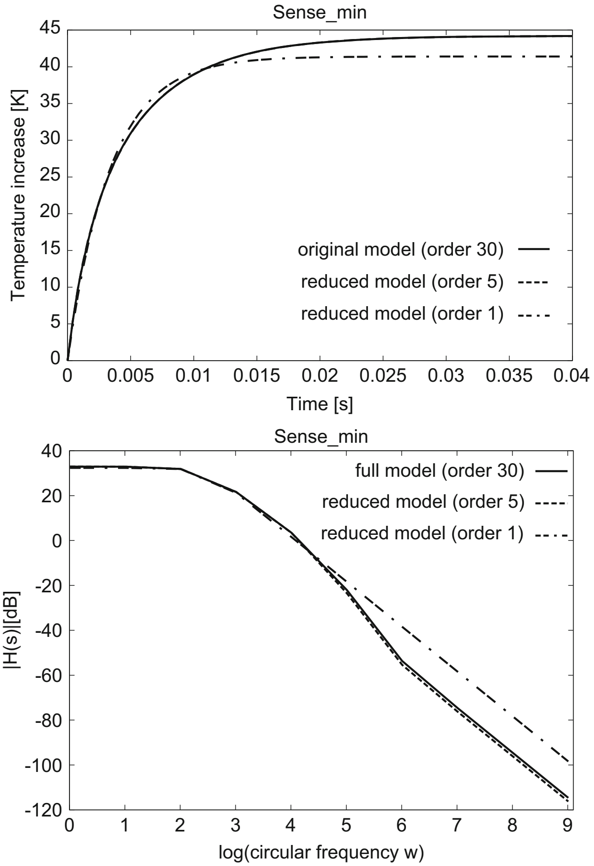


Fig. 8.13 Time response (*top*) and frequency response (*bottom*) in Sense_min output node of the micro-hotplate model of order 30 (gained through the reduction with MOR for ANSYS) and of reduced orders five and one (both gained through further reduction with BTA algorithm from SLICOT)

Figure 8.15 shows the rapid decay of Hankel singular values for the micro-hotplate model of order 30, which indicates further reducibility of the dynamical system. Figure 8.16 shows the difference between the transfer function of the full (order 30) model of the micro-hotplate and of the reduced (order 6) model.

8.2.4 Coupled Simulation (MOR + Circuit Simulation)

The main goal of implementing model order reduction feature into the COMSON *Demonstrator Platform* is to speed up the simulation of complex electronic circuits.

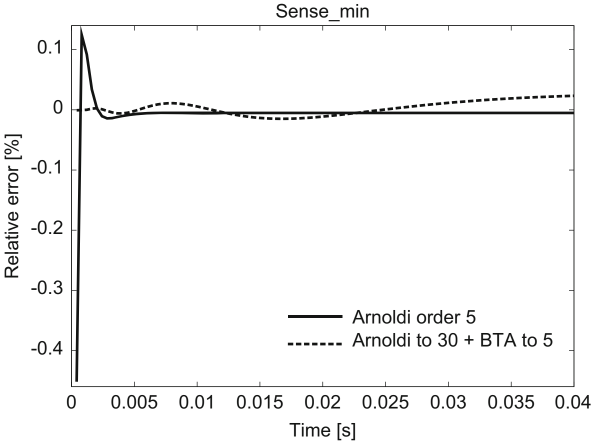


Fig. 8.14 Relative error of the single-step and of the sequential reduction of the micro-hotplate model with Arnoldi algorithm (MOR for ANSYS implementation) and with BTA algorithm (SLICOT implementation)

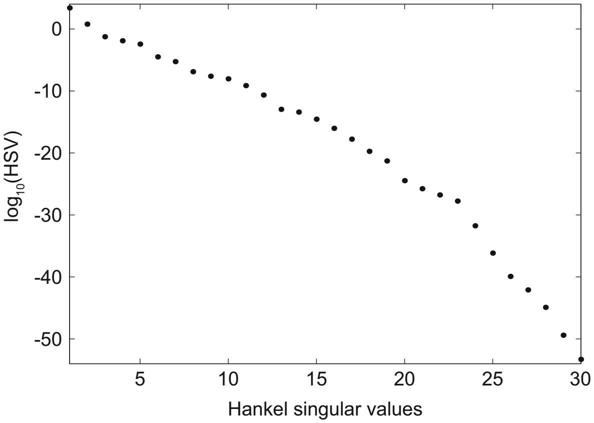


Fig. 8.15 Logarithm of Hankel singular values for the micro-hotplate model of order 30. Original model (order 60020) was reduced with MOR for ANSYS

In this section we demonstrate how MOR can be efficiently applied to build a compact model of interconnects and couple it with the surrounding circuitry.

We simulate the transmission line model from Sect. 8.2.2.2 with 20 resistors and 9 capacitors, which is coupled to the simple non-linear circuit, as shown in Fig. 8.17. The transmission line model is a linear DAE system of form (8.13) which can be reduced within the *Demonstrator Platform* with the MOR for ANSYS library. We would like to emphasize once again, that the entries of the reduced state vector (z in (8.15)) are without physical meaning, i.e. they do not represent internal voltages and charges of the transmission line circuit. However, as during

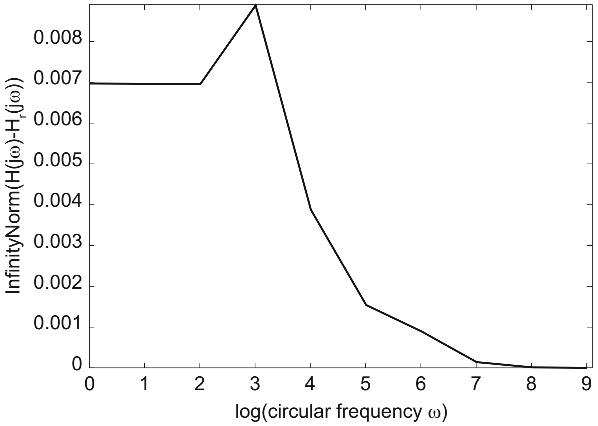


Fig. 8.16 Infinity norm of the difference between the transfer function of the full (order 30) model of the micro-hotplate and of the reduced (order six) model. Reduction was done with SLICOT (BTA) by specifying the error of (8.25) to be 0.01

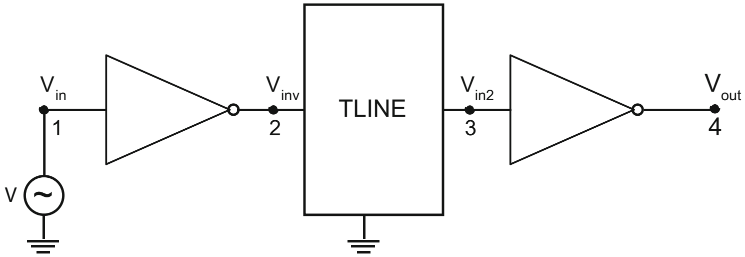


Fig. 8.17 Transmission line model coupled to non-linear circuity

the reduction the coupling states (those entries of the MNA state vector which are common for the transmission line and the surrounding circuit, as V_{inv} and V_{in2} in Fig. 8.17), are preserved, it is possible to couple the reduced order model to both inverters. This coupling is done in the same manner as without reduction, i.e. by stamping the matrices of (8.15) into the global system matrices (system matrices of the whole inverter circuit) at proper positions.

The transmission line model and the inverter circuit are parts of OCS (Octave Circuit Simulator) module. They are described in the following files:

- `MTransLine.cir` (located in the *DP_DIR*/`OCS/SBN`)
- `MTransLine.m` (located in the *DP_DIR*/`OCS/SBN`)
- `TLine2Inv.cir` (located in *DP_DIR*/`OCS/Examples/TLINE`)
- `TLine2Inv.nms` (located in *DP_DIR*/`OCS/Examples/TLINE`)

`MTransLine.cir` describes the circuit from Fig. 8.7 with 9 capacitors and 20 resistors. It has two external variables (dimension of **u** in (8.13)) and 18 internal variables (dimension of **x** in (8.13)).

MTransLine.m builds the local system matrices. It takes as input parameter a string NonReduced or Reduced. If former, the system (8.13) is built. If latter, first the system (8.13) is built and then there is a call to MOR for ANSYS with:

```
ReducedOrder = 3;
V = amor(E, K, B, C, ReducedOrder, "UMFPACK", 0,\
         "metis", 1e-15);
```

reducing the internal states down to 3. Reduced matrices are computed by projecting the original ones as follows:

```
Er = V'*E*V;
Kr = V'*K*V;
Br = V'*B;
Cr = C*V;
```

and the system (8.15) is built.

TLine2Inv.cir describes the circuit from Fig. 8.17. It contains a sinusoidal voltage source with an DC offset of 1.5 V, two inverters and a single transmission line from the template MTransLine.m. The call to transmission line is as follows:

```
% Tranmission line from MTransLine.m
MTransLine NonReduced 2 4
1 4
Rin    Rser   Rpar   Cpar
1e2    1e3    1e5    1e-7
2 3
```

where the string NonReduced can be replaced with Reduced.

TLine2Inv.nms contains the names of the circuit outputs of interest, as follows:

```
%0.1b1
1 Vin
2 Vinv
3 Vin2
4 Vout
```

One can run the simulation of TLine2Inv.cir (either with or without the reduction of the transmission line) by running the Octave script runme.m which is located in the directory *DP_DIR*/OCS/Examples/. It is first necessary to manually set the path to AMOR interface by executing

```
Usetpath(pwd);
```

within the *DP_DIR*/AMOR directory. Calling `runme` within Octave yields:

```
Chose an example to run:

  [ 1] rcs
  [ 2] DIODEEXAMPLE
  [ 3] MOR
  [ 4] nmos
  [ 5] pmos
  [ 6] inverter
  [ 7] and
  [ 8] and2
  [ 9] tl
  [10] rect
  [11] MOSIV
  [12] TLINE
  [13] RLC
  [14] RCSPDE

pick a number, any number:
```

and it is necessary to chose the *TLINE* example (number 12). The transient simulation is performed and the function `UTLplotbyname` from *DP_DIR*/OCS/UTL is used to plot the variables specified in the `TLine2Inv.nms`. Figure 8.18 shows the four node potentials over 0.2 s of transient simulation. If we now change the parameter within the `TLine2Inv.cir` into “Reduced”, we can observe and compare the outputs (nodal voltages) with and without reduction. It is possible to use the MOR post-processing functions from the *MOR_Utilities* module, which were

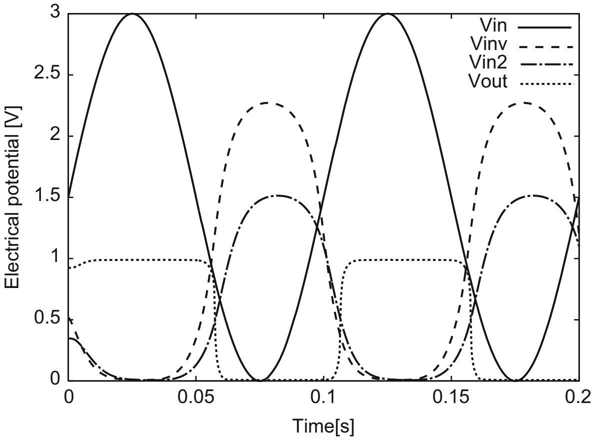


Fig. 8.18 Outputs of the circuit from Fig. 8.17. Transient simulation was done with the OCS module of the COMSON *Demonstrator Platform*

described in the previous section. We observe the nodal voltage V_{out}. The following code saves the solution with and without reduction of the transmission line:

```
if (strcmp(exmpl, "TLINE"))
  if (strcmp(outstruct.LCR(2).section, "NonReduced"))
    FullOutputVoltage = out(:,4);
    save("FullOutputVoltage", FullOutputVoltage);
  elseif(strcmp(outstruct.LCR(2).section, "Reduced"))
    ReducedOutputVoltage = out(:,4);
    save("ReducedOutputVoltage", ReducedOutputVoltage);
  endif
endif
```

We plot full and reduced output and the relative error between the both with:

```
PlotAllOutputs(FullOutputVoltage, \
    ReducedOutputVoltage, t, "TransLineOutputs.names");

PlotRelativeErrorTimeDomain(FullOutputVoltage, \
    ReducedOutputVoltage, t, "TransLineOutputs.names");
```

where in text file `TransLineOutputs.names` we just write the name `V_out`. The resulting plots are displayed in Fig. 8.19. As expected, the change in the output voltage is negligible. As the reduction was performed with MOR for ANSYS, which implements Arnoldi algorithm, and as we have chosen 0 Hz as an expansion point, the steady-state phases are better approximated than the transient steps (see Fig. 8.19, bottom).

8.3 The *Demonstrator Platform* as a Development Tool for Research

In the following it will be shown how the CoMSON *Demonstrator Platform* can be used as an effective tool to prototype new algorithms handling different physical effects in one single framework. In particular the development of a method that allows a self consistent electro-thermal simulation of a n-channel power MOS-FET will be taken into account as a case study.

A description of the physical features of this device is therefore given in Sect. 8.3.1 where the main challenges arising during its design phase are also highlighted. The state-of-the-art modeling procedures actually adopted in industry to cope with these challenges will be then introduced, showing the lack of a method that permits to simulate consistently both the thermal and electrical behavior of the MOS-FET. To overcome this limit an extension of the model is proposed that makes use of a PDE-based thermal element to account for heat-diffusion at the system level (see [1, 14–16]). The reference implementation of the novel method inside the CoMSON *Demonstrator Platform* framework will be analyzed into the details in Sect. 8.3.2. It will be shown first how the modular high-level design allows for an extreme flexibility in defining methods and solution procedures to

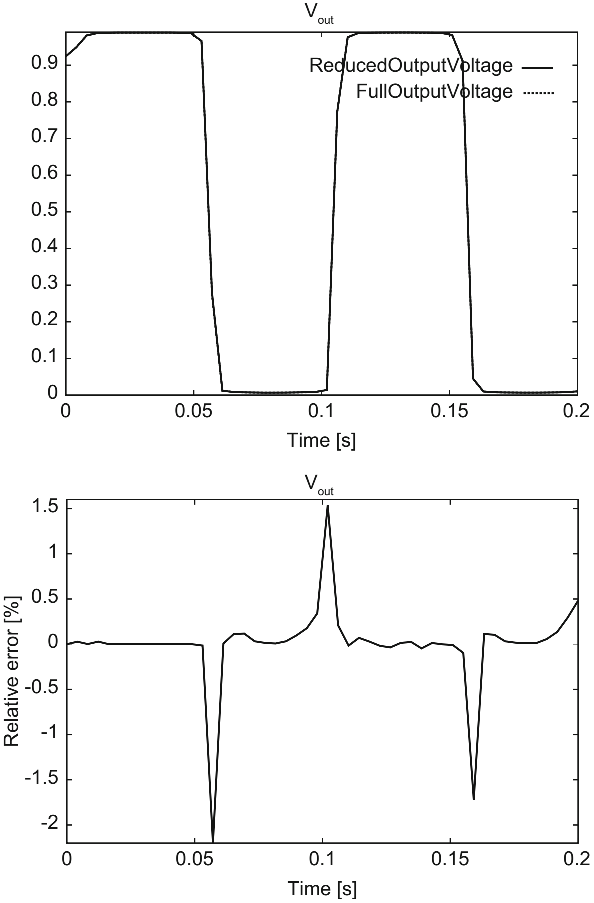


Fig. 8.19 V_{out} from circuit in Fig. 8.17 (*top*) in case when the transient simulation was run without the reduction of the TLINE (FullOutputVolatage) and with the reduction of the TLINE (ReducedOutputVoltage). Relative error between the both (*bottom*)

be used. Then the test-driven development (TDD) philosophy adopted during the implementation will be thoroughly exemplified, showing the progression from basic unit tests (class "A") to simple academic examples requiring the full-functionality of the algorithm (class "B") reaching finally the height of a real-life benchmark (class "C"). Simulation results obtained on this final problem will be then adequately illustrated and commented in Sect. 8.3.3.

8.3.1 Class "C" Benchmark: n-Channel Power MOS-FET

In this section is presented the "real-life" application upon which the algorithm proposed in [1, 16] will be tested to ensure its effectiveness. The choice to introduce the final benchmark at this stage reflects the actual development process of the CoMSON *Demonstrator Platform*, where a combination of top-down and bottom-up strategies was employed to correctly structure the modules and dimension

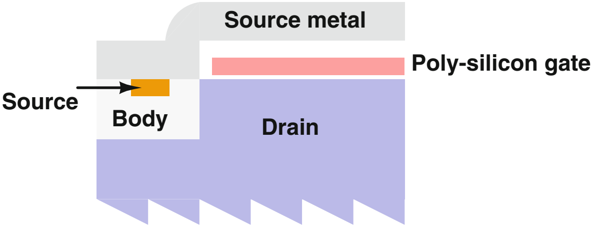


Fig. 8.20 Sketch of the cross-section of the power n-channel MOS-FET: only one metal layer and one polysilicon layer are employed during the fabrication process

the *Demonstrator Platform* itself. The correct identification of a final benchmark constitutes, within this framework, the starting point. This full-size problem will be then analyzed and divided into smaller parts in Sect. 8.3.2 where it will be also shown how already existing modules may be possibly re-used to provide specific functionalities.

8.3.1.1 Physical Features of the n-Channel Power MOS-FET

The device taken into account in the following is a vertical n-channel MOS-FET, mainly used for power applications [9]. As it can be seen in Fig. 8.20 (where a sketch of its cross-section is depicted) the drain contact is placed at the bottom of the die. To maintain the lowest possible production cost, the technology employed for the fabrication of the power MOS-FET exploits only one metal layer and one polysilicon layer. Source and body are thus short-circuited through the source metal layer (to avoid the turn-on of the parasitic npn bipolar transistor), while gate interconnects are laid-out using polysilicon. It should be stressed that the device surface is almost completely covered by source metal, with the only exception of a few regions where the metal layer is exploited to provide low-resistance gate connections.

The device layout (Fig. 8.21) is constituted by several elementary transistors cells connected in parallel to achieve the high current handling capability typical of power devices. The *active device regions* are organized in rows (each of which is a single wide-channel MOS-FET) as the polysilicon interconnects follow horizontal paths from the external gate metal. Due to the poor conductive property of the polysilicon layer, gate metal fingers are used to provide an alternative path between the gate pad and the elementary cells. However, as space has to be left toward the center to allow connection of the source bond wires, these fingers cannot extend from the upper to the lower part of the layout: some elementary cells are thus left without a direct metal connection.

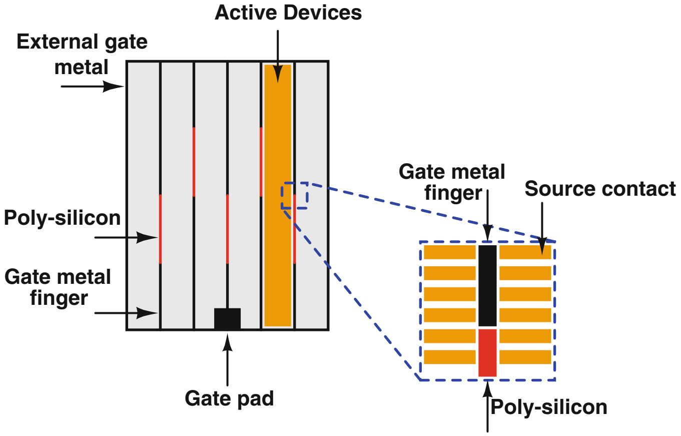


Fig. 8.21 Schematic layout of the power n-channel MOS-FET. Several active cells are connected in parallel. The external gate signal reaches every cell passing through metal fingers (low resistance) or polysilicon interconnects (high resistance)

When the power device operates at low frequencies, the effect on the elementary cells of a polysilicon gate connection instead of a metal one can be safely neglected as in this case the delay of the signal travelling through the polysilicon layer is small compared to the rise and fall times of the input. Anyhow, this condition does not hold when the switching frequencies get higher (as it is the case of many power application). In fact, due to polysilicon high electrical resistance, the signal given at the gate pad reaches some elementary cells with a delay that causes a non-uniform current distribution and the presence of a temperature gradient across the device surface.

8.3.1.2 State-of-the-Art Modeling and Simulation Procedures

The development of new models for power electron devices has been an active area of research in the last years, due mainly to the increasing use of these type of devices in many applications [11, 34, 40]. As the main interest of end users is in optimizing the performance of the circuitry driving the power stage rather than improving the device itself, the most part of these new models is “only” able to reproduce with a reasonable degree of accuracy the static or switching characteristic of the device as observed from external pins [4, 39, 46]. Anyhow they do not provide information on what happens inside the device, and therefore they are not suitable to be used in computer aided tools to improve the layout design of power devices themselves.

To overcome this weakness in [9] a *lumped-element distributed modeling approach* was introduced, which allowed to observe local maxima in the current density distribution. The main idea was to exploit the concepts employed for high frequency modeling of microstrips to describe the electrical behavior of polysilicon and metal interconnects of a power device. Hence the layout design information is used to generate a scalable electrical network feasible to be analyzed by any spice-like circuit simulator. The resulting netlist has a hierarchical structure based upon three basic building blocks, representing respectively:

- Metal over passive area,
- Polysilicon over passive area,
- Polysilicon over active area.

A thorough description of the model is reported in [9].

In [32] the model proposed in [9] was extended to account also for the self-heating of the cells. Still the dependence of the electrical characteristics of each cell on the dissipated power remains purely local, and thus mutual-heating effects are not being caught.

8.3.1.3 Extension: PDE-Based Electro-thermal Circuit Element

To allow the description of non-local heating effects the model presented in [9, 32] will be complemented in the following by the introduction of a PDE-based electro-thermal circuit element [13]. The general idea of this further extension is presented in Fig. 8.22. Starting from available layout and/or package geometry information, a thermal element model is derived directly from PDEs describing heat-diffusion

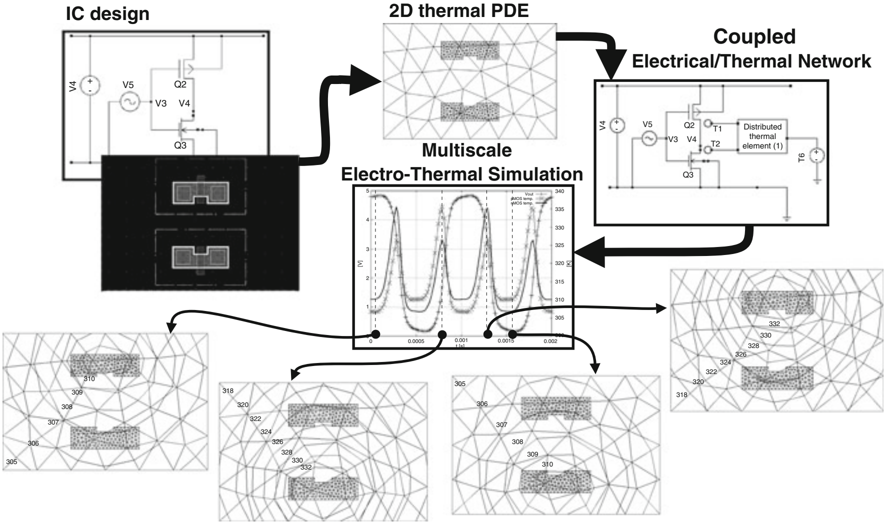


Fig. 8.22 Automated design flow for the electro-thermal simulation of ICs. A thermal element model is automatically constructed from available circuit schematic and design layout, permitting the set-up and simulation of an electro-thermal network that accounts for heat diffusion at the system level

at the system level. By imposing suitable integral conditions this element is casted in a form analogous to that of usual electrical circuit elements, so that its use in a standard circuit simulator requires only the implementation of a new element evaluator. This permits to describe possibly non-linear heat-diffusion phenomena on 2D/3D domains without modifying the main structure of the circuit solver. The mathematical model standing at the base of this approach is thoroughly treated in Chap. 2.

A particular spatial discretization scheme [17, 23, 28–30, 53], based on the use of completely overlapping non-nested meshes, was chosen to cope with multiscale issues. This method has two main advantages for the application at hand:

1. It allows to cover the whole thermal domain with a uniform triangulation without having to excessively refine the mesh to capture small geometrical features,
2. It allows to generate a mesh for each circuit element only once and deploy it at different positions on the IC with a significant time improvement during the mesh generation phase.

The latter feature may also give performance gain if an optimization of the relative device placement is to be performed. In the end the adopted algorithm resembles what it is known in literature as a *brute-force* approach [12], the only difference being that in this case no a-priori interpretation in terms of a circuit netlist is necessary for the discretized PDEs.

8.3.2 Implementation and Development Procedure

In a research project it is normal and desirable that software requirements undergo extensive modifications as development proceeds. In fact, rigidly defining even such basic things as data structures and interfaces at initial stage could severely limit possible future extensions. Nevertheless developers need clear indications to structure a software and start coding without loosing focus on the application. The definition of appropriate benchmarks constitutes an effective way to provide these indications, as it enables a unique and non controversial way of assessing validity of design choices while it allows for an extreme flexibility in implementation. In the following it will be shown how this theoretical principle was applied to prototype the algorithm briefly introduced in Sect. 8.3.1.3 within the CoMSON *Demonstrator Platform* framework.

8.3.2.1 High-Level Design

It is possible to see from the definition of the mathematical model presented in Chap. 2 that the implementation of the algorithm of interest requires functionalities that are typical either of circuit simulators, or of finite-element solvers. The `Octave Circuit Simulator` module (OCS) provides CoMSON *Demonstrator Plat-*

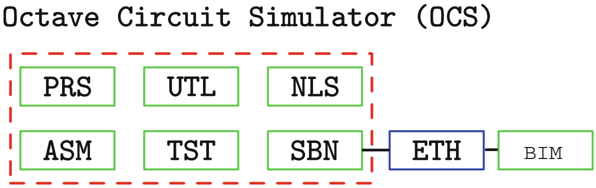


Fig. 8.23 High-level design for the implementation of the algorithm presented in Sect. 8.3.1.3. The basic features required by a circuit simulator are provided by OCS. To implement the PDE-based electro-thermal element a new module (ETH) will be designed with the same interface as OCS element evaluators (SBN)

form with the former features, while `Box Integration Method` module (BIM) provides the latter.

To implement the PDE-based electro-thermal element a new module (ETH) has therefore been devised to provide the necessary link between these different capabilities (see Fig. 8.23). The only constraint imposed by this decision is that the element evaluator should fit the structure:

```
[a,b,c] = M<elname>(string,prms,prmnms,extvar,intvar,t)
```

thoroughly described in the Intermediate File Format specifications [20].

8.3.2.2 Early Stage: Class "A" Test Cases

In the early stage of the implementation it is important to ensure the correctness of each module subroutine as soon as it is coded. This can be done exploiting `octave` testing capabilities that permits to insert test at the end of the function source code [38]. Notice that this feature provides the *Demonstrator Platform* with a simple but effective strategy to perform regression tests.

8.3.2.3 Intermediate Stage: A Class "B" Test Case

Once the designed module has been implemented and its basic functionalities have been assessed through class "A" test cases, a preliminary validation of the method is obtained applying it to a simple problem of academic size. In the case at hand the choice was to simulate the response of the CMOS-inverter circuit depicted in Fig. 8.24 to a 1 kHz sinusoidal input signal [14, 15]. The two MOS-FETs appearing in the schematic are modeled by a simplified version of the classic Shichman-Hodges model [47] with an added temperature pin (the actual parameters used in the simulations are collected in Table 8.2).

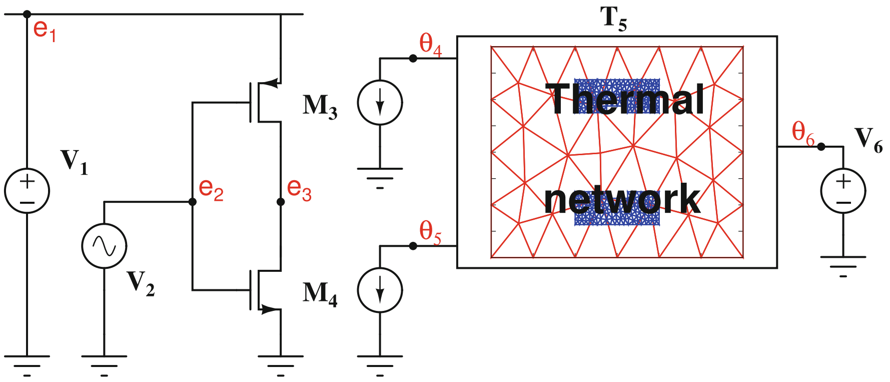


Fig. 8.24 CMOS-inverter electro-thermal network. Inside the thermal element the 2D mesh used for the approximation of heat diffusion on a distributed domain is shown

Table 8.2 Shichman-Hodges MOS-FET model parameters for the CMOS-inverter simulation

	W/L	μ_0	θ_0	V_{th}	r_d	C_{gb}	C_{gd}	C_{gs}	C_{sb}	C_{db}
nMOS	5	10^5	300	0.1	10^6	10^{-11}	10^{-12}	10^{-12}	10^{-12}	10^{-12}
pMOS	5	10^5	300	-0.1	10^6	10^{-11}	10^{-12}	10^{-12}	10^{-12}	10^{-12}

The impact of temperature is represented by a temperature dependent carrier mobility:

$$\mu(\theta) = \mu_0 \left(\frac{\theta}{\theta_0}\right)^{-3/2}, \tag{8.26}$$

where μ denotes the electron mobility for the n-channel transistor M_4 and the hole mobility for the p-channel transistor M_3 and μ_0 is the value of μ at the reference temperature θ_0. The total dissipated Joule-power is given by the simple expression:

$$P = i_{ds} v_{ds} , \tag{8.27}$$

where i_{ds} denotes the current flowing in the controlled current source appearing in the transistor model and v_{ds} is the drain-to-source voltage.

The thermally active regions on the IC substrate are taken to roughly correspond to the channel region of the transistors. Comparing Figs. 8.25 and 8.26 it can be seen that, while maintaining the same mesh refinement in the channel regions, the patched mesh greatly reduces the number of unknowns with respect to a standard conforming triangulation. Linear heat-diffusion is supposed to properly describe thermal effects on the layout (Table 8.3).

The plot in Fig. 8.27 shows the voltage waveforms and the corresponding values of the device temperatures. As it is expected the junction temperatures θ_4 and θ_5 are close to the ambient temperature value $\theta_6 = 300\,$K when the output is either in the ON or in the OFF state, as in such situation only small leakage currents flow in the

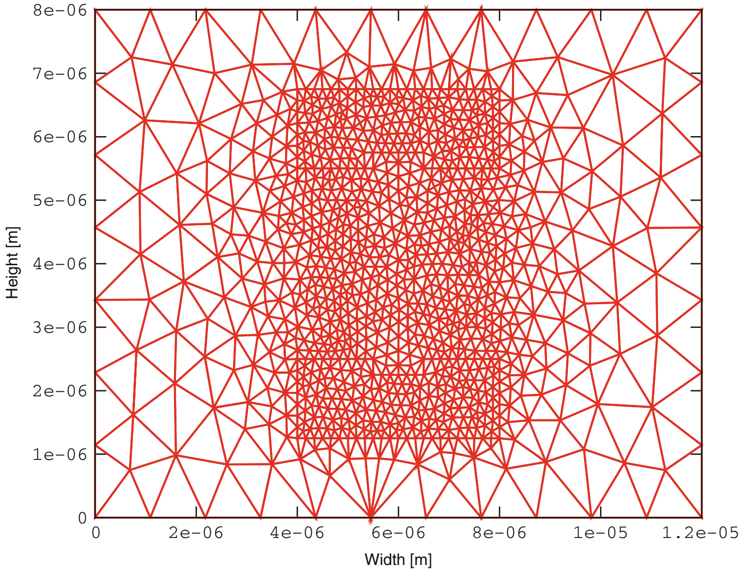


Fig. 8.25 Globally conforming triangulation of 2D chip-layout: 1,052 nodes, 2,066 elements and 1,052 unknowns

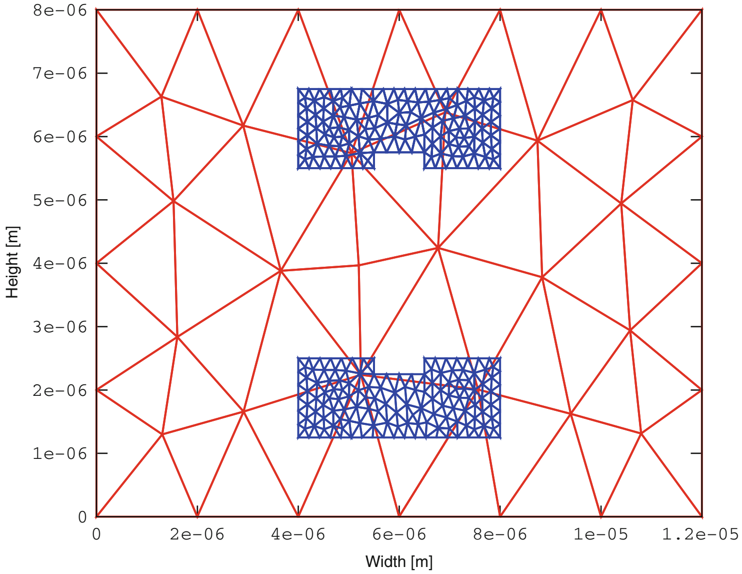


Fig. 8.26 Patched triangulation of 2D chip-layout: 339 nodes, 550 elements and 237 unknowns

Table 8.3 Heat diffusion equation parameters for the CMOS-inverter simulation

$\hat{c}_v$	$\hat{\kappa}$	$\hat{c}$	$\hat{\alpha}$
$1.5 \cdot 10^{-6}$	$1.5 \cdot 10^{-6}$	1	10

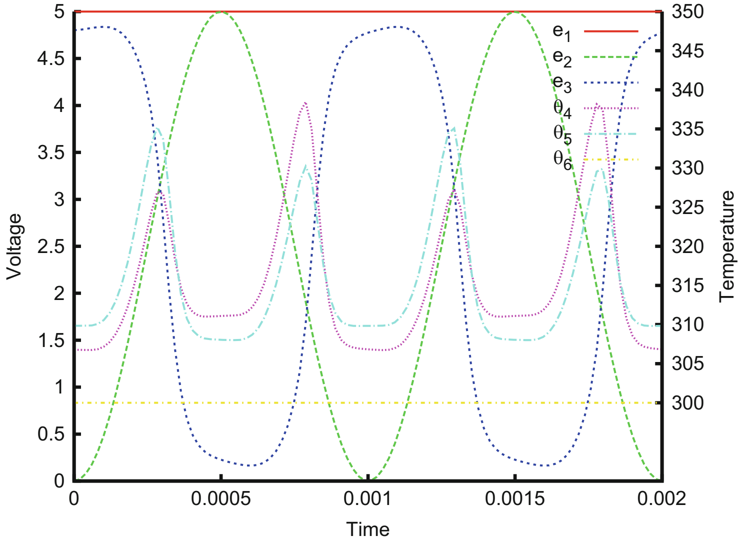


Fig. 8.27 Node voltages and junction temperatures plotted against time for two periods of an input sine voltage at the frequency of 1 kHz

devices, while the current flowing during the ON-to-OFF or OFF-to-ON transitions generates a relatively more significant heating. Figure 8.28 depicts the temperature distribution in the IC substrate at different instants during an OFF-to-ON transition. It can be noted that the heat produced mainly by the p-channel device (above), diffuses through the substrate and affects the n-channel device (below).

8.3.3 Simulation Set-Up and Results on the Class "C" Benchmark

A transient simulation of the turn-off switching of the device introduced in Sect. 8.3.1 is performed as a final validation (Fig. 8.29). This benchmark constitutes a major step toward a real industrial test case, due to its complexity that greatly exceeds the one of usual academic problems. The regularity of the n-channel MOSFET layout permits easily to show an important characteristic of the method, that is to say the possibility to replicate a fine mesh associated with a thermally active area at different positions in the die (Fig. 8.30). This feature allows for the creation of a *library* of electro-thermal devices in which a pre-computed mesh of their active region is included, diminishing thus the computational effort during the mesh

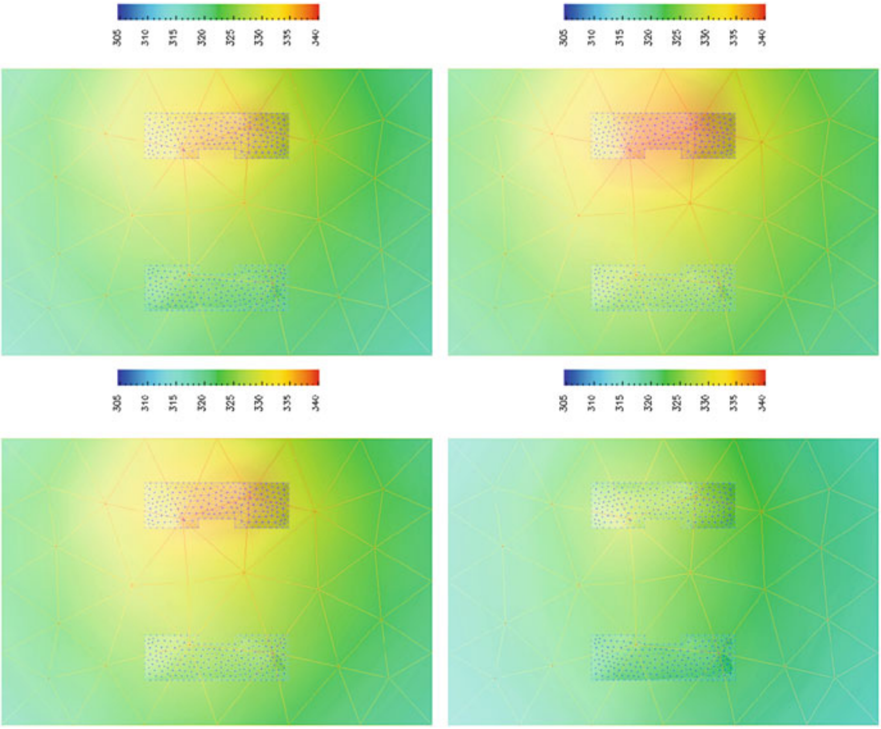


Fig. 8.28 Subsequent snapshots of the distributed temperature field taken during the first switching phase. The heat produced mainly by the p-channel device (above), diffuses through the substrate and affects the n-channel device (below)

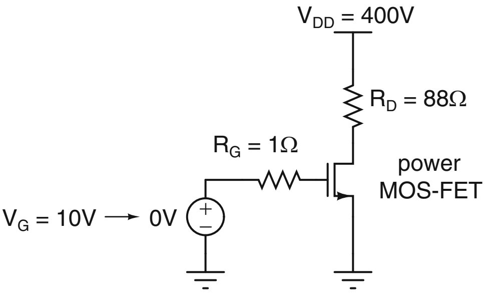


Fig. 8.29 Circuit used to simulate the turn-off switching of the power n-channel MOS-FET. The input signal switches at $t = 3 \times 10^{-9}$ s

generation phase and possibly enabling a performance gain if an optimization of the relative device placement is to be performed.

The electrical behavior of the power n-channel MOS-FET is described by the same *lumped-element distributed approach* presented in [9]. The electrical network

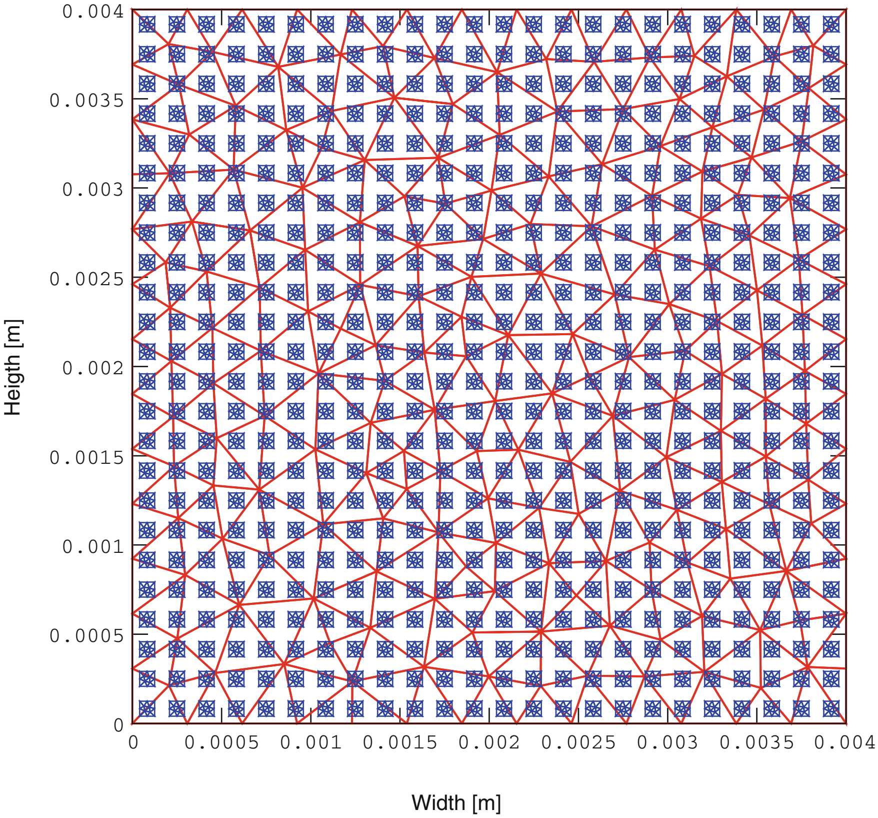


Fig. 8.30 Non-conforming meshes for the whole CHIP and the basic cells. The nMOS-FET model is here scaled to 576 active cells. A coarse grid (in *red*) covers the 4 × 4 mm die, while a fine one (in *blue*) is replicated at each active region position

Table 8.4 Basic cell parameters employed in the simulation of the n-channel MOS-FET turn-off transient. See [9] for more details

	R (series)	L (series)	R (ground)	C (ground)
Metal (passive)	10	10^{-15}	10^{12}	10^{-13}
PolySi (passive)	100	10^{-6}	10^{12}	10^{-12}
PolySi (active)	100	10^{-6}	–	–

is scaled to contain 24 × 24 active cells and 6 metal fingers. A simplified Shichman-Hodges model with an added temperature pin is used for each elementary transistor cell. Notice that the values of the parameters gathered in Tables 8.4 and 8.5, though fitted to provide realistic results, do not stem from any existing technology.

Thermal effects are supposed to be adequately described by a linear heat-diffusion equation, whose parameters are given in Table 8.6.

Table 8.5 Parameters of the simplified Shichman-Hodges MOS-FET model used to describe the behavior of each active cell

W/L	μ_0	θ_0	V_{th}	r_d	C_{gb}	C_{gd}	C_{gs}	C_{sb}	C_{db}
2.2	10^6	300	0.5	10^9	10^{-12}	10^{-15}	10^{-15}	10^{-15}	10^{-15}

Table 8.6 Parameters of the thermal element employed in the simulation of the power n-channel MOS-FET turn-off

$\hat{c}_v$	$\hat{\kappa}$	$\hat{c}$	$\hat{\alpha}$
10^{-4}	0.02	1,000	$4 \cdot 10^4$

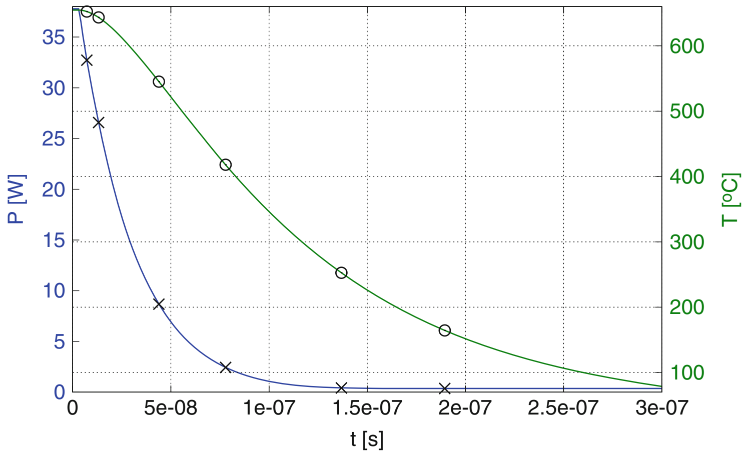


Fig. 8.31 Total dissipated power and mean temperature plotted against time for a turn-off transient. The sampled points refer to the snapshots presented in Figs. 8.32 and 8.33. A simple backward Euler scheme was adopted to time discretize the coupled system

Figure 8.31 shows the total dissipated power and the mean temperature of the device plotted against time. As expected to a lowering of the power corresponds a cooling of the device; however these two effects exhibit different relaxation times. The power densities and junction temperatures of the cells are shown respectively in Figs. 8.32 and 8.33 for six different time-points defined in Fig. 8.31. It can be clearly seen a delay in the propagation of the signal from the gate-pad in the lower part of the die to the single cells, and the presence of an hot-spot in the central upper part of the die for $t = t_2$ and $t = t_3$. Moreover the presence of a non-negligible temperature gradient over the device area is detected at times $t = t_1$ and $t = t_2$. Furthermore the different spatial distribution of heat density and temperature are an indication that non-local effects may not be negligible in estimating the device performance.

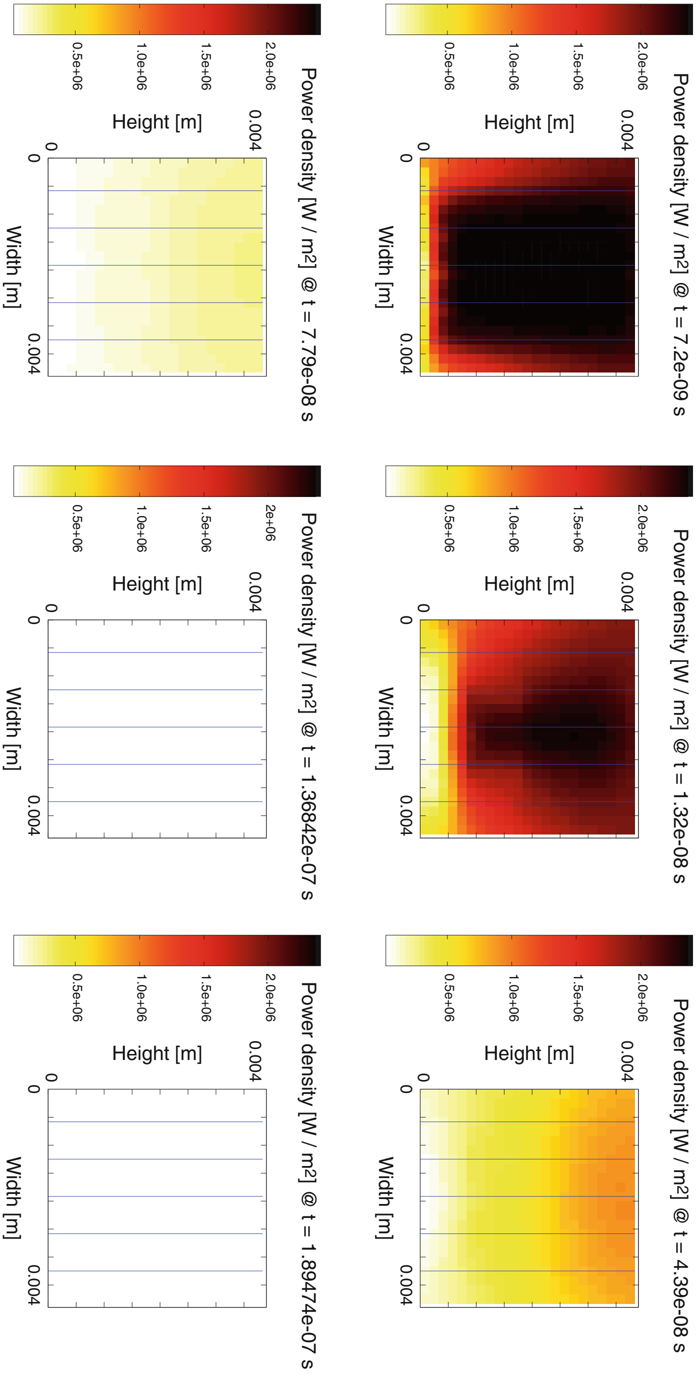


Fig. 8.32 Snapshots of the active cell power densities at the time points t_1, t_2, t_3, t_4, t_5, and t_6 defined in Fig. 8.31

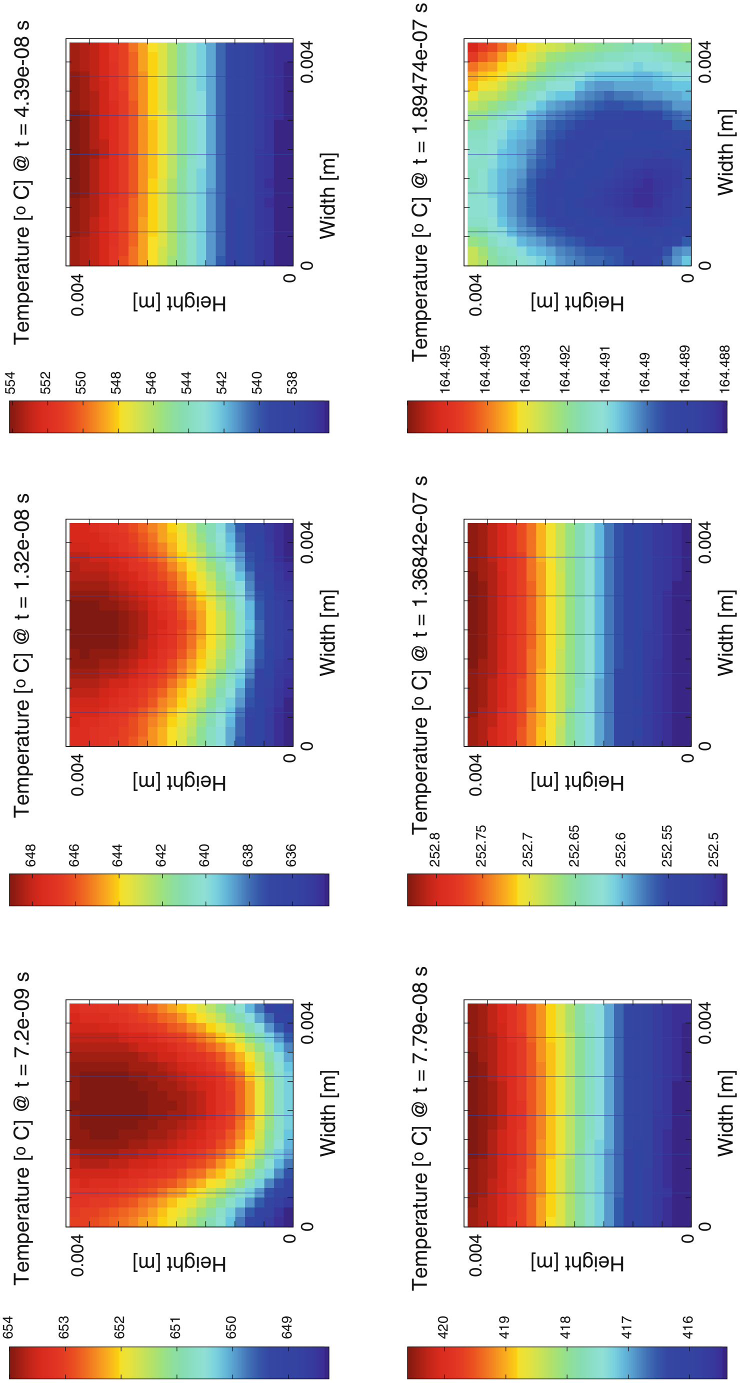


Fig. 8.33 Snapshots of the active cell junction temperatures at the time points t_1, t_2, t_3, t_4, t_5, and t_6 defined in Fig. 8.31

References

1. Alì, G., Bartel, A., Culpo, M., de Falco, C.: Analysis of a PDE thermal element model for electrothermal circuit simulation. In: Roos, J., Costa, L. (eds.) Proceedings of Scientific Computing in Electrical Engineering (SCEE) 2008, Espoo. Mathematics in Industry, vol. 14, pp. 273–280. Springer (2010)
2. Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., Sorensen, D.: LAPACK Users' Guide, 2nd edn. Technical report, SIAM, Philadelphia (1995)
3. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia (2005)
4. Aubard, L., Verneau, G., Crebier, J., Schaeffer, C., Avenas, Y.: Power MOSFET switching waveforms: an empirical model based on a physical analysis of charge locations. In: IEEE 33rd Annual Power Electronics Specialists Conference, PESC 02, Cairns, vol. 3, pp. 1305–1310 (2002)
5. Bai, Z.J., Meerbergen, K., Su, Y.F.: Arnoldi methods for structure-preserving dimension reduction of second-order dynamical systems. In: Benner, P., Mehrmann, V., Sorensen, D. (eds.) Dimension Reduction of Large-Scale Systems, Oberwolfach. Lecture Notes in Computational Science and Engineering, vol. 45, pp. 173–189 (2005)
6. Bechtold, T., Hohlfeld, D., Rudnyi, E.B., Guenther, M.: Efficient extraction of thin film thermal parameters from numerical models via parametric model order reduction. J. Micromech. Microeng. **20**(4), 045030 (2010)
7. Bechtold, T., Rudnyi, E.B., Korvink, J.G.: Error indicators for fully automatic extraction of heat-transfer macromodels for MEMS. J. Micromech. Microeng. **15**(3), 430–440 (2005)
8. Bechtold, T., Rudnyi, E.B., Korvink, J.G.: Fast Simulation of Electro-thermal MEMS. Springer, Berlin/Heidelberg (2006)
9. Biondi, T., Greco, G., Allia, M., Liotta, S., Bazzano, G., Rinaudo, S.: Distributed modeling of layout parasitics in large-area high-speed silicon power devices. IEEE Trans. Power Electron. **22**(5), 1847–1856 (2007)
10. Boisvert, R.F., Pozo, R., Remington, K.A.: The Matrix Market exchange formats – initial design. http://math.nist.gov/MatrixMarket/formats.html
11. Chen, Y., Lee, F., Amoroso, L., Wu, H.P.: A resonant MOSFET gate driver with complete energy recovery. In: Proceedings IPEMC 2000 the Third International Power Electronics and Motion Control Conference, Beijing, vol. 1, pp. 402–406 (2000)
12. Codecasa, L., D'Amore, D., Maffezzoni, P.: Compact modeling of electrical devices for electrothermal analysis. IEEE Trans. Circuits Syst. I: Fundam. Theory Appl. **50**(4), 465–476 (2003)
13. Culpo, M.: Numerical algorithms for system-level electro-thermal simulation. Ph.D. thesis, Bergische Universität Wuppertal (2009)
14. Culpo, M., de Falco, C.: Dynamical iteration schemes for coupled simulation in nanoelectronics. In: Proceedings in Applied Mathematics and Mechanics, PAMM 2008. Wiley (2009)
15. Culpo, M., de Falco, C.: Dynamical iteration schemes for multiscale simulation in nanoelectronics. In: Proceedings in Applied Mathematics and Mechanics, PAMM 2008. Wiley (2009)
16. Culpo, M., de Falco, C., Denk, G., Voigtmann, S.: Automatic thermal network extraction and multiscale electro-thermal simulation. In: Roos, J., Costa, L.R.J. (eds.) Scientific Computing in Electrical Engineering SCEE 2008, Espoo. Mathematics in Industry. Springer, Berlin/Heidelberg (2010)
17. Culpo, M., de Falco, C., O'Riordan, E.: Patches of finite elements for singularly-perturbed diffusion reaction equations with discontinuous coefficients. In: Fitt, A., Norbury, J., Ockendon, H. (eds.) Proceedings of the 2008 ECMI Conference, London. Mathematics in Industry. Springer (2009)
18. Datta, B.N.: Numerical Methods for Linear Control Systems. Elsevier Incorporation, Amsterdam/Boston (2004)

19. Davis, T.A.: UMFPACK. http://www.cise.ufl.edu/research/sparse/umfpack
20. de Falco, C.: Specification of an intermediate file format for the CoMSON demonstrator platform. Technical report, Bergische Universität Wuppertal (2006)
21. de Falco, C., Denk, G., Schultz, R.: A demonstrator platform for coupled multiscale simulation. In: Ciuprina, G., Ioan, D. (eds.) Scientific Computing in Electrical Engineering (SCEE) 2006, Sinaia. Mathematics in Industry, pp. 63–72. Springer (2007)
22. de Falco, C., Gatti, E., Lacaita, A., Sacco, R.: Quantum-corrected drift-diffusion models for transport in semiconductor devices. J. Comput. Phys. **204**(2), 533–561 (2005)
23. de Falco, C., O'Riordan, E.: A patched mesh method for singularly perturbed reaction-diffusion equations. In: Hegarty, A., O'Riordan, N.K.E., Stynes, M. (eds.) Proceedings of the International Conference on Boundary and Interior Layers – Computational and Asymptotic Methods, Limerick. Lecture Notes in Computational Science and Engineering, vol. 69. Springer (2009)
24. de Falco, C., O'Riordan, E.: Interior layers in a reaction-diffusion equation with a discontinuous diffusion coefficient. Int. J. Numer. Anal. Model. **7**(3), 444–461 (2010)
25. de Falco, C., Sacco, R., Jerome, J.: Quantum corrected drift-diffusion models: solution fixed point map and finite element approximation. J. Comput. Phys. **228**, 1770–1789 (2009)
26. Freund, R.: Krylov-subspace methods for reduced-order modeling in circuit simulation. J. Comput. Appl. Math. **123**, 395–421 (2000)
27. Glover, K.: All optimal Hankel norm approximation of linear multivariable systems and their L-infinity error bounds. Int. J. Control **36**, 1145–1193 (1984)
28. Glowinski, R., He, J., Lozinski, A., Rappaz, J., Wagner, J.: Finite element approximation of multi-scale elliptic problems using patches of elements. Numer. Math. **101**(4), 663–687 (2005)
29. Glowinski, R., He, J., Rappaz, J., Wagner, J.: Approximation of multi-scale elliptic problems using patches of finite elements. C. R. Math. Acad. Sci. Paris **337**(10), 679–684 (2003)
30. Glowinski, R., He, J., Rappaz, J., Wagner, J.: A multi-domain method for solving numerically multi-scale elliptic problems. C. R. Math. Acad. Sci. Paris **338**(9), 741–746 (2004)
31. Graf, M., Barrettino, D., Taschini, S., Hagleitner, C., Hierlemann, A., Baltes, H.: Metal oxide-based monolithic complementary metal oxide semiconductor gas sensor microsystem. Anal. Chem. **76**, 4437–4445 (2004)
32. Greco, G., Rallo, C.: XA integration in custom power MOSFET analysis flow. In: SNUG 2008 Proceedings, Bangalore (2008)
33. Hohlfeld, D., Zappe, H.: Thermal and optical characterization of silicon-based tunable optical thin-film filters. J. Microelectromech. Syst. **16**(3), 500–510 (2007)
34. Hong, S., Lee, Y.G.: Active gate control strategy of series connected IGBTs for high power PWM inverter. In: Proceedings of the IEEE 1999 International Conference on Power Electronics and Drive Systems, PEDS '99, Hong Kong, vol. 2, pp. 646–652 (1999)
35. IMTEK: Oberwolfach model reduction benchmark collection. http://portal.uni-freiburg.de/imteksimulation/downloads/benchmark
36. Liu, Y., Anderson, B.: Singular perturbation approximation of balanced systems. Int. J. Control **50**, 1379–1405 (1989)
37. NXP: SiMKit library. http://www.nxp.com/models/source/
38. Octave: homepage. http://www.gnu.org/software/octave/
39. Pagano, R.: Characterization, parameter identification, and modeling of a new monolithic emitter-switching bipolar transistor. IEEE Trans. Electron Devices **53**(5), 1235–1244 (2006)
40. Raciti, A., Belverde, G., Galluzzo, A., Greco, G., Melito, M., Musumeci, S.: Control of the switching transients of IGBT series strings by high-performance drive units. IEEE Trans. Ind. Electron. **48**(3), 482–490 (2001)
41. Romano, V.: Non-parabolic band hydrodynamical model of silicon semiconductors and simulation of electron devices. Math. Methods Appl. Sci. **24**(7), 439–471 (2001)
42. Romano, V., Rusakov, A.: 2D numerical simulation of electron-phonon MEP based model for semiconductors. In: ICTT-21, Torino (2009)

43. Romano, V., Rusakov, A.: Numerical simulation of semiconductor devices by the MEP energy-transport model with crystal heating. In: Michielsen, B., Poirier, J.R. (eds.) Scientific Computing in Electrical Engineering (SCEE) 2010, Toulouse. Mathematics in Industry, pp. 357–363. Springer, Berlin/New York (2012)
44. Rudnyi, E.: Model reduction software. http://modelreduction.com/Software.html
45. Rudnyi, E.B., Korvink, J.G.: Model order reduction for large scale engineering models developed in ansys. Lect. Notes Comput. Sci. **3732**, 349–356 (2006)
46. Schroder, S., De Doncker, R.: Physically based models of high power semiconductors including transient thermal behavior. IEEE Trans. Power Electron. **18**(1), 231–235 (2003)
47. Shichman, H., Hodges, D.: Modeling and simulation of insulated-gate field-effect transistor switching circuits. IEEE J. Solid-State Circuits **3**(3), 285–289 (1968)
48. Silveira, L.M., Kamon, M., Elfadel, I., White, J.: A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In: Technical Digest of the 1996 IEEE/ACM International Conference on Computer-Aided Design, pp. 288–294. IEEE Computer Society, Los Alamitos (1996)
49. SLICOT: The control and systems library. http://www.slicot.org
50. Spannhake, J., Schulz, O., Helwig, A., Müller, G., Doll, T.: Design, development and operational concept of an advanced MEMS IR source for miniaturized gas sensor systems. In: Proceedings of the IEEE Sensors Conference, Irrine, California, pp. 762–765 (2005)
51. Toledo, S., Chen, D., Rotkin, V.: TAUCS – a library of sparse linear solvers. http://www.tau.ac.il/~stoledo/taucs
52. Tombs, M.S., Postlethwaite, I.: Truncated balanced realization of stable, non-minimal state-space systems. Int. J. Control **46**, 1319–1330 (1987)
53. Wagner, J.: Finite element methods with patches and applications. Ph.D. thesis, EPFL, Lausanne (2006)
54. Woellenstein, J., Boettner, H., Plaza, J.A., Cane, C., Min, Y., Tuller, H.L.: A novel single chip thin film metal oxide array. Sens. Actuators B: Chem. **93**(1–3), 350–355 (2003)

Chapter 9
eLearning in Industrial Mathematics with Applications to Nanoelectronics

Giuseppe Alì, Eleonora Bilotta, Lorella Gabriele, Pietro Pantano, José Sepúlveda, Rocco Servidio, and Alexander Vasenev

Abstract The main topic of this chapter is a detailed exposition of CoMSON's attempt to give a contribution in the synergetic process of integration of research and training between Academia and Industry.

9.1 Introduction

In recent years, the use of new information and communication technologies in educational context has promoted a large spreading of innovative electronic learning environments (eLearning). The purpose of an educational environment is

G. Alì (✉) • P. Pantano
Department of Physics, University of Calabria, via P. Bucci cube 30/B, 87036, Arcavacata di Rende, Cosenza, Italy

INFN, Gruppo coll. Cosenza, Arcavacata di Rende, Cosenza, Italy
e-mail: giuseppe.ali@unical.it; pietro.pantano@unical.it

E. Bilotta • L. Gabriele
Department of Physics, University of Calabria, via Pietro Bucci 17/B, 87036, Arcavacata di Rende, Cosenza, Italy
e-mail: eleonora.bilotta@unical.it; lorella.gabriele@unical.it

J. Sepúlveda
Applied Research & Technology for Infocomm Centre – InnoVillage INV C210, Singapore Polytechnic, 500 Dover Road, Singapore 139651
e-mail: sepulveda.sanchis@gmail.com; pepe.sepulveda.sanchis@gmail.com

R. Servidio
Department of Languages and Education Sciences, University of Calabria, via P. Bucci cube 18/B, 87036, Arcavacata di Rende, Cosenza, Italy
e-mail: servidio@unical.it

A. Vasenev
Numerical Methods Laboratory, "Politehnica" University of Bucharest, 77206, Bucharest, Romania

Department of Engineering and Construction Management, Twente University, P.O. Box 217, 7500AE Enschede, The Netherlands
e-mail: vasenev@gmail.com

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_9

to support learning, but it can also be used for transfer of knowledge and training. Several eLearning systems have been designed and developed to deliver educational contents for different purposes: many postsecondary educational institutions and universities offer entire degree programs via distance education; many companies use distributed learning for internal training in order to control expenditures and at the same time to promote/encourage a flexible and quick way of improving the acquisition of knowledge and skills within a company.

Clearly, the importance of eLearning for training is directly proportional to the speed of innovation of a specific field of application. In this chapter we concentrate on eLearning in industrial mathematics, with application to micro- and nanoelectronics. Microelectronics is a field characterized by high specialization and high level of innovation. The rapid development of new microelectronic devices and technologies requires new skills to keep up with the current technological innovations. A possible strategy to face the worldwide competition is to adopt online educational and training systems to improve quickly the learning competences of the internal people. eLearning is currently used in microelectronic industry for training of personnel, usually by means of eLearning courses provided by dedicated companies.

A key aspect of training in microelectronics is its highly scientific content. Innovation in microelectronics is strictly related to scientific and technological research, usually performed in private research facilities, but also in collaboration with universities. The role of university becomes especially relevant when the innovation comes from joint university-industry research.

One of the main aims of the CoMSON (Coupled Multiscale Simulation and Optimization in Nanoelectronics) project is to define and to develop a system of eLearning in Industrial Mathematics with applications to Microelectronics, in order to facilitate the exchange of information; to share resources, scientific and educational materials; to create common standards; to facilitate the use of advanced tools. The common idea of this project is to create a bridge able to fill the gap that exists in the knowledge flow from University to Industry and vice-versa, in the field of microelectronics above all when a stronger competition among the industries and when the activities are covered by industrial secrets.

The main topic of this chapter is a detailed exposition of CoMSON's attempt to give a contribution in this synergetic process of integration of research and training between Academia and Industry.

We start the chapter with an overview of the development of eLearning methodologies, and their application in microelectronics (Sect. 9.2).

Given the importance of the relationship between industry developing and training, in the last years the European Union has funded several industrial projects devoted to use eLearning methodologies as a new training strategy. The aim of these projects was to design new platform architectures able to deliver advanced courses by using information technology infrastructure. Then, another concurrent goal of the current research in the eLearning field, concerns the development of new methodologies for content creation. eLearning educational contents are often a transposition in electronic form of the traditional didactical materials. This educational scheme implies a rigid user interfaces and then an unusable interactions procedure. It is very important to ensure to the largest number of users the use of the technological tools for educational purposes. Specifically, industrial and university should collaborate to design and test new didactical modalities to deliver educational contents by using eLearning tools. Clearly, the eLearning environments should take into account the different cognitive style of the final users. But also, it is important to assure that student's interactions with the system interface are as natural and intuitive possible. This could require a revision of the current interaction paradigm, providing the designing of new adaptive Graphical User Interfaces (GUI).

Today research in the field of GUI design has achieved important successes. An important aspect of the modern GUI is their customizability, which determines how users interact with the system and the tasks they need to perform in the environment. This design approach includes a strong relationship between learning cognitive process and graphic designer, which must know the principal aspects of the learning theory. Thus eLearning GUI must share cognitive aspects and didactical needs of the final users, in order to support the learners during their learning tasks, rather than being a mere use of advanced technologies. The GUI design process must be based on educational models and outcomes that suggest how people learn with the support of the technological tools. For instance, the integration of multimedia tools must be carefully integrated in the didactical environment, in order to avoid cognitive load problem that can affect the learning process. In fact, the use of eLearning environments does not mean to reject the traditional teaching strategies, such as simulation, cooperative work, experimental activity, and problem-based strategies. Contrarily, an eLearning environment should integrate these teaching strategies, in order to motivate the learner. In other words, to design an efficient and motivating eLearning environment, it is important to focus on the needs and goals of the students involved.

Sections 9.3 and 9.4, as an exemplification, illustrate the design and development of the CoMSON platform. The design phase has represented an important challenge for us, because we have identified the user requirements and then we have tried to implemented a platform which could respond to their needs. Next, we have adopted this methodology to design and implement the CoMSON information system, whose architecture includes several services devoted to support both communication and research activities.

One of the key ideas of CoMSON was to connect the information system and the eLearning system with the simulation environment, by appropriate GUIs. During the project's activity, we have designed some prototypical GUI to connect the eLearning system with the Demonstrator Platform (DP), which is an experimental platform to execute microelectronics experiments. Thus we have tried to design the GUI providing new and flexible functionalities, taking into account the traditional didactical techniques with the usability user requirements. This activity is expounded in Sect. 9.5.

The subsequent section deals with the learning content implementation. It is clear that the current development of hardware and software have stimulated researchers to experiment new teaching strategies. Such tools play an important role not only to organize and manage the educational contents, but also to deliver it by using meaningful visual representations. Despite recent advantages on eLearning technologies, yet much work remains to be done in terms of eLearning content creation. So far, several methodological proposals to design eLearning educational contents have been introduced and discussed. But, a didactical methodology to create educational contents based on specific guidelines is not yet available. In addition, the production of a high quality of learning material is important for students that use eLearning environment.

A common idea among eLearning developers concern the content creation, which takes much time and energy and often the course deliver insufficient and appropriate didactical contents. Thus, the eLearning researchers' community challenge is to develop new didactical approaches, in order to improve the content creation process. To this end, during the CoMSON project, we have tested and proposed to the involved partners, to experiment a didactical methodology oriented to create eLearning contents.

Taking into consideration the diversity of the content creation, the University of Bucharest has developed a set of eLearning materials and courses based on Bloom taxonomy. Their contents included not only theoretical description of the numerical optimization, but also practical applications and pseudo-code to test problems and models. The theoretical aim was to introduce to the students the optimization methods, giving them a well understanding of the algorithms. Tutorial documentation and other didactical material were delivered to the students as cognitive support to improve their skills.

Section 9.6 ends with the presentation of a document which was distributed among the participants to CoMSON, providing some hints and suggestions to turn available didactical material into seminal eLearning courses. The outcome of this strategy for obtaining learning contents was not promising. Indeed, in our experience a dedicated financial effort is needed specifically for learning contents creation, and this possibility of investment was not available within the scope of the project, whose main aim was research and training.

For this reason, the last part of the eLearning activities within CoMSON took to a different view, exploring the possibility of using a blended learning approach. This methodological approach, described in Sect. 9.7, was based on Problem-Based Learning (PBL) teaching strategies. In addition, the PBL consider the assessment as integral part of the students' learning process.

Thus to test this methodology we designed and performed two empirical sessions, whose aim was to involve University students' to create educational contents. All the students worked in groups and their developed specific topics according to the course programme. At the end of the course, the teacher evaluated the students' project analysing the quality of the educational contents. In this phase, we did not evaluate the educational value of the developed contents, from a learning standpoint for other students.

In Sect. 9.8 we describe the platform evaluation, commenting on the results of a survey conducted on the actual users of the CoMSON platform. According to the CoMSON project aims, we evaluated the information system architecture designed and developed. The basic questions concerned to know the user opinions' to use about the communication tools (e.g., mail, eLearning platform, etc.) used during the CoMSON project activities.

Finally, we underline some perspectives for future work. Recently, a number of new eLearning applications have been developed. The future challenge is to design and implement advanced eLearning functionalities based on GUI that provide easy interaction modalities attracting the students' interest. Then future work in the eLearning field, not only for microelectronics applications, is to address the problems concerning the didactic effectiveness of the eLearning applications and new procedure to create educational contents.

9.2 An Overview on eLearning

9.2.1 An Historic Perspective

eLearning, as we know it, is a relatively recent methodology. Nevertheless, its roots go back to the beginning of last century. The first teaching machines were developed by the U.S. psychologist Sidney Pressey in the early 1920s [62]. These tools were based on a very simple technology which included the submission of applications to the students, the assessment of the answers' correctness and the subsequent re-submission of the same questions in case of errors. The student's behavior was modified by the feedback obtained from the machine, as long as they would acquire an accurate knowledge of the contents.

From 1970 and until the early 1980s, Computer-Assisted Instruction (CAI) became greatly widespread [43]. CAI systems were based on exercises that included "drill and practice", tutorials and Intelligent Tutoring Systems (ITS). In these systems, the computer was programmed to teach students to acquire specific knowledge and skills. For each answer, correct or wrong, subjects received a feedback that could be either textual or graphical, such as a smile or an explosion. To acquire knowledge and skills to the highest level, it was necessary to overcome the lower levels to get to the higher ones [50]. These forms of learning were heavily influenced by the behaviorist theories (based on the stimulus-response experimental paradigm), that were unable to explain, or to encourage, the complex forms of human thinking needed to learn the meanings, to solve problems, to transfer skills to new situations, generate new ideas, and so forth.

More sophisticated was the ITS, a particular form of CAI system, developed between 1980 and 1990 by researchers of Artificial Intelligence [56] and dedicated to simulate problem-solving tasks, decision making strategies, etc. Specifically, an ITS is an educational software that records the students' work and gives back them a specific feedback. The way in which a student performs a specific task is compared with an expert algorithm that monitors the user's behaviour. When the ITS detects a discrepancy in the student's learning performance, it proposes an appropriate tutorial as educational support. Since the ITS collects information on the students' performances, it can evaluate the work done and provide individualized instructions in the problem solving strategies, suggesting them which topics to improve. Thus the ITS represents a first step towards a modern use of the technologies in the educational field [66].

The broad development of the digital technologies have influenced the way in which people access and manage both information and knowledge. This process has radically modified the conventional concepts of education and training, promoting new teaching methodologies. All these innovations have radically changed the current educational and training viewpoints. eLearning use the modern Information and Communication Technology (ICT) for learning purposes. ICT have shown a great potential in providing new tools and services to support traditional educational approaches. Strictly speaking, eLearning is a way of teaching and learning based on the delivery of online educational contents, via all available electronic media, including Internet, intranets, extranets, satellite broadcasts, audio/video tapes, interactive TV, and CD-ROMs. Thus, technology is used for designing, distributing, managing, spreading, and assessing training by carrying out personalized educational paths [72].

Web-based instruction studies have given considerable attention to flexible curricula, in order to provide adaptable and personalized learning programs [45]. Specifically, curriculum sequencing aims at designing and delivering optimal students' learning paths. This is useful since every learner has different background profile, preferences, and learning goals [28]. In this perspective, eLearning concerns the computer-based implementation of an educational system, where teacher and learner work together to achieve a common educational goal. In order to improve, from a cognitive point of view, the learning process using eLearning systems it is necessary to consider different characteristics, such as student's cultural background, technical and software equipments, and cognitive abilities of the students.

In the more recent years many educational systems and didactic approaches have been developed in the eLearning field, aimed at supporting students' interaction with digital educational materials [24]. These systems are based on adaptive algorithms that analyzing the subject's cognitive profile are able to create personalized learning paths. In an educational adaptive system, the optimal learning path aims at maximizing a combination of the learner's understanding of the courseware and the efficiency of learning.

The conceptual framework of eLearning can be summarized in the sentence "any time, any place, anywhere", that is, supporting students and teachers that live far from schools or universities and then increasing the life-long education cycle. This general program underlines a dramatic change in the traditional learning paradigm. The foreseen new learning paradigms should make provision for [40, 61]:

- An active and participating role of learners.
- A strong sense of presence and belonging (groups, working communities, virtual classrooms).

- A personalization of the learning path, by means of an articulated system of instrumental and human resources at disposal.
- A thorough exploitation of network hypertextuality as place, mean and social environment of learning.

The core of eLearning is the platform for managing the distribution and the use of educational material dedicated to training. An eLearning platform supports administrative functions, such as student's registration, assessment, and tracking of user's attendance (number of accesses, connection time, evaluation, and test results, etc.). Along with these services, an eLearning platform should also have interactive virtual classroom equipped with suitable tools [10]. A virtual classroom is an interactive environment where users can interact in a synchronous way (e.g., videoconference, audio conference, chatting, etc.), and an asynchronous way (e.g., web pages, web forum, e-mail, document repository, etc.), or in mixed mode. However, both communication modalities are available on Internet (e.g., streaming video, streaming audio, etc.). In fact, the main characteristic of an eLearning system is to overcome the obstacle of geographical location and to minimize the time constrains [4, 6].

In recent years, many educational and enterprise institutions have adopted eLearning systems to promote lifelong learning programme. This phenomenon has been favored by the Internet era, the development of communication and network technologies, the improvement of network bandwidth and quality, the real-time transmission of high-quality video and audio contents. In spite of this technological innovation, many studies underline that eLearning is still based on online newspaper form and information transmission, and it fails to provide a higher level of learning that would differentiate it and make it better than the classical classroom [23].

According to Alexander [3], four aspects should be considered to design a successful eLearning system:

1. Students' learning experiences.
2. Teachers' strategies.
3. Teachers' consideration and preparation.
4. Teaching/learning environment.

Chen and Zhang [17] underline that often individual differences, such as background, goal, learning style, that exist among the learners are not taken into account when an eLearning system is developed. In order to reduce the "cognition overload" and disorientation, they have developed an eLearning architecture called Adaptive Learning System, based on Learning Style and Cognitive State, able to select the learning contents according to the learner's cognitive style.

Numerous benefits come from eLearning. For instance, according to Kirschner et al. [40] it increases the students' skills improving their training and educational strategies; the learner can study according to his/her own work place; the contents are always available at a low cost because it is sufficient to have an Internet connection. Among the eLearning disadvantages we can mention the lack of social interaction, the high cost to assembly learning materials in a multimedia format, the high costs necessary to constantly update the contents and to provide tutorial support, so that instructor and tutor may not always be available on demand, and so forth. However, most of the aforementioned disadvantages can be overcome by the blended learning approach, where different learning style are mixed.

Yongxing [78] reports a case study on blended learning, underling that the latter modality of learning provides a good principle and idea for the choice of learning methods when eLearning tools and environment become more and more popular. Moreover, the methods of blended learning may vary from time to time, place to place, person to person. Therefore, Kang and Fengli (2007) suggested that the key of blended learning is to transfer knowledge to a "suitable" person, in "appropriate" time, with "appropriate" technology, with "appropriate" teaching style and "appropriate" e-teaching methods [38].

Another recent innovation concerns the emergence of social networks computing, which opens new opportunities for institutional learning. Social network tools empower users to produce, publish, share, edit and co-create contents, offering new opportunities in the learning field. According to Ala-Mutka [2] digital social networking offers new participative functions and new ways for cooperation supporting and facilitating knowledge exchange and collaborative content production. All these services are encapsulated in the Web 2.0 technology. Web 2.0 represent the second generation of Internet-based services that facilitate the online collaboration among users.

Universities and other educational institutions use social networking technologies as a strategy to discover new and innovative ways to enhance learning, facilitating collaboration and knowledge exchange. So, Web 2.0 can really support Universities and Companies to design and implement independent, autonomous and personalized education systems – i.e., learners are able to set their own learning goals, to develop critical thinking strategies and plan the cognitive strategies to achieve these goals [33].

In last years many educational services have been developed. The aim of these systems is to support both teachers and students, not only in the creation and communication of educational materials, but also as an scientific setting to experiment new didactical methodologies to enhance the learning process. For example, Classroom 2.0 website, is a virtual social network environment for teachers. It offers them help and advice to use in the classroom Web 2.0 tools for students' learning. Some of these services are discussion forum to exchange ideas and didactical experiences, and other social tools to create interpersonal relationships.

9.2.2 eLearning in Microelectronics Industry

So far we have discussed general concepts in eLearning. In the second part of this section, we specialize these general concepts to the specific field of Microelectronics.

Microelectronics is a field characterized by high specialization and a high level of innovation. By enabling technological innovation in other sectors, notably information technology, communications, and manufacturing, its impact is both profound and enduring. It is an innovative field which includes advanced technologies and requires specific competences to control sophisticated software systems to design and implement new devices [5, 6].

The development of professional and personal abilities in microelectronics is more and more important not only for engineering students but also for human resources that work in industry. Designers and researchers use daily software applications for design, simulation and manufacturing electronic devices. In the last years we have witnessed a rapid development of both new microelectronics devices and technologies, changing the skills required by the technical personnel employed in this field. Besides, international competition and global economy represent a continuous pressure for microelectronics industry.

In this global scenario, the capacity to handle information, knowledge, and innovation is central for microelectronics industry. To face this worldwide competition many industries and companies have adopted advanced educational and training strategies to rapidly improve the competences of the internal people. This strategic choice involves substantial investments in human capital and active absorption of technology, not only to introduce new best practice manufacturing system that integrate automation, process and product innovation, but also to experiment new educational solutions to optimize the learning activities of the internal resources. Industrial organizations have the need to improve the internal training strategy developing new educational environments.

Today training environments for web applications try to satisfy the above mentioned needs by developing new interactive educational tools satisfying not only the industry needs but also the final user's requests. The current educational approaches, like constructivism, do not use all the potentialities of the web technologies to create and manage didactical contents in a productive way. This limit comes from the fact that the web environments do not always support the user needs from a cognitive perspective. To overcome this limit, many organizations prefer to use traditional didactical approaches, because their principal aim is to assure that the human resources can change in a productive way, improving the competitiveness.

To support the industry needs, many software houses offer the industrial companies not only software systems to design electronics circuit, but also educational support by using web technologies. For instance, Cadence has developed a flexible Virtual Classroom to train users in live training events. It is possible to attend virtual lectures, participate in laboratory exercises, ask questions, and receive feedback simulating the classroom didactical activities. Cadence virtual classroom offers the users many educational opportunities that cover the main topics concerning design and implementation in microelectronics, with the goal to facilitate the adoption of Cadence solutions. Mentor Graphics is another company active in electronic design automation which delivers didactical contents in microelectronics, by using online learning with interactive hands-on activities. Virtual environments allow to the subjects to manipulate virtual commands of the software interface, simulating real context applications. Virtual systems reproduce the software interface without changing the visual organization.

Although both Cadence and Mentor Graphics have realized virtual educational system devoted to the microelectronic learning, the educational functionalities of these platforms are locked and cannot be easily customized.

However, microelectronics industries have been investing a lot of economic resources to develop autonomously high educational systems that provide an efficient learning environment. The integration into web of learning materials for microelectronics is an ideal approach for training professional people to learn new skills. In this process, the knowledge flows from University to Industry and vice-versa is especially relevant. On the one hand, it is often apparent a mismatch between what is usually taught in university courses in electronic engineering and what are the real needs of microelectronics industry, and a direct contact between University and Industry is beneficial to the quality of the university education in microelectronics. On the other hand, University has an established experience in education and training, which can be exploited for the setting of a training environment for young employees in microelectronics Industry. Moreover, Information and Communication Technology (ICT), and in particular eLearning, can be an innovative bridge between Industry and University, enhancing an intrinsic collaboration. This collaboration would ensure an effective transfer of knowledge, integrating different perspectives of how engineering disciplines are coordinated in both engineering and educational sectors.

In the last few decades, the European Union has supported this need for information exchanges between University and Industry, funding many research projects devoted to ICT applications in microelectronics, with the goal to design and develop new educational materials and eLearning platform based on interactive and multimodal environments.

For instance, the general objective of the project LIMA (Learning Platform in Microelectronic Applications, 2003) was to design an eLearning system to strengthen three leading educational centers in three dependent critical disciplines of microelectronic design and test, with active support, guidance and feedback from industry [59]. The resulting eLearning system is a web-based training platform with the purpose to satisfy different user needs, applications and levels of extensions. The main idea is to train people for conceiving, designing, verifying, and testing electronics circuits and systems.

We mention also the project E-LIMM (E-Learning for Microelectronics Manufacturing, 2004), which addressed the problem of the shortage of highly skilled industrial staff in the microelectronics industry by creating high-quality training and e-learning courses modules [58]. The goal was to apply new technologies such as multimedia learning for training and further education of the people that work in microelectronics manufacturing.

The project INETELE (Development of Multi-Media Teaching Material for Interactive and Unified E-Based Education and Training in Electrical Engineering, 2006) has been carried out within the Leonardo da Vinci Programme funded by the European Union, and involved eight universities from eight members states [36].

The aim of this project was to create a set of multimedia educational material and software for basic education in electrical engineering, by using simulated animation, virtual laboratory and final exercises for students assessment.

Finally, we mention the project CoMSON, whose activities are documented in the present handbook. As an outcome of the project CoMSON, STMicroelectronics (Catania sales, Italy) has experimented new ways to create educational contents for eLearning courses, to be used for internal training. Part of this work was done in collaboration with University of Calabria (Italy). The educational contents are based on existing materials, which were translated in electronic form. The realized courses cover basic and advanced concepts, theory, practice and analysis used for microelectronics applications. In our opinion, this experience shows a novel perspective in the possible cooperation between Industry and University, related to the creation and delivery of educational contents by using eLearning methodologies.

9.3 An Integrated Platform for Advanced Training in Microelectronics

The goal of the project CoMSON was "to realize an experimental Demonstrator Platform in software code, which comprises coupled simulation of devices, interconnects, circuits, EM fields and thermal effects in one single framework. It connects each individual achievement, and offers an adequate simulation tool for optimisation in a compound design space" [21]. This simulation environment would be complemented by an eLearning platform and a virtual working place. The eLearning platform would connect academic institutions and microelectronics companies, which collaborated together to the design of educational contents, to be delivered by the platform. The learning contents would be created by standard authoring software. This is advantageous because the system virtually supports any kind of course material that can be stored inside the web server. The virtual working place was conceived as an interactive environment where researchers from different nodes of the CoMSON Consortium could perform joint work, at distance.

As is immediately apparent just from the synthetic statement of the general objectives of the CoMSON project, this was a very ambitious objective, both from a scientific and an educational viewpoint. As we have seen in the previous section, many eLearning environments are available, and many vendors provide courses specifically designed for microelectronics industry. The problem is that most of these tools and training material are not flexible enough to cover at once the wide range of topics concurring in real microelectronics applications, keeping track of the most advanced research results. Moreover, the eLearning courses are usually detached from the simulation environments actually used in the main microelectronics industries.

For this last point, it is worth noting that microelectronics teaching generally involves the use of equipment laboratory with software tools where learners can perform experiments and simulations. In many cases, eLearning environments lack of these properties making problematic to teach scientific concepts. Usually, eLearning platform include just specific functions that partially support the students during the learning process. For example, many eLearning systems deliver educational contents using video streaming, without or with restricted interaction mechanism. If this approach works well for many disciplines, in microelectronics education it is more problematic to adopt these systems with a limited interaction level.

An eLearning environment devoted to microelectronic should include different typologies of tools such as virtual or remote laboratory, interactive software systems to design and perform experiments, simulations and an interactive evaluation systems to assess the achieved learning. The eLearning system should monitor how subjects interact with virtual tools during the following phases: design, implementation and test of an electronic circuit. The system should also provide the user a feedback of the interaction, suggesting how to rectify possible errors.

Today many software houses that operate in this sector are inclined to share information using remote laboratories. A virtual remote laboratory is an extension of a real environment. It allows to the users to interact with the interface of a system, safeguarding them from possible risks. Recently, microelectronics web-based virtual laboratory architectures have been developed that allow to simulate activities very similar to the conventional laboratory setup. Students first design the circuit and then use Internet to access the virtual laboratory to implement it. Mohtar and collaborators [53] describe an example of virtual laboratory architecture developed to design and test microelectronics circuits. This system includes a realistic Graphical User Interface (GUI) that exhibits the properties of a real laboratory environment. The visual manipulation of the circuit designed in the previous sessions, can be freely compared with other circuits or move some components with other to verify how the system work.

As a further example, we mention iLabs, a virtual remote laboratory developed by the MIT and accessed through the Internet [52]. The virtual laboratory architecture includes many functionalities which expand the range of experiments that students can perform during their undergraduate studies and not only. One of the most interesting functionalities of iLabs is that it can be shared across universities or across other institutions. One of these platforms simulates a virtual laboratory devoted to microelectronics.

We are well aware that to realize a virtual education environment that includes the above-mentioned functions is a difficult task, but not impossible. However, this approach requires a considerable effort by the developers to design a virtual remote laboratory able to simulate all the phases that are involved in the activities within a real laboratory. To achieve these goals it is important to design GUIs able to comply with different user interaction methods.

Traditional eLearning environments adopt simple user interfaces with restricted interaction modalities. Actually, many eLearning platforms use a standard architecture based on a predefined set of commands that allows the users to manage courses and educational material. This approach is fine if a teacher uses an eLearning platform as a content management system to organize the lectures and to give

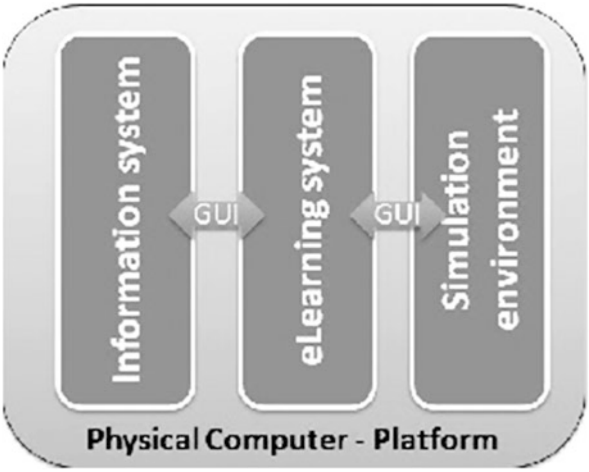


Fig. 9.1 Structure of the platform

support to the students with educational material, like lecture notes, tutorials, exercises, and so on. However, the recent findings in computer system and Human-Computer Interaction allow to design new eLearning platforms oriented to the microelectronics teaching. In this vision, virtual remote laboratory represent the future challenge to create innovative eLearning system to teach microelectronics contents.

As we have written at the beginning of this section, the CoMSON project has made a serious attempt to tackle the above mentioned problems. In fact, during this project, we designed an integrated platform devoted to the microelectronics industry. The main idea was to start from a set of advanced, scientific, research problems, intrinsically multidisciplinary, and with practical industrial relevance, trying to build around them a platform which would enable students, or young employees, to be properly trained. This platform includes three main components:

- Information system.
- eLearning system.
- Simulation environment.

All three main platforms components are integrated through Graphical User Interfaces. Figure 9.1 shows the interconnection among the components of the platform architecture designed to satisfy the project needs.

In the remaining of this section we describe the methodological procedure adopted in developing the integrated architecture, called CoMSON platform, which includes both the information system and the eLearning platform. The simulation environment is discussed in Chap. 7 (on the Demonstrator Platform).

9.3.1 User Needs Analysis

Following the Human-Computer Interaction methodology [25, 65], we collected the user information with the purpose to design a platform able to satisfy the requests coming both from the researchers that will work within the project and the students that will use the platform for educational training.

The design of the platform is based on the user analysis carried out by means of a questionnaire. The aim of this questionnaire was to gather information on the user needs concerning functions such as: communication, development, standards, and learning environment functionalities. The questionnaire was organized in five main sections:

1. Individuation of the final users of the platform.
2. Authoring and development tools, such as collaborative and communication tools and possible integration with specific software (for example, tools for the simulation of electronic circuits, etc.).
3. Communication and learning tools, aims and use.
4. Design of delivery models that provide the learning materials and resources, such as tools and communication services used in the learning environment.
5. General characteristics of the eLearning system and standards.

The questionnaire was sent to the group leaders in the different nodes of the Consortium. The reason for this choice was that the group leader was the best candidate to make an informed decision about the needs of the final users.

We collected and analyzed eight questionnaires which reflect the answer of each node. Here we detail the results of the user needs analysis, showing some of the most influential data used to design the platform.

In the past years the flexibility of Internet technology has favored the development of applications that allow to perform scientific simulation by using interactive educational environments. A virtual classroom, that is, the online environment in which students and instructors interact, can be an environment with synchronous interaction (the interactions happen simultaneously in real-time), or with asynchronous interaction (the interactions are delayed over time). It is also possible to have both kinds of interactions. This allows learners to participate according to their schedule, and be geographically separated from the instructor. Figure 9.2 shows a preference for both asynchronous and synchronous interactions.

Learning-by-doing is an educational approach which stresses the use of tools to enhance the learning process (Fig. 9.3). A great number of studies in eLearning focused on the importance to improve the learning strategies by using interactive tools. These tools not only offer the opportunity to interact with theoretical ideas in practical way, but also support the collaboration among students. The latter concept is an essential aspect in the process of constructing a shared knowledge among students.

To support the students' motivations during the learning process, an eLearning system should include many additional tools designed to deliver educational

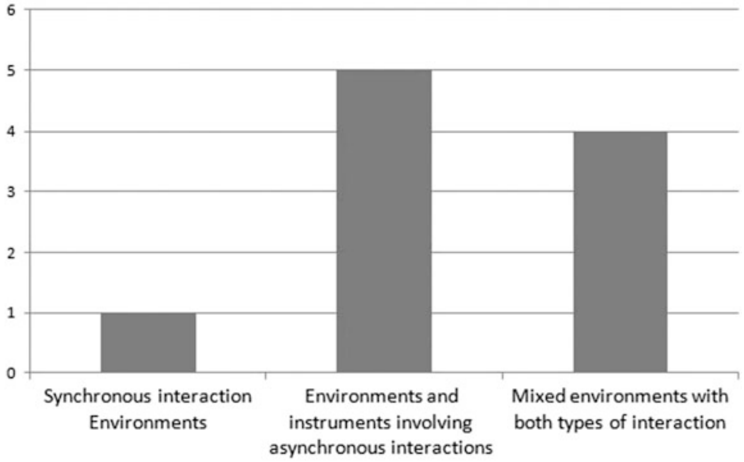


Fig. 9.2 Communication system in the eLearning environment

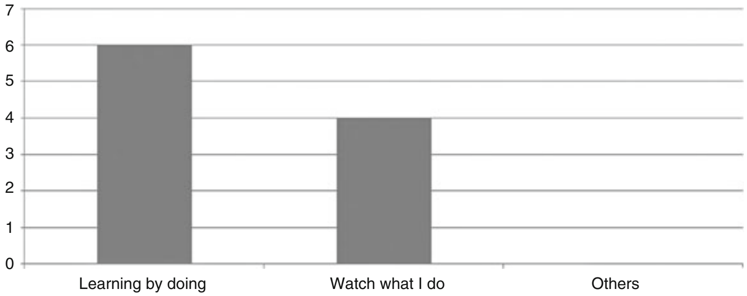


Fig. 9.3 The eLearning approach

contents (Fig. 9.4). Another important aspect of these tools concerns the possibility to perform many simulations to test the students' hypothesis, in so-called virtual classrooms, which try to extend the physical environment and interactions of a classroom to an online setting. Students can run simulations and manipulate objects analyzing in real-time the obtained results. In some cases, hands-on applications may be required.

eLearning is a collection of technologies, products, services and processes that support the learning process. In order to improve these aspects, it is important to design and implement specific Graphical User Interface (GUI) to connect these different services [69]. Figure 9.5 shows the main needs of the final user involved in the CoMSON project.

Assessment and testing are key components of any educational environment. Figure 9.6 shows the importance of self assessment in learning, for the subjects of the survey. The majority of the subjects chose the self-assessment modalities. By using this approach, the platform provides a checklist to help students assess

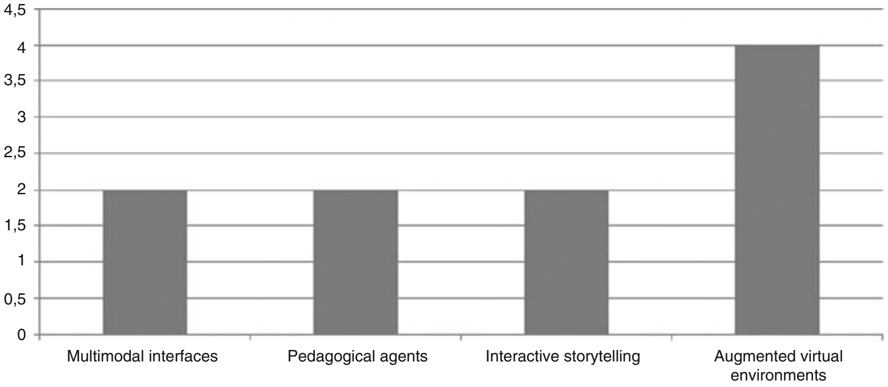


Fig. 9.4 Additional support to improve the eLearning environment

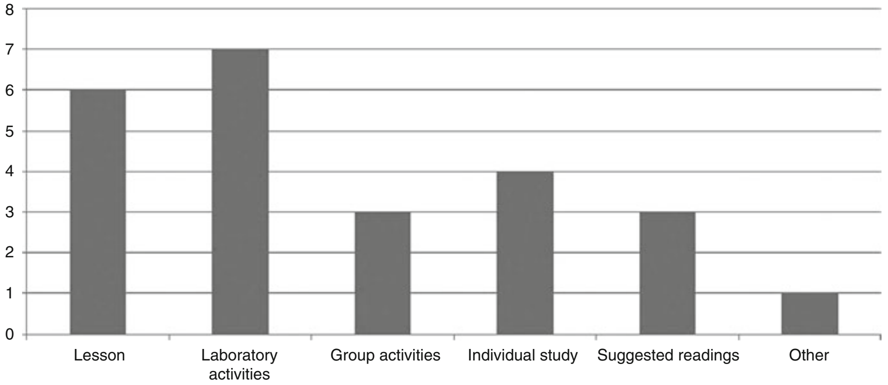


Fig. 9.5 Interface between eLearning system and Demonstrator Platform

themselves. Self-assessment is a formal evaluation technique, which enables a more fluid teaching and learning environment, which coincides nicely with the structure of eLearning environment.

Figure 9.7 shows the educational objectives that the CoMSON eLearning platform should satisfy during the project. In the initial phase, the eLearning platform will be an experimental educational laboratory in order to define specific guidelines to produce educational contents devoted to microelectronics field. After this analysis, during the project we have designed guidelines to write and organize educational contents in order to adapt it with the eLearning platform required.

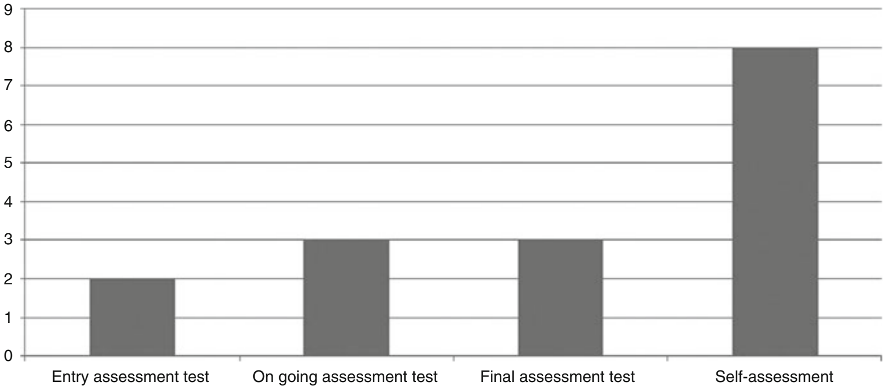


Fig. 9.6 Student evaluation modalities

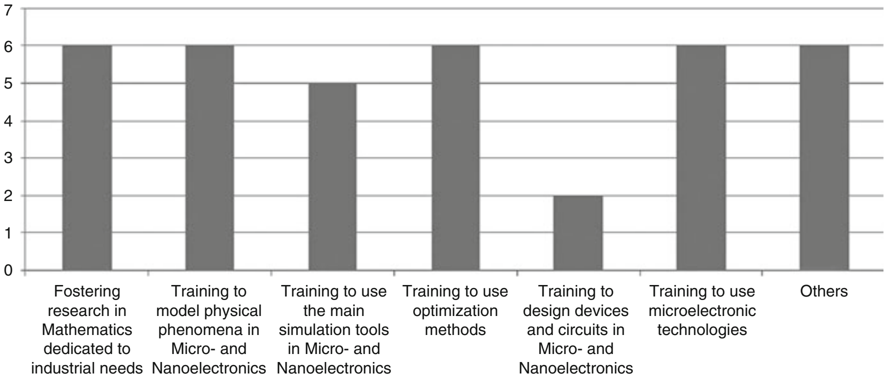


Fig. 9.7 Educational objectives of the eLearning

9.3.2 Platform Development

In order to develop the platform according to the user needs, we presented to the CoMSON scientific management board the results of the questionnaire. During this meeting we introduced a technical proposal aimed to define the design and implementation of the CoMSON platform.

According to the users' opinions, the CoMSON platform should possess the following characteristics:

- To be easy to use.
- To offer user-friendly help.
- To easily integrate existing digital materials.
- To support audio communication.
- To give the lecturer the capability to administer her/his own courses and to monitor the learners' progress and participation.

- To support multimodal interaction between the users through visual communication, and real-time display of users' activities.
- To support live document sharing applications.
- To offer an interactive and shared whiteboard.
- To integrate eLearning environment with other systems (e.g. Demonstrator Platform, Virtual Campus, and Virtual Working Place).

In short, users prefer a system that can support both types of communication and training: synchronous training (on-line lectures from a trainer on a specific theme, online meetings, on-line communication and collaboration between the members of a user group on a specific theme), and asynchronous training (autonomous training using interactive educational material and lecture notes, meeting minutes, administrative information).

From the analysis of the questionnaire results we have also taken some decisions on the following issues related to the eLearning platform:

- **User**. The final users of the eLearning platform will be students in microelectronics, but the system will be usable by microelectronics companies for employee training. At this stage of the project, the courses are being tested by CoMSON researchers, ERs (Experienced Researchers) and ESRs (Early-Stage Researchers). After this test and with the appropriate modifications the eLearning courses are made available to the general audience.
- **Authoring**. The underlying problems are: production of educational materials; collection and adaptation of existing educational materials for the eLearning tool; standardization of the educational material. In addition, all CoMSON partners agreed on the following points: each contributing professor can decide whether to take, or not, responsibility of formatting of the course. If some contributing professor does not want to take part in the formatting phase, he/she should provide the contributed material in any standard format for final adaptation. The professors will have the responsibility of the written contents (even if researchers will collaborate to write them). The writers will own the copyright of the written documents. CoMSON has to certificate the quality of the contents of the Learning Units, by university standards (certification of quality).
- **Educational aims**. The educational aims of the eLearning system are: fostering research in Mathematics dedicated to industrial needs; training to use the main simulation tools in micro- and nano-electronics. The users' future professional career will be: advanced modelling and simulation expert and designer.
- **Educational contents**. The eLearning system should provide tutorials on simulation steps (process, device, circuit, EM, optimization), including related software packages as examples. In general, no previous knowledge is needed by the user, but each Learning Unit has its own prerequisites. The eLearning system includes a wide range of topics including: Modelling of semiconductor devices; Introduction to electrical circuits; Electromagnetism; Interconnects; Basic numerical analysis; Numerical methods for DAEs.

The educational contents have been split in two categories: (1) Basic and (2) Advanced contents. Each category will consist of a minimal number of Learning Units (modules). The latter, will provide the modules on specific topics.

- **Technical specifications**. The educational contents should be importable by the main eLearning platforms used by microelectronic companies, according to the standards of IEEE P1484 [31] and Sharable Content Object Reference Model (SCORM 1.2) [67]. No specific software is required to be known by the user in advance.

9.4 The Components of the CoMSON Platform

As we have seen in the previous section, the CoMSON platform includes the following components: an information system, an eLearning platform, and a simulation environment (Demonstrator Platform). These components are connected by Graphycal User Interfaces, which will be discussed in the following section.

The CoMSON platform runs on a HP IA32 dual processor Xeon 32 bit 2 GHz frequency. The server has 15 GB of memory and two hard disk SATA architecture. The system uses a base Operative System (Linux Slackware) which hosts the following VMWare virtual machines:

(a) CoMSON, used as main communication and eLearning services.
(b) Kepler, used as Demonstrator Platform, with Current Version System (CVS) service.
(c) Copernicus, used to compile source code in the Demonstrator Platform (DP).

In order to synchronize the time between guest servers we install the Internet Systems Consortium – ISC- NTP Network Time Protocol server. A schematic representation of the system architecture is shown in Fig. 9.8.

Next, we detail the main components of the CoMSON platform.

9.4.1 The Information System

The CoMSON information system provides three main functionalities:

1. Documentation, authoring and distribution.
2. Exchange of knowledge.
3. Communication environment.

This set of functions is intended to enable interaction and knowledge exchange during the period of the project and after its completion. These services support the communication process between students and teachers as well as among researchers involved in the project. Ultimately, this initial user group will be enlarged including different academic and corporate institutions, cooperating on research. Furthermore,

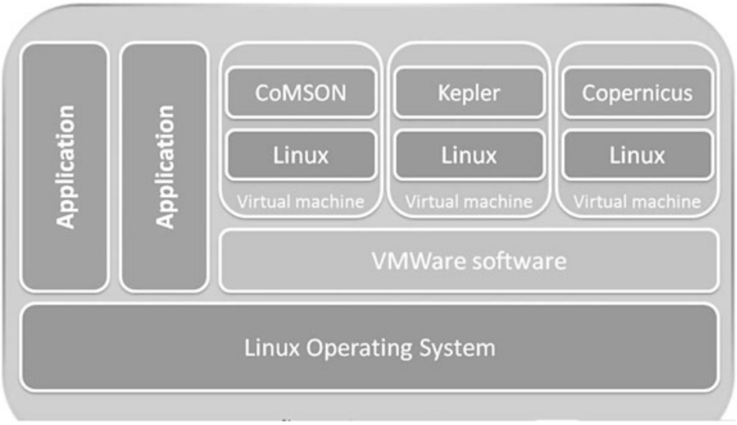


Fig. 9.8 CoMSON system architecture

the communication platform is the place where seamless knowledge exchange processes operate between academia and industry [10].

Its architecture, based on web technologies, enhances accessibility, ease of use and ease of integration with the other elements of the system [48]. The communication platform has been developed as an enabler for the above functions, comprising a set of interconnected tools. These tools are: web services including streaming server for content distribution; a forum and a mailing list system, for communication; and a documentation environment, which is used as a central information and document repository [1, 48, 64, 71].

We used Plone [7] as Content Management System (CMS) to implement the communication platform of the CoMSON project. Plone is an open source CMS built on Zope [75] application server. "Zope includes an Internet server, a transactional object database, a search engine, a web page template system, a through the web development and management tool, and comprehensive extension support". Plone, already, has a large user base and multitude of developers, usability experts, translators, technical writers, and graphic designers who are able to work with CMS [7].

The Plone workflow allows collaborative and cooperative management of content. Each object can assume different states. The objects state define whether an object can be accessible by others users. The Plone workflow includes four states: visible, pending, public, and private (Fig. 9.9).

The Plone team includes usability experts who have designed an intuitive user interface and attractive to manage the information. Other services, such as mailing lists, provide a channel to exchange information between registered users. There are mailing lists devoted to the different tasks of the project and for administration matters. This facilitates the communication among researchers on research and administrative aspects of the project.

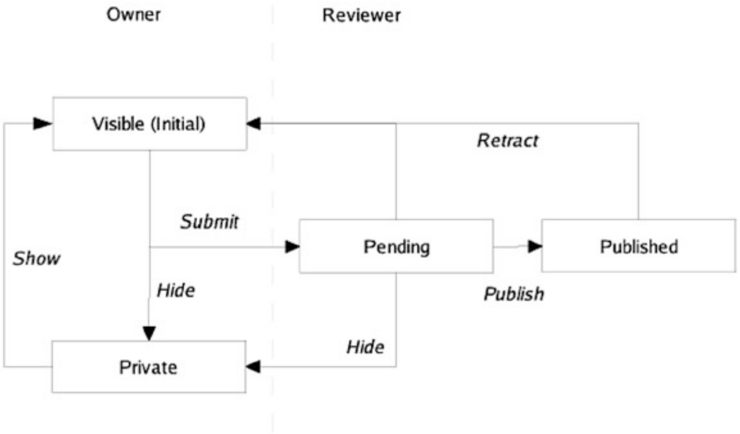


Fig. 9.9 The default workflow for Plone content

Mailing lists are implemented using Mailman, the GNU Mailing List Manager. Mailman is free software integrated with the web that allows easy management. Users can manage their accounts and the owner of the list can manage the lists.

9.4.2 The CoMSON eLearning Platform

A learning information infrastructure includes hardware, software, delivery mechanisms, and processes to manage educational paths. Hardware refers to servers, desktop computers, and mobile devices. Software refers to a Learning Management System (LMS), which is a software application for the administration, documentation, tracking, and reporting of training programs, classroom and online events, eLearning programs, and training content. A strong information infrastructure provides access to instructional content and support teachers and students to manage educational contents and deliver them in easy way. Usually, the term LMS is often used synonymously with learning information infrastructure, but an LMS by itself is usually only part of a learning information infrastructure.

The LMS used in CoMSON is based on Moodle open source software [63]. Moodle is a Course Management System (CMS), also known as a LMS or a Virtual Learning Environment (VLE). Moodle allows the management of courses, didactical modules, real-time and differed learning. Among the tools available to the teachers, we find authoring tools for creating lessons and assessment tests. In addition, to these standard tools, we would like to spend some words on a possible learning scenario, which might be consistent with the implemented Moodle platform.

The eLearning platform provides contents based upon the Sharable Content Object Reference Model (SCORM) standard [67], which allows the creation of standard contents that are exportable and executable on every SCORM compliant system. Moreover, the SCORM standard is integrated with distributed technologies, in order to develop a complete learning system. Intense research activity is ongoing on eLearning technologies especially focusing on accessibility, interoperability, durability, and reusability of components. Applying Web Service Technologies to a SCORM compatible LMS simplifies the implementation and maintenance of the LMS and gives web service consumers more choice in finding the services they require [19].

Moodle, as well as similar VLEs, is designed to include the principal aspects of the constructivist learning theory [27]. In particular it offers the possibility to visualize (with animation), and to manipulate interactively, educational contents or metaphors of learning objects. The constructivist approach is based on the learning-by-doing approach, which emphasizes the active role of the student in building his/her knowledge [8, 46, 77]. The active dimension of learning is realized by means of virtual laboratories [12] that allow students to visualize (with animation) and manipulate interactively, step by step, metaphoric representations of the functions, modules and coupling paradigms for a deeper understanding of them.

The VLE foresees the development of a new generation of educational tools, for example: 3D architecture of circuits, immersive virtual environment, intelligent agents, avatars, and so on [22, 37, 70]. These new educational tools, offer a computer-based approach for scientific instruction that provides a number of advantages over traditional learning methodologies [49]. Students are stimulated by manipulating objects that offer interactivity, authentic experiences, and a new adventure in learning [54]. Therefore, our goal was to design an eLearning system based on experimentation activity (e.g. virtual laboratory) and the scientific method (e.g. simulation program write in Java and Java 3D language).

As each didactical context, students encounter different problems that are completed by using the tools of the environment and the scientific method to solve problems. In this way, the eLearning platform assures the maximum flexibility to the learner, whose results are assessed in terms of performance on specific tasks. Results of different studies have demonstrated a positive correlation between student motivation to learn and classroom integration of technology [11]. In addition, recent researches indicate that the use of technology in the classroom not only increases the student's motivation, but also improves achievement [11, 55].

The CoMSON eLearning platform provides three kinds of learning resources. First, a repository of lecture notes, slide presentations, articles, book chapters, etc. Second, it hosts interactive courses that can be used as a stand-alone learning solution or blended with face-to-face lectures or seminars. Third, a simulation platform that interfaces with the DP to provide educational simulations. This latter section of the eLearning platform at this moment is not fully functional.

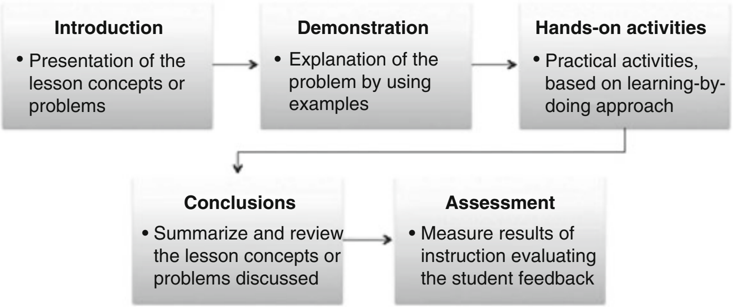

Introduction
• Presentation of the lesson concepts or problems
Demonstration
• Explanation of the problem by using examples
Hands-on activities
• Practical activities, based on learning-by-doing approach
Conclusions
• Summarize and review the lesson concepts or problems discussed
Assessment
• Measure results of instruction evaluating the student feedback

Fig. 9.10 The CoMSON eLearning conceptual model

9.4.2.1 Conceptual Aspects of the CoMSON eLearning

The CoMSON eLearning is based on constructivist methodologies, a set of assumptions about learning that guide many educational theories and associated teaching methods [44]. Constructivism learning approach guides learners to conduct and manage their personalized learning activities, and encourage collaborative and cooperative learning to improve critical thinking and problem-solving strategies. The knowledge is constructed actively through the interaction with the environment. In fact, the constructivist paradigm asserts that learning environments should support multiple perspectives or interpretations of reality, knowledge construction, context-rich, and experience-based activities [77]. In our eLearning platform, learners and instructors can interact with different technologies, which support the students in the acquisition of skill on specific topics [18, 47].

According to Horton and Horton [32], an electronic curriculum is composed of individual courses, books, and other learning objects. Courses are typically composed of clusters of smaller lessons, organized to accomplish one of the major objectives of the course as a whole. At a lower level are the individual pages, each designed to accomplish a single low-level objective that answers a single question. Such units may also be called screens in multimedia presentations or topics in online help. At the bottom level are media components. These are pictures, texts, animations, and videos that contribute to design the page content.

Figure 9.10 shows the CoMSON eLearning conceptual model adopted to realize the learning paths [68]. The CoMSON eLearning conceptual model includes five sections or steps.

In the first step, “Introduction”, the platform introduces the educational aim of the lesson, such as procedures, principles, concepts that will be discussed. The second step is “Demonstration”. In this section the platform explains with more details, by using as example results from scientific experiments, the concept introduced previously. Next, with the purpose of improving the assimilation of new concepts, the platform guides the learner through hands-on activities with the support of virtual laboratory or simulation tools. These activities are based on constructivist approaches, emphasizing an active engagement of learners. For example, connecting this section with the Demonstrator Platform (DP) through a Graphical User Interface (GUI) so that the learners can perform experiments verifying concepts or testing hypothesis. Then, the “Conclusions” section summarizes and reviews the theoretical and practical concepts discussed during the lesson. Finally, the “Assessment” section concludes the lesson. This module includes a synthesis of the main concepts discussed during the educational activities.

The integration of different tools allows the application of innovative eLearning methods and technologies based on the following aspects:

- Definition and development of educational paths for all researchers, including internal training: using the information system, a web-supported documentation and Transfer of Knowledge (ToK).
- Adaptation of the DP to training and educational needs: using suitable GUIs which highlight coupling paradigms, important modelling issues, algorithmic issues and all other issues analyzed in the training and educational paths.
- Creation of a virtual educational system, which transfers traditional classrooms to an electronic environment based on: remote access for all system users, direct interaction between students and lecturers/tutors, and support to communication among students and teacher.
- A continuing education environment supplying information about the materials and some general documentation of the platform: annual progress reports on the project, software, online lectures, and communication tools.

The use of these approaches is supported by a full integration between virtual tools and remote simulation by the DP environment. The full integration between the eLearning platform and the simulation environment is a challenging technological problem, which has not been fully solved during the project CoMSON. More details on this topic will be given in the following section.

9.5 Graphical User Interfaces

In the previous sections we have introduced the eLearning platform analyzing it from a technological point of view. The visual interface is another important component of an educational platform.

The Graphical User Interface or, as it is commonly called, GUI is a crucial part of a users experience with any computer system [69]. Why? It is the system to most users. It can be seen, it can be heard, and it can be touched. The piles of software code are invisible, hidden behind screens, keyboards, and the mouse. Each user interface has essentially two components: input and output systems. Input concerning how a person communicates his or her needs or desires to the computer system. Some common inputs devices are the keyboard, mouse, and so on. While, the output is how the computer conveys the results of its computation process and requirement to the users. Today the most common computer output mechanism is the display screen and other systems that support the subject during the interaction with the system.

User interface design is a subset of a field of study called Human-Computer Interaction (HCI). HCI [15, 16, 25] is the study, planning, and design of how people and computers work together so that the person needs are satisfied in the most effective way. HCI designers must consider a variety of factors: what people want and expect, what physical limitations and abilities people possess, how their perceptual and information processing systems work, and what people find enjoyable and attractive. Designers must also consider technical characteristics and limitations of the computer hardware and software.

The goals of interface design are simple: to make working with a computer easy, productive, and enjoyable, reducing the cognitive load during the interaction. In the last years we have assisted to an improvement to the design and implementation of the GUI [69]. The new generation of GUI includes a variety of new display and interaction techniques that improve the dialogue among subjects and system.

9.5.1 User Interfaces in the eLearning Platform

In the eLearning environment, GUI should allow the interaction between user and educational contents in an easy way with the purpose to improve the learning [60]. Clearly, not every student learns in the same way and not every curriculum should be presented in the same manner. Students are different in their learning cognitive styles, and different disciplines and contents require different presentations modalities. An eLearning system often provides dynamic and adaptive environments which allow to personalize educational materials both in terms of students learning styles and type of contents to deliver. De facto, the best interface will permit the user to focus on the information and task at hand instead of using complex interaction mechanisms that impede the communication process and involve a strong cognitive load reducing the cognitive resources.

It is known that the eLearning interface design is especially complex, as the learning effectiveness and interface design are substantially intertwined. In addition, a trend to reduce the complexity of the interface interaction is to apply the usability approaches to evaluate the quality of the system interface [57].

Usability measures how intuitive, efficient, and pleasurable the experience of using an interface application is, as well as how effective the application is in achieving a user's end goals [57]. The usability of an eLearning system refers to how easy it is to use and learn the system. In online learning system contexts, the pedagogic usability is also related to how easy and effective it is for a student to learn something using the system.

For all practical purposes, the GUI of the CoMSON eLearning platform is basically the interface provided by its Course Management System, which is Moodle.

9.5.2 A Graphical Tool to Visualize Scientific Data in the Simulation Platform

The main goal of the Demonstrator Platform (DP) is to train new recruits in the field of microelectronics. To do this, a series of modules have been created each one tackling a different problem. These modules, created by the researchers in CoMSON, provide tutorials that explain how they work and to allow the visualization of results.

These tutorials use OpenDX to visualize data obtained from DP simulation. OpenDx (Open Visualization Data Explorer) is a scientific visualization software developed by IBM [35]. This software can operate in complex domains along with measured or computed data. The OpenDX project started in 1991 and can do 3D visualizations that represent the output values as color or gray scale coded, or as vectors streamlines and ribbons. It also offers the advantage that the graphs can be viewed form the inside or make cuts and represent the data in the cutting plane. The graphs can be rotated and visualized from any angle and animations of these movements are produced.

OpenDX provides a simple toolkit that allows the user to manipulate images and modify different aspects of the visualization. Through a window menu the user selects a series of blocks that perform actions to visualize data. To visualize the results of the development platform modules using OpenDX the requisites are:

1. CoMSON DP installed.
2. OpenDX installed.
3. BIM-MSH-FPL packages loaded.

Once these programs have been installed the user can call the packages and be able to visualize the results. The user has to follow these steps:

1. Move to the example directory.
2. Start octave.
3. Prompt run_test at the command line.
4. Exit octave.
5. Use OpenDX to visualize data.
6. Repeat the procedure changing equation parameters.

This allows the user to visualize the results produced by his/her code. Then the user uses the interface provided by OpenDX to select the modules that the data is going to be filtered through. Finally running the data through OpenDX the user can visualize the solution.

The user then can change the parameters that he/she is using, to see how the result changes. It is this exploration of the problem through visualization that allows the user to learn and master the topic.

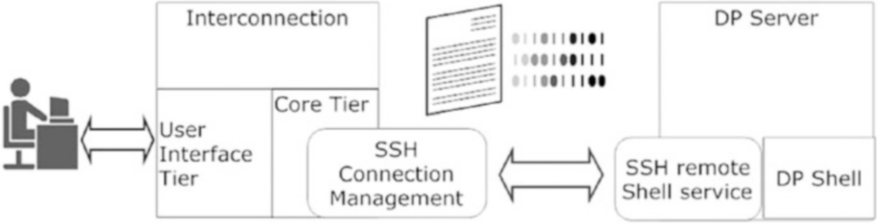


Fig. 9.11 Interconnection architecture

9.5.3 Interfacing the Components of the CoMSON Platform

Another task of the eLearning research group was to design and develop a GUI to connect the eLearning platform with the Demonstrator Platform (DP). According to the design guidelines, an application should be inviting to use. It should contemplate all the information and tools necessary to the user to complete tasks quickly, and it should guide them with an appropriate feedback. To apply with success the design principles, one needs to understand the user requirements and tasks. To understand how a final user might interact with a visual interface, it is useful to formulate a simple functional model of the functions. Figure 9.11 provides a visual representation of the components that make up the GUI system and the services that will be possible to activate. This is not an architecture model of the system, but rather a conceptual model that we can use to realize the GUI product and their functionality connected with the eLearning platform.

This interface will be realized in Java language and will allow the users to perform test and simulation realizing electronic circuits. To obtain this results, we design a prototype architecture of the interconnection based on two-tier, namely client/server architectures in which the user interface runs on the client and the database is stored on the server. A first, core tier is used to transfer inputs to the DP and to collect outputs from the DP; a second tier is used as a user interface layer (input entry, output presentation) and communicates exclusively with the core tier. The DP is basically a shell environment that users can access remotely via SSH service provided by the host machine. A simple approach to designing a core tier is to define a system that establishes and manages a SSH connection to a remote or local DP, and exchanges commands (inputs) and outputs using the DP shell.

Based on such core general design, the GUI tiers will be designed in order to support more or less complex user interactions and visual representations on the basis of user needs and suitability for specific learning objectives. The approach used to implement the interface architecture is shown in Fig. 9.11. This remote interconnection architecture provides the functionalities that let the user complete use the input and output capability of the DP to take full advantage of the DP's computation environment. While the core tier is concerned with exchanging flows of information (inputs and outputs) with the DP, the user interface tier will have the task of translating specific user interactions into such flows. In the Sect. 9.5.2 of this chapter we show how the DP output is visualized on the screen. The interconnection architecture shown in Fig. 9.11 raises several technical issues. One issue is the availability of open source components that implement for example the SSH layer, the management of graphical widgets and so on. Another issue is the integration into standard web browsers, commonly used as client applications to access a Learning Management Systems.

The technical problems posed by the above needs and by the consequent design, have not been fully solved. Nevertheless, during the project CoMSON many attempts have been made to build some effective 3D GUI prototypes. In a first prototypal scenario, testing an industrial case study [9], the GUI would offer three working environments: the model sculptor, the algorithm sculptor, and the model inspector. The first two environments are 3D authoring tools for, respectively, designing and manipulating mathematical models (equations) representing the devices and for designing and manipulating algorithms providing numerical solutions to those equations. The third environment is a tool for inspecting the value of variables in the model during simulation time. The manipulation and exploration of models and algorithms provided by the proposed GUI might be useful in contexts where learning by exploration and design by exploration are common approaches.

Another direction of research for possible design solutions for GUIs between different components of the CoMSON platform aimed at exploiting the advantages and the potentiality of the third dimension. It has been suggested that interfaces based on this concept will allow to design new virtual environments that include more interactive functionality [9, 68]. This line of research was not deeply developed during the project, but it was possible to realize, as a proof of concept, a 3D virtual environment [14, 39] that includes an avatar which can be controlled by the final users. In this virtual environment each room has a theme related to microelectronics, where the user can find different educational objects such as images, interactive movies and so on. The user is free to move his/her avatar exploring the environment and moving from one room to the next. The virtual environment is provided with 3D agents that can be used to gather information on request, to get suggestions on exploration paths and to have support in accessing other services [29].

In this experimental interface, the users are immersed in a virtual context which is populated by other users and virtual agents' avatars acting as tutors and guides. An example of avatar is shown in Fig. 9.12.

Avatars act as cognitive support for the students that use an eLearning system. The support is crucial because it stimulates human interaction among students, especially in the autonomy model. Usually, the support consists of personalized help for each student as he/she encounters an issue in problem solving tasks, and should be contrasted with the traditional educational activities, in which one teacher delivers educational contents for many students. Traditionally, the avatar shows adaptive behavior to increase the comprehension of each student from a cognitive point of view.

In this scenario, the eLearning system is based on the hypothesis that the manipulation activity improves the learning [34]. Therefore, our goal is to design

Fig. 9.12 Example of avatar

and develop an eLearning system based on experimentation activity (e.g. virtual laboratory) and the scientific method (e.g. simulation program written in Java and Java 3D language). A virtual laboratory might show some examples of the microelectronic technologies. It should be furnished of interactive animations and pictures that allow the user to interact with different learning materials. Users will be able to study different processes alive. In some virtual laboratory rooms, students can change the parameters of the objects and see how these will affect the final result. Also, animations and other educational materials are supplied with specific descriptions.

The use of virtual environments in eLearning is one of the most promising applications because it allows the subjects to interact with virtual objects, improving conceptual and practical abilities [13]. Learning through experimentation is an important strategy because it supports students during problem-solving activities. An active and collaborative learning environment provides a powerful mechanism to enhance depth of learning, increase conceptual retention, and get students involved with the material instead of passively listening to a lecture [30]. For this reason, a virtual environment should be based on five categories which included the following aspects:

1. To work on real-world problems into the virtual environment.
2. To provide the students with scaffolds and tools to enhance learning, in the virtual environment.
3. To give more opportunities for students and instructors to share ideas and to collaborate using technological tools, working on common projects.
4. To build virtual educational communities to expand learning opportunities in the microelectronics field.
5. Integration between collaboration, sharing tools and simulation environments embedded into the eLearning platform.

Each category poses an opportunity for technology integration, and a successful integration increases both the technological skills and the content knowledge.

From a methodological point of view, the informative system integrates a multidisciplinary approach, in order to share both educational contents and tools. Academic partners and industrial companies participate to implement contents and tools to integrate within system. The main goal of the platform architecture is to improve the integration between documents and tools for a better usage and collaboration by the consortium members and its future as a learning tool. This integration facilitates the collaboration and improves the learning.

9.6 eLearning Contents Creation: Methods and Strategies

eLearning scientific education is often difficult to sustain. Educational content creation is often time consuming because both technological infrastructures are not always user friendly and in some cases teachers needs to rewrite educational contents adapting them to the eLearning platform. According to Minato et al. [51], the eLearning content creation shows many problems:

- Contents are often insufficient or inappropriate.
- Content creation takes much time and energy.
- Quality production entails significantly on financial cost.
- To obtain an effective educational content it is necessary to perform many revisions.

New ways of teaching and learning are made possible by a variety of new technological applications, on-line resources and virtual environments, as well as by new didactical approaches to deliver educational contents, based on problem solving strategies. Today these changes are not only more evident, with the enormous increase of ICT in use, but also even more significant because of the new advanced modalities it is possible to carry out.

All these new technological tools used in educational context require that teachers acquire new teaching methods for the new generation of the students, who have grown up with new technologies. Moreover, teachers need to acquire conceptual and practical skills to create educational contents to be delivered by using an eLearning platform. In most cases, reviewing a course and responding to current needs is perhaps done intuitively and without a formal procedure.

Nowadays, the educational community is well aware of the importance of updating curricula and methodologies in response to the changing requirements of the information society. Developing a new course or changing an existing teaching approach is likely to feel discouraging, time-consuming and risky, especially when technology is involved. These risks and concerns can be significantly diminished if a more explicit approach is taken to evaluating needs.

9.6.1 General Methods and Strategies for the Development of an eLearning Course

Notoriously, eLearning contents creation is a difficult task, because it concerns different aspects such as conceptual (learning approaches) and operational ones (technology and infrastructure). On the other hand, it is important to consider other relevant aspects, such as planning, design and evaluation process. In this framework it is important take into account the cognitive processes of the user that attends an eLearning course. The choice of a conceptual learning model is expected to influence the design of the eLearning environment and then the learning process of the students. The conceptual aspect involves abilities to organize the didactical contents in an easy way and to apply educational strategy to design the learning paths. Hence, many learning approaches have been defined, which suggest how to organize the lesson contents integrating theoretical and empirical aspects.

The main idea of the educational approaches is to create a virtual environment where learners can share knowledge and are engaged in a communication process that makes the learning process more active. The operational approach concerns abilities to use software to create educational contents such as animations, simulations, graphical images, movies, online assessment, useful to improve the quality of the learning path making it more attractive. Besides, a teacher should know how an eLearning platform works, to adapt the educational contents to the features of the platform. An eLearning system requires specific competences and an interdisciplinary team able to support the teacher.

An eLearning application represents an intersection among contents, and design, learning and cognitive strategies. More specifically, developing an eLearning course that successfully delivers educational contents requires the joining of many different skills: technical, psychological, pedagogical and computer graphical skills. All these aspects represent the core of an eLearning application. It is possible to summarize all these needs by taking into account the following criteria:

1. Plan the eLearning project. This is a preliminary step in which the available resources and other aspects to realize the eLearning contents, are evaluated.

   (a) Estimate the economical and human resources needed to realize the educational system and its contents.
   (b) Define the criteria to analyze the user cognitive profile. Knowing the final user profile will allow to organize appropriate educational contents.
   (c) Create a project plan of the eLearning paths needed to deliver specific educational contents. This is the final phase of the first step and concerns the organization of all the activities.

2. Choose the eLearning platform. Many eLearning platform today have been developed. For example, Moodle is the principal platform used in academic context.

   (a) Design the layout of the system. In many cases, the design needs to be improved creating new functions.
   (b) Improve the interaction quality of the system. It is possible to use the standard applications of the system or to design new functions that improve the quality of the learning.
   (c) Create a visual theme. This aspect concerns the quality of the graphical layout of the system.

3. Develop educational contents. After this initial planning phase it is possible to start with the next step that involves the creation of eLearning contents.

   (a) Design the learning paths. Each learning path should cover a course's contents, including the main topics and subtopics, pre-tests or practice sessions, overviews, quizzes, and summaries.
   (b) Develop the Learning Objects (LO). A LO is a piece of knowledge that include all parts of the educational process: lesson, assignment, evaluation and so on.
   (c) Choose an instructional approach. There exist several approaches to deliver educational contents. It is possible, to combine text and other media elements in order to attract the user's attention.

4. System evaluation. This final step concerns the usability evaluation of the eLearning platform and then of the learning objects.

   (a) Usability of the eLearning platform. Evaluate the quality of the interaction with the interface of the system.
   (b) Test the quality of the developed eLearning objects. Test the quality of the educational contents before delivering them.

9.6.2 Some Examples of Course Implementation

Creating eLearning materials is a complex task with attention to delivery effective material as well as providing learning path to encourage future study of the audience. According to requirements and existing background of the students, a course should provide flexibility and exploit full potential of the learners. Within CoMSON, the Bucharest node implemented an eLearning course on optimization [73], which is one of the important aspects of the project and collaboration scheme between project's nodes. Thus, the connection between optimization and eLearning modules was strengthened with the deployment of a code on the Demonstrator platform, involving actions from Bucarest (Technical University), Catania (STMicroelectronics), Calabria (University) and other nodes [42]. Materials were presented in a form of a Moodle course, and the programming source code was made available as Octave and Scilab implementation. Widely available and open software to solve these tasks were primarily used.

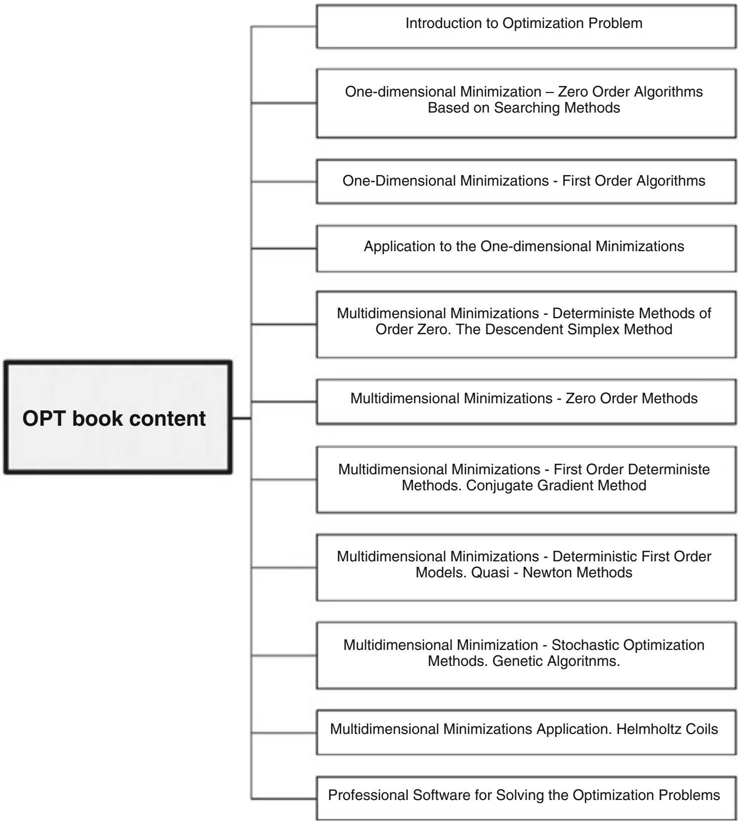

OPT book content
Introduction to Optimization Problem
One-dimensional Minimization – Zero Order Algorithms Based on Searching Methods
One-Dimensional Minimizations - First Order Algorithms
Application to the One-dimensional Minimizations
Multidimensional Minimizations - Deterministe Methods of Order Zero. The Descendent Simplex Method
Multidimensional Minimizations - Zero Order Methods
Multidimensional Minimizations - First Order Deterministe Methods. Conjugate Gradient Method
Multidimensional Minimizations - Deterministic First Order Models. Quasi - Newton Methods
Multidimensional Minimization - Stochastic Optimization Methods. Genetic Algoritnms.
Multidimensional Minimizations Application. Helmholtz Coils
Professional Software for Solving the Optimization Problems

Fig. 9.13 Numerical optimizations handbook

Numerical optimization techniques is an advanced effective module for undergraduate and postgraduate students at the University Politehnica of Bucharest. The optimization course's purpose is to present the fundamental concepts and main numerical optimization methods used in scientific computing and the computer aided design of electromagnetic devices. The courseware is developed in two languages – local language of academic partner node and English. The English version is a translation of the original Romanian book [20]. The book was widely used locally especially by the final year students from the Computer Aided Electrical Engineering Department.

Taking into consideration the diversity of the optimization problem encountered, algorithms and computing programs, it is difficult to initially find existing solvers that are optimized and efficient for a particular real problem. Usually, for solving a real problem, an appropriate baseline algorithm, as close as possible to the encountered problem must first be selected. The offered course presents not only the theory, but also practical applications and a pseudocode for test problems and models from the main approaches used upon which a more refined solution can be developed (Fig. 9.13).

With the creation of online contents on CoMSON DP, students now have easy interactive access to the course. When used in conjunction with another materials on Bucarest's ROMI (Reduced Order Modeling Interactive) on the CoMSON DP, they can exploit the algorithms and write the codes in order to gain a deeper understanding of the theory, the methods, their advantages and drawbacks. The theoretical presentations of the optimization methods at the beginning of each chapter prepare students and give them better understanding of the methods algorithm. For solving recommended tutorial problems, they can also use code sources available in Octave language on DP and expand upon them. Exemplary codes can be archived and showcased for future use.

Materials are arranged to develop higher-order thinking skills in students and successfully meet the cognitive domain educational objectives outlined in Bloom's taxonomy of educational objectives. The presented pseudocode for a number of optimization routines and the implementation in high level GNU language Octave and in Scilab, encourage students to experiment with the code for a better understanding, analysis, synthesis and evaluation of the available solutions [74]. By means of different representation forms and possibilities of interactively exploiting the code, better understanding and training results can be achieved.

In addition to the classical optimization procedures, mostly oriented for graduate students, the course includes genetic algorithms (Particle Swarm Optimization and Intelligent Particle Swarm Optimization) code. Moreover, industrial CRS (Controlled Random Search) global optimization algorithms, implemented by STMicroelectronics, was prepared and deployed on DP. Due to the purpose to support legacy and consistence of the code, it was presented in a form of program interface for Octave.

Besides the optimization handbook and the scientific optimization code, Bucharest node has developed a set of eLearning materials to support user in the creation of eLearning course. That set included eXe tutorial, LayoutEditor tutorial, professional communication course and information about technology supported learning and training.

As an addition to the professional training, many software solutions with descriptive materials were developed. For a better user's understanding and evaluation of the complex 3D forms, LayoutEditor was used. Using imbedded script-language, a subset for visualization was created (Fig. 9.14) in assistance to represent form of the circuit in the way for interactive exploration. For the purpose of further calculation a solution was created [26] to visually identify fundamental loops on the gds layout for extraction of the self and mutual reluctances using Finite Integral Technique (FIT) between the hooks of Manhattan shapes (union of rectangles) for further calculation in mathematic software (Fig. 9.15).

In practical applications students are expected to have not only information, but skills with existing mathematical applications, such as meshing strategies [41]. Therefore, the created materials were oriented not only to deliver knowledge to students, but also to offer them solutions for real task and develop knowledge in different areas.

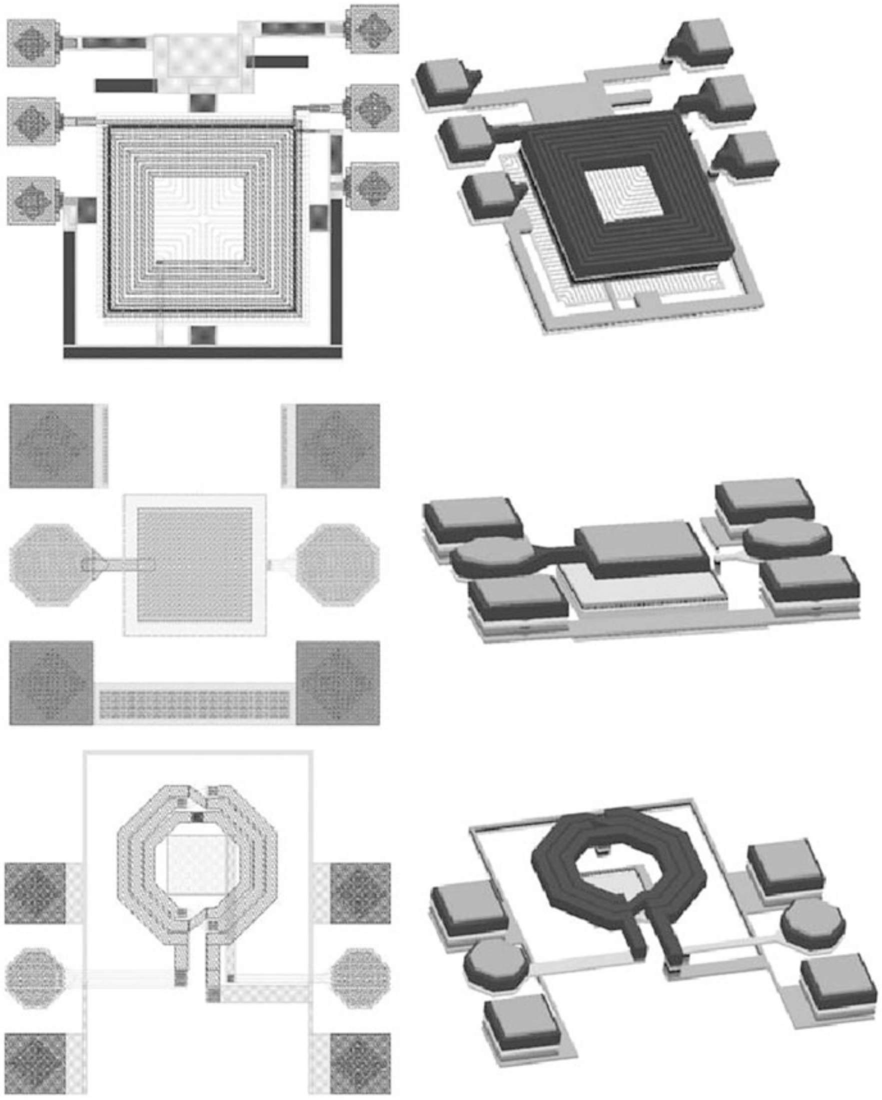


Fig. 9.14 Visualization of the layout

9.6.3 Practical Strategies for eLearning Contents Creation

Several factors affect the success of an eLearning course. In practical terms, we can reduce them to: time, money, competence, and technological infrastructures. These are the main aspects that enable to achieve both the didactical and the learning aims.

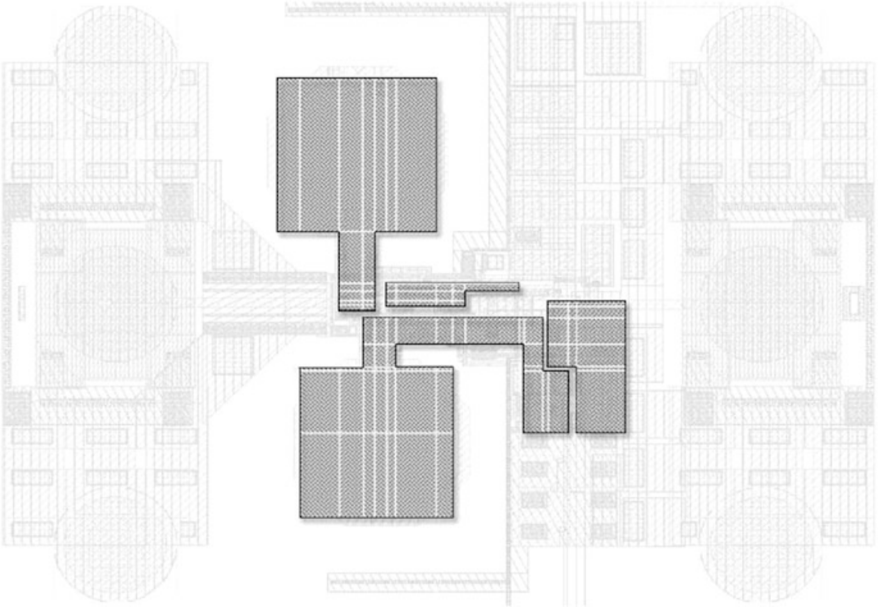


Fig. 9.15 Visual loop identification on the layout

The task of the eLearning research group was to realize an eLearning system for micro and nanolectronics, including the design of a learning path to manage educational documents and a Content Management System (CMS) to store, update and retrieve the educational materials. Unfortunately, the CoMSON project did not have specific funds devoted to design and implement educational contents for microelectronics, so it was not possible to adopt all the steps of the methodological approach mentioned before. For this reason, initially we analyzed different approaches to design, implement and deliver eLearning contents. Then, the real problem was to translate this preparatory study in action, that is, creating eLearning contents in microelectronics.

Since CoMSON project was devoted to training the researchers, its participants were mainly specialized in the microlectronics topics. In some cases, participants also held university courses, and some traditional educational material was available, in other case only some research papers were available. Thus, we made an attempt to translate traditional educational materials in eLearning contents, or to create new materials based on research papers. The idea was to ride out the economic limitations, suggesting an easy way to produce eLearning contents, involving all partners of the project consortium.

This was done by distributing to all participants a document with some guidelines for the realization of educational material. The document, called "learning@CoMSON", introduces the main conceptual aspects and some practical strategies to create eLearning contents.

9.6.3.1 Learning@CoMSON

As for the conceptual aspects, they can be summarized by the following questions: What is a learning path? Why is it important? And how to create it? How to create a learning unit, and how to edit existing material to create learning units? These concepts are useful when creating a course, which is nothing but a hierarchically organized collection of learning units. The minimum required materials for the creation of a course are: a text file with the learning path, materials for the course (e.g. derived from teaching material), and materials for the assessment of the course (e.g. derived from homework and exams).

1. *What is an learning path?*

Learning Path, a methodology developed by Jim Williams and Steve Rosenbaum [76], is a practical approach to produce an effective sequence of training, practice, teaching, and experience to reach specific competence in a particular field. A learning path is a guide that describes the necessary steps that a student should take to learn a concept or a skill. This is very similar to the outline of the course. A learning path is created as a helping guide to create a course.

The learning path is composed by a series of Learning Objects (LO) that are independent educational modules interlinked with each other. These LO are organized from simple to complex so that, as the student masters the simpler tasks, he/she builds on that to learn more complex tasks. In order to make these learning objects reusable in other courses, they should be complete and coherent by themselves.

Each module should have the following elements: a goal, a description or explanation of the subject and assessment for the concept explained. When developing the learning path, we should have in mind that the student will take the course without an instructor so every single step in the learning path should be present. We cannot assume that the student is brilliant and will be able to solve it by herself. We should provide a complete and coherent path.

Learning path is a general concept, for example we can have a learning path for a Computer Science career, the learning path will involve the student in taking courses in Programming, Operative Systems, Databases, Discrete Math, etc. This will be the learning path for the student. But the to learn Programming the student will have the Programming learning path that will include learning Object Oriented Programming, Procedural Programming, and Scripting Languages. The Object Oriented Programming will be a learning path that will include learning what are objects, what is inheritance, what is encapsulation, what is recursion, etc.

2. *Why is it important?*

The creation of a learning path has several advantages:

1. The learning path helps to create the course and to devise a perfect and complete learning track step by step. In the learning path there are no empty spaces and little or none pre-knowledge is assumed.
2. The outline will help to collaborate with others in the compilation of the course. The learning path allows for collaboration.
3. The Learning path allows scheduling and division of tasks.
4. The learning path in the platform will be associated to an XML database that will allow users to reuse the same modules for different outcomes. Different courses for different outcomes can be created with the same materials.

3. *How to create a learning path*

To explain how to create a learning path, we give a practical example for a course in Semiconductor Modeling, commenting on its various steps. The comments are in parenthesis.
Title of the course: Modeling of semiconductor devices
Course category: Modeling
Steps of the learning path:
Step 1. Description of the course

- Introduction to the course.
  (Here, one should give a brief description of the course)
  Introduction to mathematical modelling of semiconductor devices, with a special emphasis on content on physical-mathematical aspects, perturbation analysis, numerical simulation which are more relevant in the applications in microelectronics industry.
- Goals of the course.
  (Here, one should give a wide description or justification of the course. It should answer the questions: what is the final outcome of the course? What is it useful for? What does the course prepare the student for? What will the student be able to do when she has mastered the material?)
  The students will have a general understanding of the most common models of semiconductor devices, of their mathematical content, and of the most common strategies of numerical solution.
- Objectives of the course.
  (Here is where to write the detailed outcomes and the detailed description on how the goals should be reached. This paragraph should answer the questions: how do we get to the goal? and what is that goal going to allow the students to do?)
  Students will acquire knowledge by working on a simple test case: a 1D diode, modelled by time

-dependent and steady-state drift-diffusion equations, solved numerically by using the Scharfetter-Gummel discretization for the currents and the Gummel map for the resolution of the resulting nonlinear system.

Step 2. Outline of the course
(When creating the outline of the course one should think how to evaluate or assess the learning of the student. These are the main questions to be addressed: How can you tell if the student has learned the concept? Why is the concept important? How can the concept be used in real life or in a practical setting? How does this concept relate to other concepts of the course? What previous knowledge is needed to understand the concept? During this stage one can identify examples to be added to explain the concept, as well as additional materials that can be linked. All this information can be added to the learning path and will be really helpful in the future implementation of the contents. It is useful in one thinks of the learning path as a dynamic document that you will improve progressively.)

Chapter 1. Notes on semiconductor physics
(one can decide how to implement your course – by chapters, by lectures, by sessions, etc.)

Section 1.1. Basic concepts
(in each section state one should state the concept that is going to be explained and the elements that are used to explain that concept. To explain concepts one can use text, images, animations, video, etc.)

Lecture 1.1.1
(one can add information of what will you use to create this part of the resource)

- Inverse lattice and Brillouin zones
  (e.g., here will be used the demo from website x, y or z)
- Lattice wavenumber pseudo-vector
  (e.g., for this concept it will be used the explanation of book X)
- Conduction and valence band in semiconductors
- Electrons and holes
  (e.g., the animated graphic A.gif or B.java or C.flv will clarify this concept)
- Semiclassical approximation
- Lattice vibrations and phonons

Section 1.2. Physics at equilibrium of semiconductors
(Continue with the same specifications to create the whole learning path)

Lecture 1.2.1

- Fermi-Dirac distribution and carriers number densities

Lecture 1.2.2

- Hypothesis of non-degeneracy and mass action law

Lecture 1.2.3

- Parabolic band approximation and temperature-dependence of intrinsic concentration

Lecture 1.2.4

- Partial equilibrium and total equilibrium

Lecture 1.2.5

- Intrinsic semiconductors at total equilibrium

Lecture 1.2.6

- Extrinsic semiconductors at total equilibrium and nonlinear Poisson equation

Lecture 1.2.7

- Boundary conditions for the nonlinear Poisson equation

This is just an extract of the complete learning path created for the course of Modelling of Semiconductor Devices (A.A. 2007–2008). The learning path is a dynamic concept and can be enriched and actualized.

4. *How to create a learning unit*

Learning units should be created according to the learning path. One can think of a learning unit as a single node that is a step in the learning path. To build a learning unit is possible to reuse existing teaching materials.

A learning unit should be a single and independent unit that is interlinked to others but that has a meaning by itself and conveys a well-identified concept. A learning unit should be a unit that states a goal, explains how to reach that goal and assesses that the goal has been reached. One can think of a learning unit as a section of some notes or of a scientific paper, with the addition of a short pre-description and some assessment questions, that is, what in notes would be called “homework” or “exercises”.

One should remember that the students of an eLearning platform will work without an instructor. For this reason the contents should be richer in examples and demonstrations to explain in a clear way every idea. When creating a learning unit it is important to identify where could practical examples or simulations be useful. Also, in the learning units one can add video, animated graphics, or links to additional materials, and use all these elements to make a clear explanation of the subject.

To have a well-rounded learning unit one can follow these steps:

1. Objectives and requisites.
    (a) Inform of the objectives. What will the student learn in this unit? Informing of the objectives helps the student to focus on the goal, avoiding distraction produced by other elements in the unit. These other elements are necessary for the explanation but are not crucial for the goal. If the student has a goal will pay more attention to the elements that help to achieve that goal.
    (b) Explain the requisites. What previous knowledge is needed? Students should know what are the prerequisites and links should be provided to additional information that the student may need to understand the unit.
2. Present your explanation.
    This is the main part of the unit, here you have to explain the concept. In this section you can use text, graphics, equations, video or interactive simulations. To explain the concept, use as many examples as needed to clarify the concept. Remember the student is alone. Provide examples to multiple situations where the concept may apply and provide links to additional information.
3. Assessment and reflection.
    (a) Introduce assessment. The assessment's main goal is to know if the concept has been understood; but it is also useful to present cases and exceptions. The questions should provide an insight of the concept and help to deepen the understanding of the concept. It is convenient to provide a multiple-choice question with commented answers. Each answer should have an explanation pointing out why it is correct or incorrect.
    (b) Reflection. A reflection or practical example shows the student a wider scope of how what has been explained in the unit works. This can be a practical situation where the concept applies, a real life example or other relation that the concept may have with other concepts.

5. *How to transform existing material into eLearning*

To create eLearning courses it is possible to reuse own Slides, Notes, Video and Simulations. Anyway, one should always keep in mind that the creation of an eLearning course is not the mere translation of previous material to the website format.

The courses implemented by the scientific content experts will be a series of modular units that guide the student through a learning path. Each unit should have the following elements: a goal, a description or explanation of the content, and a final assessment for the concept explained. When implementing these units one should have in mind that the student is going to take this course without any instructor's help. In the design of the units one should aim for the highest degree of interaction with the material, using lots of examples, graphics, visually dynamic content (videos, animated graphics, etc.). For this end the already existing materials will have to be modified. For example already existing presentations will have to be modified to be more interactive and include assessment. The final eLearning quality of our platform will depend on the clarity of the courses and how well they train the students.

In the following section, are described some recommendations to transform the already existing materials to create courses for the CoMSON e-Learning platform.

5.1. Presentations

Slide presentations can be useful to create a course for the CoMSON eLearning platform. The presentation should be improved to be a stand-alone course that will be used without an instructor. This means that the student won't be able to ask questions to an instructor. The material should provide more examples more interactivity and links to materials that may help the student to understand better the concept. You can reuse your slides but you have to modify them to provide this kind of environment (remember this is not a translation, is an enrichment process). Slide presentations that have embedded animations, simulations or video are encouraged and will work flawlessly in the platform. To create or modify an existing presentation follow these instructions. The first thing to do when creating an eLearning course is to identify the learning objects and create assessment to see if the concept has been understood. This is like writing a scientific paper. You have to ask yourself if your explanation is clear for an external individual to understand it unaided. Most of the presentations come in the form of power point or PDF. The presentations can be created or modified in two main ways:

Creating assessment.

Identify the concepts and create questions to see if they have been understood. To do this in power point, create a new slide with the following 1. Question. 2. Hint to solve the question. 3. Options to answer. Each option should come with a reason "why" that option is right or wrong. Create assessment for each concept or module of the learning path that is covered in the presentation. If you have a PDF presentation create a document with the assessment in the same way and specify where should be the assessment placed. This addition is mandatory (a presentation without questions will provide no interactivity and has little or no value for an eLearning platform).

Adding a voice-over.

To do this in power point, write in the comment box and our software will automatically read the comments of every slide. To do the same with a presentation in PDF create a document and create a comment for every slide. Number the comments with the number of the slide. The voice over is an important optional modification. Remember that if your presentation doesn't have comments will run silently in the server.

5.2. Text notes

Course notes and already exiting textual material can be easily modified to be used in the platform. Text based material in CoMSON to our knowledge comes from in PDF, LaTex, Word, or HTML formats. The first stage to modify your notes to create an eLearning course is to identify the learning objects and create an assessment to see if the concept has been understood. This learning units can be enriched using additional materials (animations, graphics, video, etc.). In the end the learning unit should be a well rounded unit that can be used independently. The textual material with the assessment and additional materials can be edited using eXe software.

9.6.3.2 Outcome

So far, we have realized some experimental eLearning lessons applying the guidelines summarized in the "learning@CoMSON" document. The lessons' aim was to verify both adequacy and efficiency of the educational material developed take into account the cognitive and didactical aspects. It is important highlight that the CoMSON eLearning platform will continue to work after the end project activities. We hope that the partners of the CoMSON consortium will apply both guidelines to create educational contents and the eLearning platform to deliver microelectronics course. These future applications will improve the quality of this initial prototype improving the quality of the didactical contents creation.

9.7 Blended Learning

After this initial phase, the next step was to experiment the didactical approach adopting the eLearning guidelines. Initially, we use the blended learning as didactical strategies. Blended learning is a term getting a lot play particularly in the corporate training course. Practically, it refers to the use of more than one learning medium, usually it includes a combination of teaching modalities supported by web-based tools. However, eLearning does not eliminate existing educational methods and technologies. Rather, it complements them by using new tools supporting learn cognitive abilities.

In the next two subsections, we describe the empirical studies carried out during the CoMSON project. The aim of this work was twofold. The first aim was to cerate eLearning educational contents as support to the traditional didactical activities. The second one was to identify new operational strategies concerning the eLearning content creation.

9.7.1 First Empirical Study

Due to the complexity of the subject of the CoMSON project and the scarcity of the content-experts time, the project has a wide gap between content creation and eLearning implementation.

To fill this gap we have tested a collaborative Project-Based Learning (PBL) approach. This approach has been tested in a specific course on "Modeling of semiconductor devices" at the Engineering Faculty (University of Calabria). The program of the course covered physics of semiconductor devices and mathematical methods of simulation. The program of the course included the following modules:

- Revision of semiconductor physics.
- Physics of equilibrium in semiconductors.
- Kinetical models of transport.
- Monte Carlo methods.
- Macroscopic models of transport.
- Numerical methods.

The activities of the course was divided in two sections, theoretical lessons and laboratory activities. The theoretical lessons comprised the aforementioned course topics. The laboratory activities explored physical phenomena using software simulations such as Matlab® and Octave®. For the examination session students had to develop a project, in the form of a learning module. In a first experimentation of this assessment procedure, the students had to tackle a single problem, related to numerical simulation of semiconductor devices, with the possibility to choose several variants. These modules would compose a sort of online handbook of the numerical method used for the simulation. In a second experimentation the students have been asked to present the main topics dealt in the course, creating independent learning modules. These modules should be able to be used as stand-alone learning contents. Upon the completion of the project the students had to undergo an oral examination where they presented their learning modules and were questioned about the problem tackled, the solution provided and the general pedagogical presentation produced.

Next, we describe in detail the first group of assessment projects. Overall, the projects had a central common topic, that is, the simulation of a one-dimension diode described using the drift-diffusion equations. The students' work was based on pre-existing Matlab codes which they had to modify or to replace with new programs to implement a simulation with the following variations:

- Fixed geometry/variable geometry.
- Uniform discretization/Non uniform discretization.
- Uniform mobility/field-dependent mobility.
- Generation-recombination without impact ionization /with impact ionization.
- Physical variables/nondimensional variables.
- Time independent simulation/time dependent simulation.
- One dimension simulation/two-dimensional simulation.

For the final presentation of the project the students had to create a stand-alone learning module, which described the problem they solved, the simulation algorithm used for the numerical solution, the interpretation of the numerical results. In this module the students used as a guide the following scheme:

- Inform about the goals of the learning activity.
- Explain what prerequisites are needed for the understanding of the module.

- Present the central explanation of the project.
- Introduce assessment (this assessment has to provide feedback).
- Finish with a reflection or a practical example that shows how does it works in real life or related to a bigger picture.

The learning modules were produced by using eXe®, an open source eLearning XHTML editor to create educational contents, compatible with the SCORM standard.

The final evaluation of the course consisted in an oral exam, where the students presented and discussed the learning modules created for the project, and were questioned by the teacher on the contents of the course. The evaluation of the project took 30 % of the final grade of the student. Prior to the oral exam, the students delivered to the teacher a copy of the project, which performed an initial evaluation of it. A the end of the evaluation process, the teacher approved or rejected the student's admission for the oral examination. If the project did not fulfill the required quality the student was asked to review it till the project could be approved for the oral exam.

The projects have been analyzed concerning the code and the learning module. In the analysis of the code produced by the students the projects present a modular structure where the students had explained the process as it is produced in the code. This structure provided evidence of the understanding of the code as well as the organizational capacity of the student. The students have adapted the code creating substantial modifications and new pieces of code to solve the specific problem. The results obtained are then compared with the theoretical predictions. Some of the students have produced a system of blocks that gives a scheme of the possible different combinations guiding the learner through all the possibilities of the software.

The students have produced a detailed analysis of the project describing each piece of code used in the solution. Also the different parts of the code have been thoroughly commented explaining the functioning of the code in a step by step manner. Some of the students have also produced innovative ways to interact with the software implementing graphical interfaces that can be used as a teaching and demonstration aid. Using this interface the software can easily produce a visual presentation of concrete examples.

The results of the analysis of the code are summarized in Table 9.1 and in Table 9.2 in percentage of students.

In the analysis of the project we have observed that the students provide new and innovative ways of solving the problems and presenting them. Collaboration among the students has created new ways of presenting explanations of the problems.

With an analysis of the projects we can see some advantages of the PBL approach. These included the consolidation of the competences, collaboration with peers and, improvement of communication and presentation skills. In sum, the ability to accomplish a project from beginning to end, producing the deliverable and the documentation for it, is very satisfactory.

Table 9.1 Analysis of the students' projects: technical aspects

Items	Good (%)	Medium (%)	Poor (%)
Has implemented new code	20	60	20
Has modified the existing code	40	60	0
Gives an explanation of the problem and inner works of the code	40	40	20
Comments the code to explain the process	20	60	20
Implements graphical interface	60	0	0
Recommends bibliography and produces help files	20	0	0

Table 9.2 Analysis of the students' projects: structural aspects

Items	Good (%)	Medium (%)	Poor (%)
Inform of the objectives	40	40	20
Explain the prerequisites	0	60	40
Present your explanation of the project	20	60	20
Introduce assessment and provide feedback	20	40	40
Finish with a reflection or a practical example that shows how what you explained works	0	0	0

From the analysis of the materials produced by the students, we conclude that even though the materials present high quality work they cannot be directly used in an eLearning course without some modification and edition. Some of them need minor editing while others need more work, such as modifications, corrections or amplification of the explanations. These deficiencies can be attributed to the lack of knowledge of engineering students of didactic and pedagogic approaches and techniques. We think that PBL approach could be a good solution to provide the students with knowledge on communication techniques needed later in their professional lives. In a second phase of this approach we will provide students with more detailed instructions and help to take full advantage of this approach in improving their communication skills. We think that giving more detailed instructions on how to create the educational method will bring better outcomes and will help students learn better about the topics and how to explain them.

9.7.2 Second Empirical Study

As mentioned above the mathematical content creation devoted to eLearning environments is complex not only because it is time consuming, but also need to manage media elements such as graphs, tables, links, and formula. In order to experiment new didactical strategies to optimize the eLearning content, we have tested a new approach involving the students to create educational contents. Student's task was to design and create educational materials to deliver by eLearning environments.

The pedagogical strategy was based on constructivism methods, which involve the student to design educational contents. This approach has been tested within the mathematical course at University of Calabria, Cosenza – Italy (Engineering Faculty). We have organized many students' groups that worked on different course topics. Before starting with the examination phase, students submitted the projects and then we analyzed them from a qualitative point of view. The results of this first experimentation were not very good. We found that students had difficulty in organizing the learning paths of the educational concepts. The teacher, after this evaluation phase, decided to do some revisions in order to improve the quality of the projects.

These educational limits showed the impossibility to create efficient eLearning courses. So, we decided to design a new experimentation, again with university students. To improve the eLearning content creation, we designed and provided to the students some guidelines concerning: pedagogical aspects about lesson organization; projects editing in order to improve the quality of content description, and finally we suggested to the students to reduce the MatLab application preferring the didactical aspects of the educational contents.

The eLearning course topic has been organized by didactic units. Each group chose one topic and then realized the didactic unit by using the eXeLearning editor to uniform the student's projects according to the eLearning platform standards. Below, we list the didactic units:

1. Physics of a semiconductor in equilibrium and non-linear Poisson equation.
2. Drift-diffusion model: the case stationary IV characteristic.
3. Drift-diffusion model: the case and time-dependent analysis of small signal.
4. Drift-diffusion model: the case of time-dependent and-effect impact ionization.
5. Model of drift diffusion: dependence on mobility model and models generation-recombination.
6. Models of energy-transport.
7. Semiclassical Boltzmann equation and Monte Carlo method.
8. Hydrodynamic models for semiconductors.
9. Drift diffusion model with quantum corrections.
10. Numerical methods: finite differences (box integration method).
11. Numerical methods: finite element.
12. Numerical methods: numerical solution of nonlinear algebraic equation (Newton's method with damping).
13. Scharfetter-Gummel method for the numerical solution of equations drift-diffusion.

An example of the project organization build with eXeLearning editor is showed:

- Home.
- Project description: objectives.
- Preliminary knowledge.
- In-depth examination.

- Core of the project (Contents).
- Observations and conclusions.
- Self-examination (by using different answer modalities: yes/no, true/false, multiple choices test, and so on).
- References.

The scheduled project activity is based on the following organization:

- Content. The educational contents must have realized by using an easy language allowing to the others people to understand the educational concepts.
- Didactical. Every project had to include the following didactical aspects: objectives, preliminary knowledge, topic and simulation, self-examination and references.
- Structural. It is need to use, when necessary, the following ramification: Topic, Section, and Unit.

By applying these detailed instructions, the project layout was better that previous. Besides, this organization makes the projects contents easily usable to create a distinct eLearning course.

The preliminary evaluation of the student's projects was satisfactory, but the projects still required a deeper analysis. The next step was to ask the other students to improve the existent didactical units adding other topics and contents. However, we needed to reduce the redundancies of the concepts because often many subjects used the same contents to implement different educational didactic unit.

9.8 Platform Evaluation: Test and Revision

We have designed a questionnaire with the aim to collect information on the use of the eLearning platform from the Experienced Researchers (ERs) and Early Stages Researchers (ESRs) which work in different node of the project consortium. In particular, we were interested to know the following aspects: what are the useful features available in the eLearning platform?, what kind of materials are more frequently used by final users? We believe that this information is essential to understand if the CoMSON eLearning platform support users needs, from a communication standpoint.

The questionnaire consists of 13 items with mixed answers modalities (nine question with closed answers; four questions with closed and open-end questions). Finally, at the end of the questionnaire we have asked to the users to indicate three negative and positive aspects of the CoMSON eLearning platform, respectively.

The sample was composed of seven people, six males and one female. All the users have been working in the CoMSON project from 1 to 4 years. All the questionnaires collected were analyzed. The results are the following: 71 % of the sample has created didactic contents for the CoMSON eLearning platform (see Fig. 9.16).

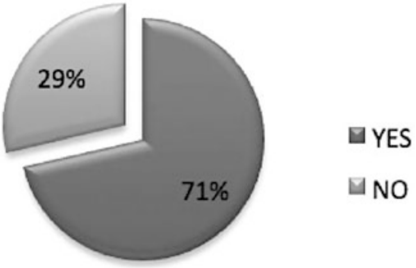


Fig. 9.16 Percentage of sample that has created didactic contents for the CoMSON eLearning platform

29 % of the sample used "Once per day" the CoMSON eLearning platform, another 29 % "Once per week" and 43 % of the sample "Rarely". None used "Several times per day" and "Several times per week" the eLearning platform (see Fig. 9.17).

We found that ERs people used the eLearning platform in different manners. 57 % of the sample used the CoMSON eLearning platform, for "Less than 1 hour", 14 % "Between 1 and 2 hours", while 43 % only "More than 2 hours" (see Fig. 9.18).

Both ERs and ESRs used the CoMSON platform for different aims. In particular, the 71 % of the sample used it "For educational purposes", while the 29 % used it both "For research purposes" and "For getting information" (see Fig. 9.19).

The CoMSON eLearning platform includes different functionalities, such as: Lectures; Courses; Virtual laboratories; Students homework; Communications with students; Students verification. We find that the more used functionalities are: "Lectures" from 57 % and "Courses" used from 71 % (see Fig. 9.20).

The majority of the final users (83 %) were satisfied of the eLearning materials delivered by the platform. In addition, all users agreed that the eLearning materials stored in the platform are interesting and engaging. 79 % of the users found that the educational materials are well organized and easy to understand.

As comes out from the literature on eLearning, it is very important to support the final users during their learning process. The 71 % of the sample found a sufficient help and support while using the CoMSON eLearning platform.

The eLearning platform is based on SCORM (Sharable Content Object Reference Model) standard, which allows the creation of standard contents that are exportable and executable on every SCORM compatible system. In the opinion of the 83 % of the users this eLearning technology works well.

Moreover, 67 % of the sample underlined that the CoMSON eLearning platform helped them to increase skills in topics related to the micro- and nano-electronics, even if the 33 % of the users did not agree with this statement. The same percentage 67 % asserted that the CoMSON eLearning platform supported them during the research activities.

Finally, both ER and ESR that worked in CoMSON and that used the eLearning system indicated some negative and positive aspects of the platform. Among the positive aspect, the sample mentioned: courses with tests are useful, because you can learn first, and, just after that, test what you have retained (14 %); it requires

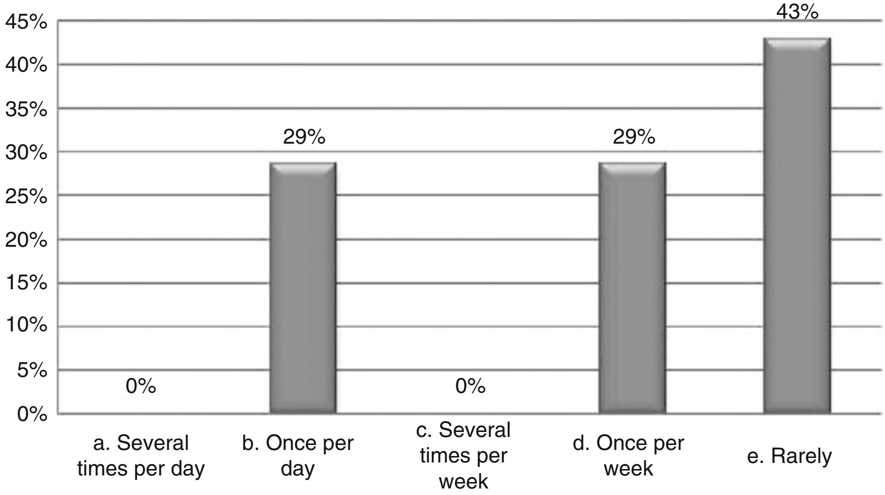


Fig. 9.17 How often CoMSON ERs and ESRs use the eLearning platform

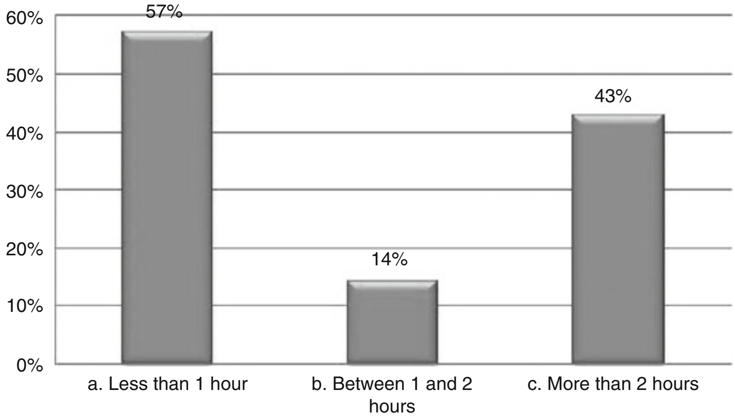


Fig. 9.18 How long CoMSON ERs and ESRs remain connected to the eLearning platform

a distributed collaboration (14 %); the material and the way it is described is also nice and new (14 %); it is clear and well organized (42 %); the eLearning platform is simple; the course on optimization is open to public (28 %).

Among the negative aspect, the following things were pointed out: there are very often technical problems with the server (42 %); there is the need of more content and to be actively used (14 %); the platform is not well supported (14 %); it is not well advertised (14 %); small amount of information (14 %); no recent updates (14 %).

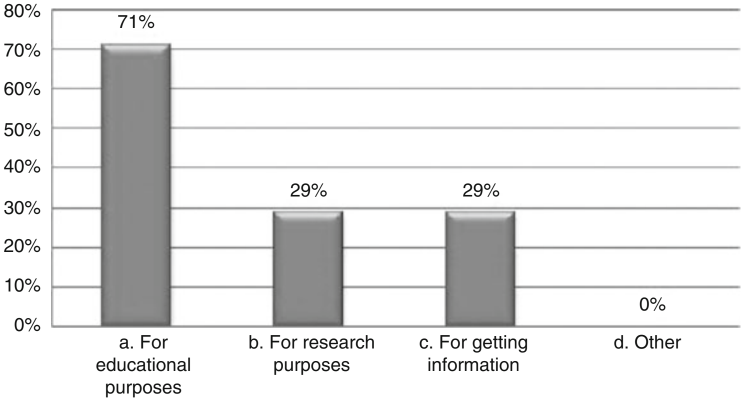


Fig. 9.19 Why CoMSON ERs and ESRs use the eLearning platform

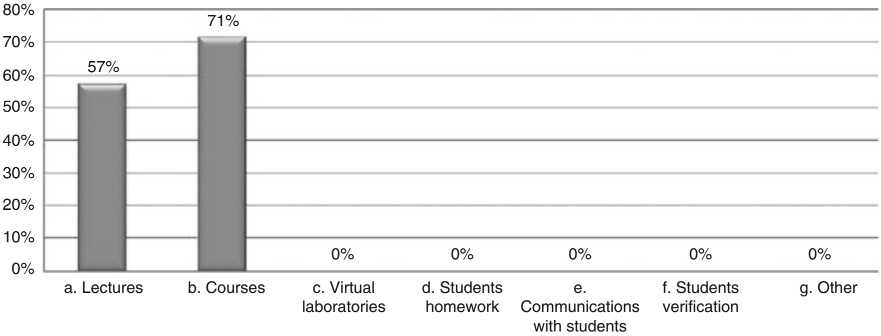


Fig. 9.20 Functionalities of the CoMSON eLearning platform more frequently used

9.9 Conclusions and Future Perspectives

The chapter aims to provide a straightforward introduction to the creation and use of eLearning approaches to support and teaching both university students and industry people that work in microelectronics field. It is aimed to introduce eLearning systems to enhance teaching, and it does not assume a high level of technical knowledge to use these tools. Although this chapter introduce together pedagogical-psychological, practical and technological aspects of the eLearning, it does not assume that many people will become an expert in the theories of learning, or a person with a high level of technical expertise.

Contrarily, this chapter wants to show how to use the techniques suggested (theoretically and practically) within a specific sector as the microelectronic. Both technological and psychological aspects, underpins the potentiality of the eLearning environment, suggesting activities that enhance teaching, learning and student assessment. In our opinion, CoMSON project has represented a scientific opportunity both students and researchers that work in the microelectronics field to experiment limits and potentiality of the eLearning.

The educational aims of this chapter are based on constructivist learning theory. According to the constructivist approach, students learn actively by doing things, rather than simply reading or being told about them, and construct their own conceptions of what they are learning. By using this approach, subjects want to test their own hypothesis with other people, thus redefining their understanding. The active construction of knowledge is thus better realized by virtual laboratories that allow to visualize (with animation) and manipulating interactively, step by step, metaphoric representations of the functions, modules and coupling paradigms for a deeper understanding of them. The CoMSON platform supports both online content creation and running of learning trails and allows a great flexibility in online material uploading. A series of tools are present in it, that are easy to use both for students and trainers, and allow a flexible and personalized acquisition of competences.

In our opinion, it is important to think about eLearning methodology as another educational innovation or modification of the traditional course design, and consider how it could be used to improve specific conceptual and practical abilities. This latter aspect is important for industry companies, which have the needs to experiment new educational strategies to ensure in short time the design of educational materials to train the internal people to use specific tools.

Of course, there exist different models of eLearning and procedures that explain how to integrate this approach into a traditional course for different circumstances. The assessment of eLearning course is another key issue, and it is important to ensure the validity of the chosen assessment. The introduction of eLearning is a good opportunity to think to find new strategies and approaches to evaluate the contents of a course. In this context, evaluation is important to ensure that the educational practice is effective to improve the way teaching, learning and assessment are carried out.

There is some experimental result that eLearning environments, and in particular the constructivist approach, increase the student's motivation and can be considered an effective strategy to enhance learning, improving not only conceptual skills but also practical competence to use tools (e.g. simulation environment and other system that stimulate students to interact). In the developing processes of an eLearning systems it is really important to adopt an interdisciplinary perspective, taking into account that different competences converge in this process: psychological (student's cognitive style, learning theories, cognitive strategies, user profile); pedagogical (objectives, contents, organization, methodology and didactic strategies), technological (technological resources, hardware and software solutions), user interface design (to foster the interaction between human and machine), usability (to evaluate the user interaction with the eLearning environments).

Considering the relationship between eLearning and microelectronics education, there are a number of possible future research directions. In particular, at the end of this chapter we mention the possibility to design and implement an mLearning platform devoted to microelectronics educational activities. Mobile learning (mLearning) refers to the use of mobile and handheld information technology devices in teaching and learning. The idea is to create an educational platform able to deliver electronic courses whose contents will be accessible through mobile devices such as a Personal Digital Assistant (PDA).

It is important to note that the most effective electronic educational environments stress the collaboration with others, allowing students to work together, learn from each other, and test their understandings. These aspects are in line with the constructivist paradigm discussed concerning the eLearning environment. It is possible to consider mLearning as an extension of the eLearning technologies and approaches. In the microelectronics field, mLearning could have many advantages for students, allowing them to carry out their learning activity from any location, any time in a connected environment using a personal PDA. Most likely, PDA could be used to deliver accessible engineering education to traditional and non-traditional students in blended or online way.

Students using mLearning technology can share documents and other educational materials with other people, enriching the learning process. This collaborative learning allows the creation and sharing of documents among students and teacher-authored resources. Resources can be hosted and linked to relevant websites in order to enhance the diffusion between students. Finally, we raise the issue that to use mobile devices as educational tools it is necessary to improve data transmission; to create a set of technology to unify mobile devices; to improve the quality of the display screens; to have more memory space to store data, and so on.

References

1. Aggarwal, A.K.: Web-Based Education: Learning from Experience. IRM Press, Hershey (2003)
2. Ala-Mutka, K.: Social computing: study on the use and impacts of collaborative content. IPTS Exploratory Research on Social Computing. JRC Scientific and Technical Reports (2008)
3. Alexander, S.: E-learning developments and experiences. Educ. Train. **43**(5), 240–248 (2001)
4. Alì, G., Bilotta, E., Gabriele, L., Pantano, P.: An e-learning platform for academy and industry networks. In: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), Pisa, pp. 231–234. IEEE Computer Society (2006)
5. Alì, G., Bilotta, E., Gabriele, L., Pantano, P., Servidio, R., Talarico, V.: An e-learning platform for applications of mathematics to microelectronic industry. In: Proceedings of the 14th European Conference on Mathematics for Industry (ECMI), Madrid, pp. 736–740. Springer (2007)
6. Alì, G., Bilotta, E., Pantano, P., Servidio, R., Talarico, V.: E-learning strategies in academia-industry knowledge exchange. In: Proceedings of the Interactive Computer Aided Learning, Villach, pp. 1–10. Kassel University Press, Kassel (2007)
7. Aspeli, M.: Professional Plone Development. Packt Publishing Ltd, Birmingham (2007)
8. Bertacchini, P.A., Bilotta, E., Gabriele, L., Pantano, P., Servidio, R.: Apprendere con le mani. Strategie cognitive per la realizzazione di ambienti di apprendimento-insegnamento con i nuovi strumenti tecnologici. Franco Angeli, Milano (2006)
9. Bilotta, E., Pantano, P., Rinaudo, S., Servidio, R., Talarico, V.: Use of a 3D graphical user interface in microelectronics. In: Proceedings of the Fifth Eurographics Italian Chapter Conference, Trento, pp. 217–224. IGD, Darmstadt (2007)

10. Bilotta, E., Pantano, P., Sepúlveda, J., Servidio, R.: Collaborative research and elearning platform for a distributed microelectronics project. Wseas Trans. Adv. Eng. Educ. **12**(10), 655–664 (2008)
11. Blume, J., Garcia, K., Mullinax, K., Vogel, K.: Integrating math and science with technology. Master of Arts Action Research Project. Saint Xavier University and Skylight Professional Development Filed-Based Program (2001)
12. Bonnaud, O.: Microelectronics technology course for a virtual campus. In: Proceedings of 2nd International Conference on Information Technology Based Higher Education and Training, Kumamoto, 4–6 July 2001, pp. 1–6. Kumamoto University (2001)
13. Bouras, C., Tsiatsos, T.: Educational virtual environments: design rationale and architecture. Multimed. Tools Appl. **29**, 153–173 (2006)
14. Brunetti, G., Servidio, R.: Conceptual design scheme for virtual characters. In: International Conference on Facets of Virtual Environments (FaVE), Berlin, pp. 1–12. Springer (2009)
15. Card, S.K., Newell, A., Moran, T.P.: The psychology of human computer interaction. Lawrence Erlbaum, Hillsdale (1983)
16. Carroll, M.J.: Interfacing Thought: Cognitive Aspects of Human-Computer Interaction. MIT, Cambridge (1987)
17. Chen, S., Zhang, J.: The adaptive learning system based on learning style and cognitive state. In: 2008 International Symposium on Knowledge Acquisition and Modeling (KAM'2008), Wuhan. IEEE Computer Society (2008)
18. Chen, H., Wu, S., Song, C., Zhan, J., Chen, J., Kang, D.: E-learning system model construction based constructivism. In: Fifth International Joint Conference on INC, IMS and IDC (NCM'09), Seoul, pp. 1165–1169 (2009)
19. Chu, C., Chang, C., Yeh, C., Yeh, Y.: A web-service oriented framework for building SCORM compatible learning management systems. In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, pp. 156–161. IEEE Computer Society (2004)
20. Ciuprina, G., Ioan, D., Munteanu, I., Rebican, M., Popa, R.: Numerical Optimization of Electromagnetic Devices. Printech, Bucharest (2002)
21. COMSON Consortium: Annex i: Description of work (2005). Contract for Marie Curie Project COMSON
22. da Luz Reis, R.A., Soares Indrusiak, L.: VRML and microelectronics education. In: IEEE International Conference on Microelectronics Systems Education/International Symposium on Multimedia Software Engineering, Arlington, p. 84 (1999)
23. Fetaji, F., Fetaji, M.: E-learning indicators methodology approach in designing successful e-learning. In: Proceedings of the International Conference on "Computer as a Tool", Cavtat (2007)
24. Gerbaud, S., Gouranton, V., Arnaldi, B.: Adaptation in collaborative virtual environments for training. In: Edutainment 2009, Banff, pp. 316–327. Springer, Berlin/Heidelberg (2009)
25. Ghaoui, C.: Encyclopedia of Human Computer Interaction. Idea Group Reference, Hershey (2006)
26. Gim, S., Vasenev, A., Stefanescu, A., Kula, S., Mihalache, D.: A novel graphical based tool for extraction of magnetic reluctances between on-chip current loops. In: Roos, J., Costa, L.R.G. (eds.) Scientific Computing in Electrical Engineering (SCEE 2008), Espoo. Mathematics in Industry. Springer (2010)
27. Goyal, M., Murthy, S.: Student perceptions on the use of new technologies in engineering courses recorded lectures on the internet and Moodle. In: International Workshop on Technology for Education (T4E'09), Bangalore, pp. 36–41 (2009)
28. Graf, S., Kinshuk, Liu, T.C.: Identifying learning styles in learning management systems by using indications from students' behaviour. In: Eighth IEEE International Conference on Advanced Learning Technologies, Santander, pp. 482–486. IEEE (2008)
29. Haake, M., Gulz, A.: Visual stereotypes and virtual pedagogical agents. Educ. Technol. Soc. **11**(4), 1–15 (2008)

30. Hamada, M.: An Integrated Virtual Environment for Active e-Learning in Theory of Computation. Springer, Berlin/Heidelberg (2007)
31. Hodgins, W.: IEEE LTSC Learning Technology Standards Committee P1484. ADLNET, USA (2001)
32. Horton, W., Horton, K.: E-learning Tools and Technologies: A Consumer's Guide for Trainers, Teachers, Educators, and Instructional Designers. Wiley, Indianapolis (2003)
33. Hsiao-Ya, C., Shi-Zong, W., Chieh-Chung, S.: Apply web 2.0 tools to constructive collaboration learning: a case study in MIS course. In: Fifth International Joint Conference on INC, IMS and IDC, 2009 (NCM'09), Seoul, pp. 1638–1643 (2009)
34. Hwang, W.Y., Su, J.H., Huang, Y.M., Dong, J.J.: A study of multi-representation of geometry problem solving with virtual manipulatives and whiteboard system. Educ. Technol. Soc. **12**(3), 229–247 (2009)
35. Xu, Z., Han, H., Zhang, Y., Zhang, C.: Research and practice on new interactive teaching model based on constructivist learning theory. In: IEEE International Symposium on IT in Medicine and Education, 2008 (ITME 2008), pp. 182,186, 12–14 Dec. 2008. doi: 10.1109/ITME.2008.474384
36. Janos, H., Zoltan, S., Istvan, N.: Signal processing by multimedia in nonlinear dynamics and power electronics: review. World Acad. Sci. Eng. Technol. **13**, 34–44 (2006)
37. Jimenez Orostegui, D.F., Soares Indrusiak, L., Glesner, M.: Proxy-based integration of reconfigurable hardware within simulation environments: improving e-learning experience in microelectronics. In: IEEE International Conference on Microelectronics Systems Education/International Symposium on Multimedia Software Engineering, Anaheim, pp. 59–60 (2005)
38. Junxia, K., Fengli, W.: Practice of blended learning in computer instruction. J. Hebei North Univ. **23**(3), 65–68 (2007)
39. Kao, F.C., Tung, Y.L., Chang, W.Y.: The design of 3D virtual collaborative learning system with circuit-measuring function. In: Proceedings of the 16th International Conference on Computers in Education, Kaohsiung, pp. 105–110. Asia-Pacific Society for Computers in Education (2008)
40. Kirschner, P., Kester, L., Corbalan, G.: Designing support to facilitate learning in powerful electronic learning environments. Comput. Hum. Behav. **23**, 1047–1054 (2007)
41. Kula, S., Vasenev, A.: Meshing strategies in the high frequency interconnects modelling. In: 14th Scientific Conference on Computer Applications in Electrical Engineering (ZKWE 2009), under the auspices of Electrical Engineering Committee of Polish Academy of Sciences and IEEE Poznan, Poland, April 2009, pp. 67–68. COMPRINT AR, Poznan, Poland (2009)
42. Kula, S., Vasenev, A.: Implementation of an Interactive E-Learning Education Network in the Field of Electrical Engineering. In: Baltic Conference "Learning in Networks", University of Rostock, 20–21 Aug 2010, pp. 53–60 (2010)
43. La Noce, F.: E-learning. La nuova frontiera della formazione. Franco Angeli, Milano (2001)
44. Lily, S., Shirley, W.: An instructional design model for constructivist learning. In: Proceedings of World Conference on Multimedia, Hypermedia and Telecommunication, Lugano, pp. 2476–2484 (2004)
45. Lin, Y.T., Cheng, S.C., Yang, J.T., Huang, Y.M.: An automatic course generation system for organizing existent learning objects using particle swarm optimization. In: Edutainment 2009, Banff, pp. 565–570. Springer, Berlin/Heidelberg (2009)
46. Liu, D., Ma, S., Ru, Q., Guo, Z., Ma, S.: Design of multi-strategic learning environment based on constructivism. In: First International Workshop on Education Technology and Computer Science (ETCS'09), Wuhan, vol. 3, pp. 226–228 (2009)
47. Luo, H., Li, X.: Research on the design of network courses based on constructivism. In: First International Workshop on Education Technology and Computer Science (ETCS'09), Wuhan, vol. 3, pp. 283–287 (2009)
48. Ma, Z.: Web-Based Intelligent E-Learning Systems: Technologies and Applications. Information Science Publishing, Hershey (2006)

49. Merriënboer, J.J.G.V.: Cognition and Multimedia Design for Complex Learning. Open University of the Netherlands, Heerlen (1999)
50. Merrill, P.F., Tolman, M.N., Christensen, L., Hammons, K., Vincent, B.R., Reynolds, P.L.: Computers in Education. Prentice-Hall, Englewood Cliffs (1986)
51. Minato, J., Mitsuhara, H., Kume, K., Uosaki, N., Teshigawara, N., Sakata, H., Yano, Y.: Student centered method to create learning materials for niche-learning. In: Proceedings of IADIS Multi Conference on Computer Science and Information Systems 2008 (e-Learning 2008), Amsterdam, vol. 1, pp. 177–184 (2008)
52. MIT: iLabs. http://icampus.mit.edu/projects/ilabs/
53. Mohtar, A., Nedic, Z., Machotka, J., Auer, M.E.: A remote laboratory for testing microelectronic circuits on silicon wafers under a microscope. In: Auer, M.E. (ed.) Proceedings of the Annual International Conference on Remote Engineering & Virtual Instrumentation (REV'08), Düsseldorf. Kassel University Press (2008). CD ROM
54. Nanko, R., Okada, Y., Konishi, T., Itoh, Y.: Constructing intelligent virtual laboratory for high school chemistry to support learners' consideration. In: Proceedings of the 16th International Conference on Computers in Education, Taiwan, pp. 105–110. Asia-Pacific Society for Computers in Education (2008)
55. Neo, M., Neo, T.K.: Engaging students in multimedia-mediated constructivist learning–students' perceptions. Educ. Technol. Soc. **12**(2), 254–266 (2009)
56. Newell, A., Simon, H.A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs (1972)
57. Nielsen, J.: Usability Engineering. Morgan Kaufmann, San Francisco (1994)
58. Oechsner, R., Pfeffer, M., Pfitzner, L., Ryssel, H., Beer, K., Boldin, M.: E-learning for microelectronics manufacturing. In: Proceedings of the Thirteenth International Symposium on Semiconductor Manufacturing, Tokyo (2004)
59. Ostermann, T., Lackner, C., Koessl, R., Hagelauer, R., Beer, K., Krahn, L., Mammen, H.-T., John, W., Sauer, A., Schwarz, P., Elst, G., Pistauer, M.: LIMA: the new e-learning platform in microelectronic applications. In: Proceedings of the International Conference on Microelectronic Systems Education, Anaheim, pp. 115–117. IEEE Computer Society (2003)
60. Ozkan, S., Koseler, R.: Multi-dimensional evaluation of e-learning systems in the higher education context: an empirical investigation of a computer literacy course. In: Proceedings of the 39th ASEE/IEEE Frontiers in Education Conference, San Antonio, pp. 1–6. IEEE Computer Society (2008)
61. Papanikolaou, K.A., Grigoriadou, M.: Towards new forms of knowledge communication: the adaptive dimension of a web-based learning environment. Comput. Educ. **39**, 333–360 (2002)
62. Pressey, L.S.: A machine for automatic teaching of drill material. Sch. Soc. **25**(645), 549–552 (1927)
63. Rice, H.W.: Moodle. E-Learning Course Development. A Complete Guide to Successful Learning Using Moodle. Packt Publishing Ltd., Birmingham (2006)
64. Roberts, T.S.: Computer-Supported Collaborative Learning in Higher Education. Idea Group Publishing, Hershey (2005)
65. Sarmento, A.: Issues of Human Computer Interaction. IRM Press, Hershey (2005)
66. Schön, A.D.: Educating the Reflective Practitioner. Jossey-Bass, San Fransico (1987)
67. SCORM: Sharable Content Object Reference Model (SCORM), Advanced distributed learning. ADLNET (2004)
68. Sepúlveda, J., Servidio, R., Gabriele, L., Alì, G.: Learning microelectronics through technology and research. In: Proceedings of the International Conference on Computer Science and Information Technology, Beijing, pp. 122–126. IEEE Computer Society, Los Alamitos (2009)
69. Shneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interactions. Addison Wesley Longman, Reading (1997)
70. Soares Indrusiak, L., da Luz Reis, R.A.: 3D integrated circuit layout visualization using VRML. Fut. Gener. Comput. Syst. **17**(5), 503–511 (2001)
71. Tomei, L.A.: Encyclopedia of Information Technology Curriculum Integration. Information Science Reference, Hershey (2008)

72. Urdan, T.A., Weggen, C.C.: Corporate e-Learning: Exploring a New Frontier. WR Hambrecht, San Francisco (2000)
73. Vasenev, A.: Collaboration and interaction by Bucharest node in RTN COMSON project. Abstracts of the Marie Curie Actions contribution to ESOF 2008, Barcelona (2008)
74. Vasenev, A., Stefanescu, A., Mihalache, D., Kula, S., Gim, S.: Interactive e-learning on reduced order modeling in electromagnetics using comson federated repository. In: Ninth IEEE International Conference on Advanced Learning Technologies (ICALT'09), Riga, pp. 408–409. IEEE Computer Society (2009)
75. Weitershausen, V.: Web Component Development with Zope 3. Springer, Berlin/New York (2008)
76. Williams, J., Rosenbaum, S.: Learning Paths: Increase Profits by Reducing the Time It Takes to Get Employees Up-to-Speed. Pfeiffer, San Francisco (2004)
77. Xu, Z., Han, H., Zhang, Y., Zhang, C.: Research and practice on new interactive teaching model based on constructivist learning theory. In: IEEE International Symposium on IT in Medicine and Education (ITME'2008), Xiamen, pp. 182–186 (2008)
78. Yongxing, W.: Blended learning design for software engineering course design. In: Proceedings of the CSSE, Wuhan, vol. 5, pp. 345–348 (2008)

Index

Adapted POD, 374
Adaptive and personalized eLearning contents, 508
Adjoint
  backward adjoint method, 328
  Backward Reduced Adjoint Method (BRAM), 331
  equations, 320
  method, 324
  model, 325
Adjoint field technique (AFT), 315
AMOR module, 459
Arnoldi method, 176
Asymptotic waveform evaluation, 176

Backward Differentiation Formula (BDF), 321
Backward Euler method, 321
Balanced truncation, 181, 243, 369, 412
  bilinear systems, 367
  Hamilton-Jacobi differential equations, 366
  nonlinear problems, 366
  Sliding Interval Balancing, 366
  Truncated Balanced Realization (TBR), 184, 272
Balun, 408
BDF, *See* Backward Differentiation Formula
Bilinear systems, 365, 377
BIM module, 458, 491
Bode plot, 185
Boltzmann equation, 78, 92
Boltzmann-Peierls equation, 78

CHAMY, 315
CHORAL, 414
Class A, 458, 486, 491
Class B, 458, 486, 491
Class C, 458, 486, 494
Co-simulation, 103
Codestar, 314
Cognitive psychology, 508
Collaborative and cooperative learning, 525
CORE, 275, 280
Covariance, 338
CRS module, 460

DAEN module, 458
D4MEKAI module, 459
Deflation, 198, 203
Demonstrator platform, 456, 485
  coupled simulation, 480
  electro-thermal simulation, 485
  micro-hotplate, 463
  MOR case studies, 463
  n-channel power MOS-FET, 486
  transmission line, 467
Differential algebraich equations, 15–17, 21
Differential Equation Macromodel (DEM), 304
Discrete Empirical Interpolation Method (DEIM), 375, 378
Displacement current, 25, 27, 28, 65
Dominant
  parameter, 341
  pole, 187, 195

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8

spectral zeros, 230, 232
zero, 207
Dominant Pole Algorithm
subspace accelerated, 201
with deflation, 199
Dominant Spectral Zero Method (dominant SZM), 232, 234
Dynamic iteration, 103
contraction condition, 119
error recursion, 114
Gauss-Seidel iteration, 110
Jacobi iteration, 110
Picard iteration, 110
simple coupling, 121
stability, 123

Educational environments for eLearning, 555
eLearning methodologies, 504, 513
eLearning systems, 504, 508, 509, 514, 553, 554
Electro-static discharge analysis (ESD), 381
Electro-thermal network, 45, 46
Electro-thermal simulation, 485
Electromagnetic effect, 17
Electron-phonon system, 78, 79, 85
ε-embedding, 342
ET_MEP_MOSFET module, 459
ETH module, 491
ETMEP1D module, 459
Example
class A, 458, 486, 491
class B, 458, 486, 491
class C, 458, 486, 494
Expected value, 334, 337
External library, 457

FIDES module, 459
Finite difference method, 33
Finite element method, 33, 52
Finite Integration Technique, 69
Finite Integration Technique (FIT), 290
Finite-Difference Time-Domain (FDTD), 290
FPL module, 460

Galerkin, 175
Generalized Polynomial Chaos (gPC), 333
expansion, 335
GLIMDA module, 458
Gramian
algebraic, 367
controllability, reachability, 181, 366
cross, 377
observability, 181, 366
Graph partitioning
(h)METIS, 383
Sparse Modal Approximation (SparseMA), 381, 385, 388
Gummel map, 33, 34, 37, 41

Hankel Singular Values, 182, 413
Heat equation, 17, 20, 21
Hodge operators, 291

Inductive coupling, 398
Injection locking, 399
Interpolating input-output behavior, 377

Jitter, 327, 399

Kane dispersion relation, 98
Karhunen–Loève expansion, 179, 372
Kirchhoff's laws, 16, 22, 64
Krylov method, 175
Krylov space, rational, 210

Lagrangian multiplier, 325
Lanczos method, 176
Laplace transform, 169
Library, external, 457
Linear multi-step method, 51
Linear periodically time varying (LPTV) problems, 365
Lyapunov equations, 184
generalised, 367

Magnetoquasistatic, 20, 21, 64, 65
Matrix reordering
AMD, 384
BBBD, 384
Maximum entropy principle, 78, 82
Maxwell equations, 20, 21, 64, 65, 288, 294
Maxwell Grid Equations (MGE), 290
Mean global sensitivity, 339
Mean, expected value, 334, 337
MEMS, 463
Method of lines, 69
Micro-hotplate, 463
Missing Point Estimation (MPE), 373

MNA, 467
Modal approximation, 193, 243, 388
  Sparse Modal Approximation (SparseMA), 381, 385, 388
Modal equivalent, 195
Modal Truncation, 185
Modified Nodal Analysis, 22
Modified nodal analysis, 16, 467
Module, 457
  AMOR, 459, 461
  BIM, 458, 491
  CRS, 460
  DAEN, 458
  D4MEKAI, 459
  ETH, 491
  ETMEP1D, 459
  ET_MEP_MOSFET, 459
  FIDES, 459
  FPL, 460
  GLIMDA, 458
  MOR, 459
  MOR4ANSYS, 459, 462
  MSH, 460
  OCS, 459, 482, 490
  OPTBOOK, 460
  RADAU, 458
  ROMI, 459
  ROM-WB, 459
  SECS1D, 460
  SECS2D, 460
  SECS3D, 460
  SKIF, 459
  SLICOTINT, 459, 461
Monte Carlo (quasi), 337
MOR module, 459
MOR4ANSYS module, 459
MOS-FET, 486
MSH module, 460
Multi-Parameter Moment Matching, 273
Multi-terminals (ports), 381

Newton method, 35
Nonlinear phase macromodel, 399

Observation function, 324
  augmented, 325
  sensitivity, 324
OCS module, 459, 482, 490
OPTBOOK module, 460
Oscillator
  autonomous system, 400
  capacitively coupled, 416
  coupled to balun, 408, 421
  coupled to transmission line, 412, 423
  free-running, 400
  inductively coupled, 398, 414, 419
  injection locking, 399
  locking, pulling, 398
  mutually coupled, 404
  orbital deviation, 401
  Perturbation Projection Vector (PPV), 401
  Phase Locked Loop (PLL), 399
  phase noise theory, 400
  phase shift, 401
  Voltage Controlled Oscillator (VCO), 398

Parameterized, parametric Model Order Reduction (pMOR), 269, 320, 334, 339
  ε-embedding, 343
  Piecewise $\mathcal{H}_2$-Optimal Interpolation pMOR, 341
  PIMTAB, 349, 351
  Uncertainty Quantification (UQ), 339, 340
Partial differential algebraic equations, 15, 17
Partial differential equations, 15, 21, 24, 34, 65
Passive Parameterized Time-Domain Macro Models, 274, 281
Passivity, 172, 177, 215, 217, 242
  preserving, 221, 237
  preserving MOR (Antoulas), 226
  preserving MOR (Sorensen), 222
Periodic Steady State (PSS), 400
Perturbation Projection Vector (PPV), 401
Perturbation-based Projector Fitting scheme, 272
Per unit length (p.u.l), 295
Phase Locked Loop (PLL), 399
Phase noise theory, 400
Phase shift, 401
PIMTAB, 349
Poisson equation, 24, 26, 30, 31, 35, 37–41, 78
Pole Analysis via Congruence Transformations (PACT), 381, 385, 386
Polynomials
  Hermite, 335, 336
  Legendre, 335, 336
  orthonormal, 334
Poor Man's TBR (PMTBR), 184, 187, 243, 347, 349, 351, 369
  Variational (VPMTBR), 277, 281
Positive real, 172, 217
  Balanced Truncation (PRBT), 230, 243
Power MOS-FET, 486
Power Spectral Density (PSD), 414

PRIMA, 177, 187, 221, 230, 234, 243, 248, 272, 280
Principal Component Analysis, 179, 372
Probability
distribution density, 334
space, 334
Proper Orthogonal Decomposition (POD), 179, 320, 329, 367, 372, 377
Adapted POD, 374
Discrete Empirical Interpolation Method (DEIM), 375, 378
Missing Point Estimation (MPE), 373
Pstar, 235

Quadratic-bilinear ODE-systems, 377
Quadrature
Gaussian, 337
multi-dimensional, 337
sparse grids, 337
Quantum effect, 16, 21, 78, 90, 93, 96

RADAU module, 458
Rayleigh Quotient Iteration, 197
Realizability, 173
Foster impedance realization, 235, 242, 244
Reciprocity, 173
RLCSYN, 242, 245, 248
ROM-WB module, 459
ROMI module, 459

SADPA, 187, 201, 230, 237, 239, 385
SAMDP, 207, 385
frequency dependent, 386, 389
SASPA, 209
Scharfetter-Gummel discretization, 35, 37, 38, 40, 41
Search space, 201
SECS1D module, 460
SECS2D module, 460
SECS3D module, 460
Selection strategy, 202
Semiclassical Boltzmann equation, 78, 92
Semiconductor, 15, 17–21, 23, 24, 29, 78, 79, 82
Sensitivity, 320, 338
analysis, 320
eigenvalue, 208
observation function, 324
pole, 306
residue, 306
Simulation, electro-thermal, 485
Singularly perturbed system, 342
ε-embedding, 342
SKIF module, 459
SLICOTINT module, 459
Snapshot, 330, 369, 373
Sparse Modal Approximation (SparseMA), 381, 385, 388
Sparsity preservation, 381
Spectral zeros, 219, 221, 231
interpolation, 232, 243
MOR (Antoulas), 226
MOR (Sorensen), 222
SPICE, 242, 244, 288
Splitting function, 108
SPRIM, 178, 187, 234, 242, 243, 245, 248
Stability, 171, 216
Stochastic Collocation, 337
Stochastic Galerkin, 337
Structure preservation, 173, 178, 278
Structure preserving block (BSP), 283
Subspace acceleration, 187, 201
Synthesis, 173, 178, 242, 305, 310, 381
Foster, 242, 243
parametric, 313
RLCSYN, 242, 245, 248

Table modelling, 377
Thick restart, 204
Trajectory PieceWise Linearization (TPWL), 367, 377
Transfer function, 170
ε-embedding, 344
MIMO, 207
moments, 176
pole, 185, 194
Pole-Residue Representation, 186, 194
positive real, 242
zero, 185, 207
Transistor, 16, 17, 19
Transmission line, 17–20, 467
Truncated Balanced Realization (TBR), 184

Un-stamping, 173, 242
Uncertainty Quantification (UQ), 333
covariance, 338
mean global sensitivity, 339
mean, expected value, 334, 337
parameterized, parametric Model Order Reduction (pMOR), 339, 340
sensitivity, 338
Stochastic Collocation, 337

  Stochastic Galerkin, 337
  variance, 338
University versus industry training, 512
User needs analysis, 516

Variance, 338
Variational multi-point moment matching, 277
Vector Fitting (VFIT), 303, 316
Voltage Controlled Oscillator (VCO), 398
Volterra-series expansion, 365, 377

Wigner equation, 78, 92
Wigner-Poisson system, 90