NOTE TO USERS

This reproduction is the best copy available.

UMI[®]

Transient Simulation of Thermal Networks Using Multi-Dimensional Model Reduction

by

Michal Brajtman, B. Sc.

A thesis submitted to the
Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Applied Science

Ottawa-Carleton Institute for Electrical Engineering
Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Canada

© Michal Brajtman, 2004

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-97401-4

Our file Notre référence

ISBN: 0-612-97401-4

The author has granted a non-exclusive license allowing the Library and Archives Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

A new approach for thermal transient analysis is introduced in this work. This approach makes use of the Multi-Dimensional Model Reduction (MDMR) technique followed by an Eigen Decomposition of the reduced model. The reduced solution is easily mapped back to the original model by a transformation matrix. By exploiting the multi-dimensional nature of the model reduction, boundary, as well as initial conditions, are incorporated explicitly into the reduced model, resulting in a boundary and initial conditions independent (BICI) model. This allows to perform multiple simulations with different sets of boundary conditions (BC's) applied to the reduced model. The transient solution obtained using the proposed method is much faster than existing transient solvers. It offers a speed-up of two orders of magnitude.

The proposed method is also applied to complex thermal systems consisting of a base-model and one or more component-models linked to it. MDMR is performed on the component-models, creating BICI macro-models. The *mixed method* is presented as an efficient way to link macro-models to a base-model in such a way that all boundary conditions of both the base-model and the component-models are explicitly incorporated. A second reduction is then performed on the mixed-model, resulting in a reduced mixed-model that can be quickly solved.

Among the advantages of this method is the ability to easily generate the nodal temperatures for both the base-model and component-models at any desired time point for various boundary conditions as well as any power excitation of the component-models. The proposed methodology offers not only great speed-up but it also enables transient simulations of complex systems, consisting of many component-models. These systems cannot be solved using existing methods due to their size and complexity.

The reduced/macro-models have shown a number of advantages over models derived from resistor network models in terms of creation, accuracy, flexibility and capability of predicting multiple junction temperatures.

Acknowledgments

I would like to express my gratitude to my academic and research supervisor Prof. Tom Smy, whose valuable guidance, insight and patience made this thesis an enjoyable experience for me.

I would also like to thank Prof. Pavan Gunupudi and Prof. Michel Nakhla for their invaluable help, guidance and feedback.

Special thanks are extended to Mr. Dritan Celo for his helpful comments, ideas and suggestions.

Many thanks to my fellow students at Carleton, Michael and Greg, for the lively discussions and many laughs we shared.

I leave a special note of appreciation to my family for their love and many years of unconditional support and encouragement.

Dedicated to Simon
for his love and support

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Transient Simulations	3
1.3	Thermal Modeling	5
1.4	Thesis Methodology	8
1.5	Electro-Thermal Analogy	11
1.6	Outline of the Thesis	12
2	Theoretical Background	14
2.1	<i>Atar</i> Simulator	14
2.2	Model Reduction Techniques	16
2.2.1	Concepts of Moment Matching Techniques	16
2.2.2	Moments of the Response	17
2.2.3	Direct Moment Matching	19
2.2.4	Padé Approximation	19
2.2.5	Single Expansion Moment Matching	20
2.2.6	Limitations of Direct MMT Algorithms	23

2.2.7	Multi-Point Moment Matching	24
2.2.8	Indirect Moment Matching Techniques	26
2.2.9	The Arnoldi Algorithm	26
2.2.10	Model-Reduction through Arnoldi	28
3	Thermal Applications of Multi-Dimensional Model Reduction	30
3.1	Introduction	30
3.2	Numerical Generation of Thermal Network	
	Equations	32
3.2.1	Explicit Temperature Boundary Conditions	34
3.3	Multi-Dimensional Model Reduction	39
3.3.1	Krylov-based Model Reduction Technique	39
3.4	Eigen Decomposition of the Reduced Multi-Dimensional System . . .	47
3.5	Computational Issues	50
3.6	Results	50
3.6.1	Determination of Model Parameters	54
3.6.2	Rectangular Models - Heat Flow Variation	67
3.6.3	CBGA Model	76
3.7	Summary	83
4	Transient Macro-models for Electronic Components Using Model Reduction	85
4.1	Introduction	85
4.2	Transient Thermal Macro-models	87
4.2.1	Macro-model Definition	87

4.2.2	Macro-model Equations	91
4.3	Model Reduction Technique - Theoretical Approach	93
4.3.1	Computation of the Multi-Dimensional Subspace	96
4.4	Forming the Mixed Matrices	101
4.5	Eigen Decomposition of the Mixed Matrices	105
4.6	Computational Issues	107
4.7	Results	108
4.7.1	Example 1: Simple Square Model	109
4.7.2	Example 2: Simple Rectangular Model	113
4.7.3	Effect of the Port Configuration	119
4.7.4	Large PCB and Package Example	123
4.7.5	Pulse Power Simulation	132
4.8	Summary	135
5	Conclusions and Future Work	137
5.1	Summary	137
5.2	Future Work	138
A	Simulation Summary	140
A.1	MDMR-ED Simulations	140
A.2	Macro-model Method Simulations	142

List of Figures

2.1	A two-node circuit	21
2.2	Poles captured a) using AWE b) using CFH	25
3.1	Boundary Conditions	36
3.2	A depiction of the thermal model of single <i>Atar</i> block. The block has a mixed BC's on the top face and fixed BC's on the front face. All other faces are connected to adjacent blocks.	37
3.3	Dimension and boundary conditions	51
3.4	Different mesh grids	52
3.5	Transient response for different h values at three expansion points . .	55
3.6	Error measurements for $h = 10^{-12}$	56
3.7	Error measurements for $h = 10^{-5}$	56
3.8	Error measurements for $h = 10^{-3}$	57
3.9	Number of Moments	59
3.10	Cross Derivatives	61
3.11	Initial conditions and BC's variation	65
3.12	200 and 62,096 block model results	68

3.13 Comparison of the analytical and MDMR-ED solutions for the 62,096 block model	70
3.14 Comparison of Analytical, TLM and MDMR-ED solutions for the 62,096 block model	71
3.15 Steady State temperature rise as a function of h	72
3.16 Temperature contour plot of the 62,096 block model	73
3.17 Speed Up as a function of number of simulations	74
3.18 Model of a low-profile capped CBGA	77
3.19 CBGA model results	79
3.20 CBGA Speed Up	80
3.21 SS Temperature rise as a function of h	80
3.22 CBGA model with three h 's	81
3.23 CBGA model contour plots	82
4.1 Component-model linking to a base-model	89
4.2 Macro-model linking to a base-model	90
4.3 Process flow for creating BCI reduced macro-models and forming a mixed-model.	103
4.4 Square model	111
4.5 Results for a square model	112
4.6 Mesh and contour plot for a rectangular model a) Mesh. The base- model consists of 385 blocks and the component-model of 5986 blocks b) Temperature contour plot at SS for the full model	114
4.7 Maximum temperature-rise as a function of time	115
4.8 Component-Model Error	116

4.9	Base-Model Error	117
4.10	Different Port Configurations	120
4.11	Error for the component-model node	121
4.12	<i>Atar</i> models:	124
4.13	Simulation time for PCB model as a function of the number of connected package models	127
4.14	Heat coefficient assigned	128
4.15	Results for a large scale model	130
4.16	Error due to Port Configurations	131
4.17	a) Temperature rise for each set of packages with varying power in each set b) Temperature rise for Package 6 for three different excitations. The lower part of the plot illustrates when particular packages are “on” for each power configuration	133
4.18	Results for pulse input	134

List of Tables

1.1	Analogy between Thermal and Electrical quantities	12
3.1	Computational cost	50
3.2	Simulation times for rectangular models. All runs were done on a SUNBlade 100 workstation	74
3.3	Simulation times for CBGA model	78
4.1	Computational cost	107
4.2	Model Sizes for PCB base model and a variable number of packages .	126
A.1	MDMR-ED Simulations	142
A.2	Macro-model Method Simulations	145

List of Acronyms

IC	Integrated Circuit
3D	Three Dimensional
CPU	Central Processing Unit
BGA	Ball Grid Array
C4	Controlled Collapsed Chip Connection
PCB	Printed Circuit Board
SS	Steady State
BC's	Boundary Conditions
BCI	Boundary Conditions Independent
BICI	Boundary and Initial Conditions Independent
TLM	Transmission Line Matrix
FDM	Finite Difference Method
FEM	Finite Element Method
SOR	Successive Over Relaxation
MMT	Moment Matching Techniques

AWE	Asymptotic Waveform Evaluation
CFH	Complex Frequency Hopping
LU	Triangular Matrices
MR	Model Reduction
MDMR	Multi Dimensional Model Reduction
MDMM	Multi Dimensional Macro Model
ED	Eigen Decomposition
PPB	Port Per Block
w.r.t.	with respect to

List of Symbols

$\mathbf{C}$	matrix of discretized thermal capacitance [J/W]
$\mathbf{G}$	matrix of discretized thermal conductance [W/K]
$\mathbf{P}$	vector of thermal excitations [W]
$\mathbf{A}_h$	matrix of conductance of the mixed BC's surface nodes [m^2]
$\mathbf{A}_s$	vector of power flow from mixed BC's surface nodes [m^2]
$\mathbf{G}_f$	vector of power flow from fixed BC's surface nodes [W/K]
$\mathbf{V}_p$	vector of volumes with non-zero power excitation [m^3]
T_m	reference (ambient) temperature [K]
T_f	surface temperature [K]
P_j	volumetric power excitation [W/m^3]
h	heat transfer coefficient ($W/\mu m^2 K$)
h_{exp}	expansion point for h
$\mathbf{T}_p$	vector of port temperatures [K]
$\mathbf{I}_p$	vector of port currents [W]
n_p	number of ports

$\mathbf{A}$	Matrix A
$\mathbf{A}^T$	Matrix A transposed
$\mathbf{A}^{(r)}$	r -th derivative of A
$\mathbf{0}$	Matrix of zeros
$\mathbf{I}$	Identity matrix
$\mathbf{I}_q$	Identity matrix of size $q \times q$
$\Re^N$	The set of all real vectors of size N
$\Re^{N \times N}$	The set of all real matrices of size $N \times N$
$(d_{i,j})$	Matrix composed of elements $d_{i,j}$ in the i -th row and j -th column
s	Laplace operator ($= j\omega$)
$\kappa(\mathbf{A}, \mathbf{R}, q)$	Krylov subspace
$\mathbf{Q}$	multi-dimensional subspace or orthogonal congruent matrix
q	reduced vector of unknowns in reduced model
$\mathbf{M}_n^p$	n -th moment vector for an excitation at port p
κ	thermal conductivity [W/mK]
C	thermal capacity [J/KgK]
ρ	material density [Kg/m^3]
Subscripts	
p	associated with a port
b	associated with a base model
c	associated with a component-model
m	associated with a mixed model

Chapter 1

Introduction

1.1 Motivation

Two main issues motivated this work. The first is the importance of accurate thermal modeling and the second is the necessity of transient analysis of devices.

The classical theory of electrical circuits does not consider the heating phenomena which takes place inside electronic devices. It regards temperature as a fixed physical property and self heating as a disturbance of normal electrical operation [1]. Thermal effects become important every time the specific power dissipation is relevant. Miniaturization of electronic devices results in placement of more functions in smaller packages, creating higher power densities in the package or module leading to more difficult and complicated thermal design problems that can no longer be overlooked by designers. Shrinking the cycle-time from design to production give rise to additional thermal considerations.

Therefore, thermal analysis of solid-state devices has been the object of increas-

ing interest for several years. Thermal problems of dissipating heat from devices are extremely important in many applications spanning from digital circuits to analog circuits and power devices [2]. Unwanted effects such as hot spot formation, thermal runaway and thermal coupling between neighboring devices directly impact the reliability and the performance of semiconductor devices and circuits [3], [4].

A variety of physical failure mechanisms have been identified and more than a half of failures of electronic parts are related to thermally overheating. For instance, thermal stresses and material degradation are one cause of breakdown in electronics components. Mismatch of the thermal coefficients expansion between two different materials, especially at the interface conditions, could result in the separation of interfaces and bonds between different parts in a module at higher temperature. In addition, fatigue in the solder connections and cracking in substrate are common failure in electronics components when operating at off-limit temperature [5].

Thermal and electronic effects couple at all levels, from chip to package and system level. An appropriate thermal management imperative is to maintain the electrical performance of the chips and packages. It is important to develop mathematical models able to detect thermal problems in order to properly design the device or circuit layout and optimize the thermal characteristics.

Currently, even though the thermal modeling at systems level can be carried out for analysis of parametric trends, the predictive capabilities of these models are rather limited in most cases. To predict the operating temperature of various critical components as a function of space and time purely through modelling is still not feasible in most cases [6]. An efficient thermal modeling including both chip and system level is required. The method should be fast, accurate and able to predict all critical

temperatures.

1.2 Transient Simulations

For a growing number of applications, the characterization of the transient thermal response of a system, also known as the “heating curve”, is important.

Transient thermal modeling and analysis is used during the design of packages, modules and systems to provide information for reliability assessment, thermal coupling, and to model the switching performance of semiconductors power amplifiers with significant self-heating [3]. In addition, it is also used to determine effective circuit board thermal conductivities [7] and provides information about internal heat flow paths in complex modules and packages [8].

Maximum operating power is usually limited by thermal constraints. Since modern circuits operate at ever increasing frequencies, power rating in pulsed mode heavily relies on dynamic thermal effects [9]. This includes IGBT failure analysis [10] and avalanche [11].

Temperature gradients are more severe in the dynamic regime than at steady-state (SS). Not only do they affect performance due to thermal unbalance between circuit devices, but they also have a negative effect on reliability. Dynamic thermal stresses are far more dangerous on die attach, vias and interconnects than static stresses due to their higher amplitude and repetitive nature causing fatigue [12], [13]. Transient temperature map of the chip is of great importance to study those effects. Transient thermal analysis would also help studying electro-thermal problems including self-heating effects [3] and self-sustained oscillations resulting from the non-linear dynamic

interaction between thermal and electrical state variables [14].

At the package, PCB and system levels, thermal time constants are larger and it may seem that thermal dynamics are less important. However, previous work shows that dynamic testing is the most efficient way to detect package defects such as voids in the solder-die attach resulting from thermal stresses [15], [16]. Another interesting point is that dynamic testing can be a better way to obtain static characteristics such as the thermal resistance formed by the interface between device and board, heat sinks or cold plates [17].

Micro-Electro-Mechanical-Systems (MEMS) is another application where transient thermal phenomena are important. Both for sensors, which are made small to be able to capture fast dynamic effects [18], [19], as well as actuators which are sometimes actuated by thermal expansion [20], [21]. Dynamic thermal analysis is needed to study their behavior. Integrated optical systems such as laser devices, are also known to be highly sensitive to temperature dynamics [22].

Large temperature gradients and peak temperature values pose serious limitations on manufacturing and assembly processes. Hence, the need of investigating the package thermal behavior during the assembly process is highly important. Previous work includes study of thermal behavior during Rapid Thermal Processing (RTP) [23] and Anisotropic Conductive Film (ACF) [24] processes.

In addition, there are new trends in the thermal management of portable devices, based on the use of thermal inertia, together with an adequate control of system activity (dissipated power) to reduce cooling load requirements [25].

All of the above issues will play an important role in rapidly advancing nanotechnologies [26].

1.3 Thermal Modeling

In thermal modeling, a few terminologies are commonly used. A *detailed* model generally means a model that attempts to represent the physical geometry of a device very closely. A detailed model will usually look physically similar to the actual device geometry [1], [27]. By contrast, a *compact model*, or a *macro-model* is a simplification of a detailed model of a device. It is a mathematical representation of a component, which models the component's thermal behavior by a linear relationship of surface temperatures and heat flows [28]. In general, the size of a macro-model may be larger than a compact model and it is usually not suitable for hand calculation. In this thesis, the term *component-model* is used as the detailed model of a component to be reduced to a macro-model. The use of macro-models allows for the replacement of complex parts of a detailed model with a simple mathematical relationship between surface temperatures and heat flows. A full thermal simulation of such system will involve a detailed numerical model referred to as the *base-model* to which component-models are linked through thermal ports. The component-models are then reduced, using model reduction techniques, to macro-models. The macro-models are linked to the detailed base-model, creating a *mixed-model*.

Given the high degree of geometrical complexity of modern electronic packages and modules analytical approaches to thermal modeling are very limited. The only practical approach to thermal analysis of realistic systems is the construction of three dimensional (3D) numerical models. It is also important to note that accurate solution of the thermal problem in electronic packages often requires a multi-scale approach. The heat sources present (individual active elements such as transistors) typically range in size from sub-micron to 10's of microns in size. High thermal gradients are

often present throughout the chip and package, requiring meshing techniques that span orders of magnitude in size.

Detailed transient thermal models of components are commonly built using numerical methods based on finite-difference (FD) or finite-element (FEM) methods. Another similar technique useful for transient analysis is the Transmission Line Matrix (TLM) [29]. These methods can provide a very detailed description of the heat flow and temperature field in the model and are suitable for detailed design, particularly where the design options have been narrowed to a few choices [6]. These methods are, however, computationally intensive and their systematic use is often limited by the complexity of the structure to be modeled and the expected simulation effort. The need to represent small scale geometric detail of a component or package is associated with an increase of the model size and a concomitantly large system of equations that must be solved. Transient solutions for such large systems is often prohibitively CPU expensive taking hours or days of simulation time.

Most thermal macro-models in use employ some kind of thermal network. In its most elementary form it consists of a single resistor. In order to account for various boundary conditions and temperature variations, Bar-Cohen [30] extended the single thermal resistor model to include a star-shaped network resistor topology in which thermal resistors radiate from the junction node to nodes on the surface and to the nodes for the leads. Later it was found that the boundary conditions of the packages should be governed not only by a fixed temperature but also for convective heat transfer by a heat transfer coefficient h and environment temperature T_m . Lasance [31] proposed an optimization approach to determining the resistance values for intuitively chosen network topologies. These methods, however, are limited to the analysis of

the package itself.

In the European community, a great effort has been dedicated to the methodology of thermal compact models. The research project DELPHI (Development of Libraries and Physical Models for an Integrated Design Environment), initiated in 1993 [32], has been concerned with the creation and experimental validation of Boundary Condition Independent (BCI) compact thermal models of packages. DELPHI, the three-year project which ended in November 1996, was followed by another project funded by the European Union SEED (Supplier Evaluation and Exploitation of DELPHI) in which component suppliers evaluated the DELPHI methodology. SEED came to an end in 1998.

The DELPHI approach suffers from a few limitations. The major one is that DELPHI methodology is limited to use for simple PCB packages, not for other components. Another limitation stems from the process itself, which includes a time-consuming optimization and simulations of the fully detailed model. Another issue is how to use a compact model at board and system level simulation. Previous work [33]–[35] show that compact thermal models mounted on boards yield 5% error in predicting junction temperatures (compared to the results found using the fully detailed model), but the error rises in the presence of high thermal gradients (10%).

Several other methods are reported in literature to simplify thermal modeling. Among them are the RC ladder approach which is based on fitting an RC ladder network on the observed system response using a suitable error minimization technique. In this method, the internal nodes are successively eliminated until only the external nodes exchanging heat with the surroundings remain [36]. Another approach [37] relies on using a coarser meshing to reduce the model size. This is done by ignoring

technical and process details as well as detailed geometry of the package. Genetic algorithms (adaptive search procedures) are also used to create compact models. They are based on identifying the variables such as the boundary conditions and thermal conductivities that are most important in a system [38].

Model Reduction (MR) techniques have been suggested for the efficient solution of thermal transient problems [39]. Among those techniques are the circuit reduction method [40] and the Krylov sub-space methods such as Lanczos and Arnoldi algorithms [41]–[43].

Model reduction, in general, captures the behavior of a large set of differential/algebraic equations with a reduced (smaller) system. Typically, a set of equations characterized by a sparse matrix is reduced down to much smaller set of strongly coupled equations. For a thermal problem MR, in essence, maps the vector of unknown temperatures $\mathbf{T}$ of dimension N into a smaller vector $\hat{\mathbf{T}}$ of dimension q ($q \ll N$) using the orthogonal matrix $\mathbf{Q}$ in the relation $\mathbf{T} = \mathbf{Q}\hat{\mathbf{T}}$ [44]. The problem can then be solved quickly in the reduced system and the full solution obtained by a mapping back to the large set of unknowns.

1.4 Thesis Methodology

One disadvantage of traditional MR techniques is that for a given component, a change in the BC's of the component (for example the heat flow off the top side of the package or board) requires a new reduced model to be created. Very often in electrical system design the component's physical geometry is unchanged, however, the thermal boundary conditions are a design parameter to be determined by heat sink design,

board structure and component placement. It would therefore be advantageous to be able to quickly determine a detailed transient response for a variety of BC's for a single physical component.

As a solution to this problem the use of Multi-Dimensional Model Reduction (MDMR) technique to create thermal reduced models is proposed. This technique was introduced recently [45], [46] and allows for model parameters to be incorporated explicitly into the reduced model. MDMR was previously used for steady-state thermal simulations [45], [47], but in this work, it is applied to transient simulation.

This work shows that the use of MDMR for transient thermal analysis provides both fast and accurate solution. The method is applied to the quick analysis of complicated 3D models by combining a numerical thermal simulator *Atar* [48] with MDMR. *Atar* is used to generate a full 3D model and the system of equations characterizing the thermal network. The boundary conditions in the detailed model are explicitly incorporated into the reduced model by exploiting the multi-dimensional nature of the model reduction algorithm. Therefore, any given set of BC's can be applied to the reduced model through a set of parameters. A key point of MDMR is that the reduction procedure is only performed once, generating a BCI reduced-model.

In chapter 3, transient solution of different electronic packages is obtained using MDMR. The solution is much faster than any other existing method, due to the small size of the reduced model. Speed up of two orders of magnitude will be presented.

In chapter 4, MDMR is applied to systems consisting of a base-model with one or more component-models attached to it. MDMR is performed on each of the component-models, resulting in macro-models. The macro-models are then linked to the base-model and the system is quickly solved. The major advantage of this

methodology is its ability to solve large systems that, due to their size and complexity, can not be solved in any other way.

Both the reduced models generated in chapter 3 and the macro-models from chapter 4 have been shown to have a number of advantages over resistor network models:

1. The creation of the reduced/macro-model is far faster than the optimization of a specified resistive network.
2. All internal temperatures of the original detailed model are accessible allowing for the prediction of multiple junction temperatures.
3. The accuracy of the temperature prediction is very high because the method does not restrict the model to obey a specific network model.
4. The method allows for design parameters (geometry, heat flow off the surface, ambient temperature and multiple power levels) to be incorporated explicitly into the reduced/macro-model.
5. The method is general, allowing for any reasonable number of thermal ports, power sources and external heat flows.

After applying MDMR to the model, Eigen Decomposition (ED) is performed. Using the poles and residues of the reduced model, the transient solution in the reduced space is obtained. For every change in the model's boundary or initial conditions, the eigen decomposition is the only procedure that has to be repeated. Since the size of the reduced model is very small (usually less than 20), this process is very quick.

The procedure followed to obtain a transient response in this thesis is therefore:

1. Build the model in *Atar* and generate the thermal equations describing the model.
2. Create the reduced/macro-model using MDMR.
3. Perform Eigen Decomposition to obtain the poles and residues of the reduced system and generate a transient response in the reduced space.
4. Map the reduced solution back to nodal temperatures and heat flows in the original *Atar* model.

This process is referred to as MDMR-ED (Multi-Dimensional Model Reduction - Eigen Decomposition).

1.5 Electro-Thermal Analogy

A thermal system is found to be analogous to an electrical system if they have similar equations and BC's. The equations describing the behavior of an electrical system can be transformed into the equations for the analogous thermal system by simply changing the variables. The analogy between thermal and electrical systems allow us to predict the thermal characteristics with widely known electrical basic laws and methods for electrical circuit simulations, such as the model reduction technique described in Section 2.2. The summary of thermal variables, their units and thermal electrical analogies are given in Table 1.1.

Thermal Quantity			Electrical Quantity		
		Unit			Unit
Temperature Rise	ΔT	(K)	Voltage	V	(V)
Heat Flow (Power)	P	(W)	Current	I	(A)
Heat	Q	(J=W.s)	Charge	Q	(C=A.s)
Thermal Resistance	R	(K/W)	Resistance	R	($\Omega=V/A$)
Thermal Conductance	G	(W/K)	Conductance	G	($S=\Omega^{-1}$)
Thermal Resistivity	ρ	(K.m/W)	Electrical Resistivity	ρ	($\Omega.m$)
Thermal Conductivity	K	(W/K.m)	Electrical Conductivity	σ	(S/m)
Heat Capacity	C	(J/W)	Capacitance	C	(F=C/V)
Specific Heat	c_p	(J/Kg.K)	Permittivity	ϵ	(F/M)

Table 1.1: Analogy between Thermal and Electrical quantities

1.6 Outline of the Thesis

In this thesis, transient thermal simulations based on Multi Dimensional Model Reduction, followed by Eigen Decomposition (MDMR-ED) are presented.

- Chapter 2 provides a brief overview of model reduction techniques in general and the Krylov-based technique combined with the Arnoldi algorithm, used in this thesis.
- Chapter 3 presents the use of MDMR for transient thermal simulations. MDMR allows for model parameters (such as heat coefficient, h) to be incorporated explicitly into the reduced model. MDMR-ED is used to produce reduced models which are independent of the initial conditions as well as the boundary conditions applied to the model. This is of significant consequence as from a single MDMR one can perform many simulations of a detailed model under different BC's and initial conditions. In addition, the MDMR-ED technique offers a significant speed up when compared to other transient solvers.

- Chapter 4 presents the thermal transient analysis of electronic components when thermal ports are present. Thermal ports are the thermal links between a base-model and a component-model, allowing for heat exchange between them. MDMR is applied to every component-model, resulting in a reduced component-model, or macro-model. The macro-models are then linked to the base-model through defined multiple ports to make the large model solvable. The *mixed method* is presented as an efficient way to link the macro-models to the base-model, such that the BC's of both the macro-models and the base-model are all explicitly incorporated. A hierarchal MDMR can be performed on the mixed matrices, resulting in a very small model that can be easily solved. The above technique is boundary condition independent, and as such has distinct advantages over models derived from resistor networks in terms of accuracy, speed and flexibility.
- Chapter 5 presents conclusions and future work suggestions.

Chapter 2

Theoretical Background

This chapter provides a brief description of *Atar* the thermal simulator used to build the models in this work, followed by an overview of model reduction techniques. The Krylov-based model reduction, combined with the Arnoldi algorithm is the technique used in this thesis and is outlined in this chapter as well.

2.1 *Atar* Simulator

Atar was developed by the thermal modeling group at Carleton University [48]–[50]. *Atar* is a novel 3D thermal simulation tool for semiconductor integrated devices. The simulator is used to automatically generate a detailed 3D physical model of the device to be simulated from layout information. *Atar* produces an appropriate mesh of the device based on a rectangular block structure. A 3D model is then built up from the surface of the semiconductor, which represents both the substrate of the device and any metallization present. The 3D quad-tree mesh is created such that a fine mesh is produced around heat generation regions, but a moderate number of blocks is used

for the entire device. From the model, a network of thermal resistors and capacitors are extracted that mathematically represents the model.

Atar offers two SS solvers and a single transient solver. The first SS solver uses a direct solution of linear set of equations. It performs an LU decomposition and then solves the system. The second SS solver uses a Successive Over Relaxation (SOR) technique and is particularly useful for large or non-linear problems with temperature dependent material properties or BC's.

The transient solver is based on the TLM method, which is particularly suited to the extracted network approach described above. TLM modeling represents a physical model of heat flow as a sequence of voltage pulses traveling through a matrix network of transmission lines. By computing the voltage (temperature) at the nodes that connect the transmission lines, an explicit unconditionally stable solution to the heat flow equation can be found. The TLM method has a number of advantages over conventional numerical FDM/FEM techniques. As the algorithm is unconditionally stable for linear problems this leads to an ability to reduce execution time by optimizing the time step during the simulation. For example, for a step response the time step can be substantially increased as the simulation progresses towards steady-state [29], [48], leading to greatly reduced simulation times.

In this thesis, *Atar* is also used for defining thermal ports, creating component-models and forming mixed-models.

2.2 Model Reduction Techniques

Model order reduction techniques have been introduced in the circuit simulation literature as a way to deal with the large models. [41]–[46], [51]–[62]. The basic concept is that while these models contain a large number of poles, only a fraction of the poles has a significant impact on the circuit response. Model reduction takes advantage of this fact by approximating the original large system with a reduced one that captures the essential characteristics, or the dominant poles of the original system. In general, reduction methods can be classified into two main categories: 1) Direct moment matching based on Padé approximation 2) Indirect approximation based on finding the leading eigenvalues (with the largest magnitude).

2.2.1 Concepts of Moment Matching Techniques

A brief mathematical description of the underlying concepts of Moment Matching Techniques (MMT) is given below. Consider a single input/single output system and let $H(s)$ be the transfer function. $H(s)$ can be represented in a rational form as

$$H(s) = \frac{P(s)}{Q(s)} \quad (2.1)$$

where $P(s)$ and $Q(s)$ are polynomials in s . Equivalently, (2.1) can be written as

$$H(s) = c + \sum_{i=1}^{N_p} \frac{k_i}{s - p_i} \quad (2.2)$$

where p_i and k_i are the i^{th} pole residue pair, N_p is the total number of system poles, and c is the direct coupling constant. The time-domain impulse response can be

computed in a closed form using inverse Laplace transform as

$$h(t) = c\delta(t) + \sum_{i=1}^{N_p} k_i e^{p_i t} \quad (2.3)$$

In case of large networks, N_p , the total number of poles can be of the order of thousands. Computing all N_p poles will be highly CPU intensive even for a small network and for large networks it is completely impractical. Model-reduction techniques address the above issue by deriving a reduced-order approximation in terms of dominant poles, instead of trying to compute all the poles of a system. Assuming that only dominant poles were extracted, (2.1) can be rewritten to obtain approximate frequency/time responses, as

$$H(s) \approx \hat{H}(s) = \frac{\hat{P}(s)}{\hat{Q}(s)} = \hat{c} + \sum_{i=1}^L \frac{\hat{k}_i}{s - \hat{p}_i} \quad (2.4)$$

$$h(t) \approx \hat{h}(t) = \hat{c}\delta(t) + \sum_{i=1}^L \hat{k}_i e^{\hat{p}_i t}. \quad (2.5)$$

Note that $L \ll N_p$.

2.2.2 Moments of the Response

Consider the Taylor series expansion of the transfer function $H(s)$ at point $s = 0$

$$H(s) \approx \tilde{H}(s) = H(0) + s \frac{(H(0))^{(1)}}{1!} + s^2 \frac{(H(0))^{(2)}}{2!} + \dots + s^n \frac{(H(0))^{(n)}}{n!} \quad (2.6)$$

where the superscript (n) denotes the n^{th} derivative. Using a simpler notation, (2.6) can be rewritten as

$$H(s) \approx \tilde{H}(s) = m_0 + m_1 s + m_2 s^2 + \dots + m_n s^n = \sum_{i=0}^n s^i m_i, \quad m_i = \frac{H(0)^{(i)}}{i!} \quad (2.7)$$

The coefficients of Taylor series expansion (m_i) are also identical to the *time-domain moments* of the impulse response $h(t)$. This can be seen by using the Laplace transform of $h(t)$ [59]

$$\begin{aligned} H(s) &= \int_0^\infty h(t) e^{-st} dt = \int_0^\infty h(t) \left[1 - st + \frac{s^2 t^2}{2!} - \dots \right] dt = \\ &= \int_0^\infty h(t) dt + s \int_0^\infty (-1)t h(t) dt + s^2 \int_0^\infty \frac{t^2 h(t)}{2!} dt + \dots = \\ &= \sum_{i=0}^\infty s^i \left(\frac{(-1)^i}{i!} \int_0^\infty t^i h(t) dt \right) \end{aligned}$$

Due to the analogy, the coefficients of Taylor series expansion (m_i) are generally referred to as *moments*. It has been shown that the moments provide an estimation of delay and rise times [63], [64]. Elmore delay [63], which approximates the mid-point of the monotonic step response waveform by the mean of the impulse response, essentially matches the first moment of the response. This can be considered as one of the basic forms of approximation. However, in order to get accurate prediction of interconnect effects, it is essential that the reduced model must match (or preserve) as many moments as possible.

2.2.3 Direct Moment Matching

Direct Moment Matching Techniques [52]–[58] are based on finding a rational approximation of the impulse response of the system in the frequency domain, through the use of Padé approximation. This rational approximation is constructed so that it captures the dominant poles of the system and matches the behavior of the original impulse response. Once a rational function approximation of the circuit response is found, it can simply be evaluated in order to find the frequency domain response or it can be used to find the poles and residues of the system and generate the time domain response.

2.2.4 Padé Approximation

Consider a system transfer function $H(s)$ that is approximated by a rational function $\bar{H}(s)$ as

$$H(s) \approx \bar{H}(s) = \frac{a_0 + a_1s + a_2s^2 \cdots + a_Ls^L}{1 + b_1s + b_2s^2 \cdots + b_Ms^M} = \frac{P_L(s)}{Q_M(s)} \quad (2.8)$$

where $(a_0, \dots, a_L), (b_1, \dots, b_M)$ are the unknowns (total of $L+M+1$ variables). Using the Taylor series expansion of $H(s)$ at $s = 0$ we can write,

$$\frac{a_0 + a_1s + \cdots + a_Ls^L}{1 + b_1s + \cdots + b_Ms^M} = m_0 + m_1s + m_2s^2 + \cdots + m_{L+M}s^{L+M} \quad (2.9)$$

Cross multiplying and equating the coefficients of similar powers of s starting from s^{L+1} to s^{L+M} on both sides of (2.9), the denominator polynomial coefficients

$(b_1, \dots, b_M)$ can be evaluated as

$$\begin{pmatrix} m_{L-M+1} & m_{L-M+2} & \dots & m_L \\ m_{L-M+2} & \ddots & \vdots & m_{L+1} \\ \vdots & \dots & \ddots & \vdots \\ m_L & m_{L+1} & \dots & m_{L+M-1} \end{pmatrix} \begin{pmatrix} b_M \\ b_{M-1} \\ \vdots \\ b_1 \end{pmatrix} = - \begin{pmatrix} m_{L+1} \\ m_{L+2} \\ \vdots \\ m_{L+M} \end{pmatrix} \quad (2.10)$$

The numerator coefficients $(a_0, \dots, a_L)$ can be found by equating the remaining powers of s (from s^0 to s^L) as

$$\begin{aligned} a_0 &= m_0 \\ a_1 &= m_1 + b_1 m_0 \\ &\vdots \\ a_L &= m_L + \sum_{i=1}^{\min(L, M)} b_i m_{L-i} \end{aligned} \quad (2.11)$$

Once equations (2.10) and (2.11) are solved, all the necessary coefficients are known and the rational function (2.8) can be generated.

2.2.5 Single Expansion Moment Matching

Moment matching based on a single expansion point (normally at the origin of the complex plane) [53], [54] is referred to as Asymptotic Waveform Evaluation (AWE).

As an example of moment computation, consider the linear problem

$$\mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) = \mathbf{P}(t) \quad (2.12)$$

where $\mathbf{T} \in \Re^N$ is the vector of unknown node temperatures, $\mathbf{C} \in \Re^{N \times N}$ and $\mathbf{G} \in \Re^{N \times N}$ represent the thermal capacitance and conductance of the system, respectively. $\mathbf{P} \in \Re^N$ is the input vector and N is the size of the original system.

As a simple example of the structure of $\mathbf{C}$, $\mathbf{G}$ and $\mathbf{P}$ consider the circuit presented in fig. 2.1.

Figure 2.1: A two-node circuit

$$\mathbf{C} = \begin{pmatrix} c_1 & -c_1 \\ -c_1 & c_1 \end{pmatrix}; \quad \mathbf{G} = \begin{pmatrix} g_1 + g_3 & -g_3 \\ -g_3 & g_2 + g_3 \end{pmatrix}; \quad \mathbf{P} = \begin{pmatrix} J_1 \\ 0 \end{pmatrix}; \quad \mathbf{T} = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix}$$

where T_1 and T_2 are the unknown temperatures of the two nodes.

Premultiplying (2.12) by $\mathbf{G}^{-1}$ we have

$$\mathbf{G}^{-1}\mathbf{C}\frac{d\mathbf{T}(t)}{dt} + \mathbf{T}(t) = \mathbf{G}^{-1}\mathbf{P}(t) \quad (2.13)$$

In the Laplace domain it can be written as

$$s\mathbf{G}^{-1}\mathbf{C}\mathbf{T}(s) + \mathbf{T}(s) = \mathbf{G}^{-1}\mathbf{P}(s) \quad (2.14)$$

or

$$\mathbf{T}(s) = (\mathbf{I} + s\mathbf{G}^{-1}\mathbf{C})^{-1}\mathbf{G}^{-1}\mathbf{P}(s) \quad (2.15)$$

we define

$$\mathbf{A} = -\mathbf{G}^{-1}\mathbf{C} \quad (2.16)$$

$$\mathbf{R} = \mathbf{G}^{-1}\mathbf{P} \quad (2.17)$$

(2.15) can be written as

$$\mathbf{T}(s) = (\mathbf{I} - s\mathbf{A})^{-1}\mathbf{R} \quad (2.18)$$

Expanding (2.18) in terms of Taylor series

$$\begin{aligned} \mathbf{T}(s) &= (\mathbf{I} + s\mathbf{A} + s^2\mathbf{A}^2 \dots + s^q\mathbf{A}^q)\mathbf{R} \\ &= \sum_{k=0}^q s^k(\mathbf{A}^k\mathbf{R}) \\ &= \sum_{k=0}^q s^k m_k \quad \text{where } m_k = \mathbf{A}^k\mathbf{R}. \end{aligned} \quad (2.19)$$

The moments $\mathbf{M}_i$ (or the derivatives with respect to frequency) can be expressed

as

$$\begin{aligned}
\mathbf{M}_0 &= \mathbf{G}^{-1}\mathbf{P} = \mathbf{R} \\
\mathbf{M}_1 &= -\mathbf{G}^{-1}\mathbf{C}\mathbf{M}_0 = \mathbf{A}\mathbf{R} \\
\mathbf{M}_2 &= -\mathbf{G}^{-1}\mathbf{C}\mathbf{M}_1 = \mathbf{A}^2\mathbf{R} \\
&\vdots \\
\mathbf{M}_i &= -\mathbf{G}^{-1}\mathbf{C}\mathbf{M}_{i-1} = \mathbf{A}^i\mathbf{R}
\end{aligned} \tag{2.20}$$

Note that only one LU decomposition of $\mathbf{G}$ is required in order to calculate all the moments.

2.2.6 Limitations of Direct MMT Algorithms

Padé approximation based methods are limited by some fundamental properties of Padé approximation.

The major limitation is the ill-conditioning of Padé. As can be seen from (2.20), when successive moments are explicitly calculated, they are obtained as powers of $\mathbf{A}$. With the increasing values of i , this process quickly converges to an eigenvector corresponding to an eigenvalue of the largest magnitude [65]. As a result, for relatively large values of i , the explicitly calculated moments $m_i, m_{i+1}, \dots$ will not add any extra information to the moments, as all of them contain information only about the largest eigenvalue. In other words, the “moments beyond i ” are almost identical (or parallel to each other).

Other limitations of Padé approximation are:

1. Padé often produces unstable poles on the right-hand side of the complex plane.
2. Padé accuracy deteriorates as we move away from the expansion point.
3. Padé provides no estimates for error bounds.

In addition, there is no guarantee that the reduced-model obtained as above is passive. Passivity implies that a network cannot generate more energy than it absorbs, and no passive termination of the network will cause the system to go unstable [41], [44], [60]. The loss of passivity can be a serious problem because transient simulations of reduced networks may encounter artificial oscillations.

2.2.7 Multi-Point Moment Matching

For systems with many dominant poles, a single Padé expansion will not be able to capture all the required poles. For such systems, Complex Frequency Hopping (CFH) is used [58]. CFH extends the process of Moment Matching to multiple expansion points in the complex plane near or on the imaginary axis, and uses a binary search algorithm in order to determine an effective set of expansion points. Each expansion (hop) captures the nearby poles. In this way, enough information is obtained to enable the generation of an approximate transfer function. This function will match the original function up to a predefined frequency of interest. By using the information from all the expansion points, CFH ensures that all dominant poles are captured.

Fig. 2.2a presents the poles captured by a single point moment matching. The poles captured by a multi-point moment matching are presented in fig. 2.2b.

CFH provides clear bounds on the region of accuracy. This approach addresses most of the difficulties that a single Padé approximation presents.

The only remaining difficulty (for both single and multi-point expansions) is passivity. Direct MMT do not guarantee the passivity of the reduced model. The indirect MMT, which will be discussed in the next section, overcome this problem.

Figure 2.2: Poles captured a) using AWE b) using CFH

2.2.8 Indirect Moment Matching Techniques

In order to address the difficulties the limitations of the direct MMT present, indirect MMT were developed [41]–[46], [60]. These algorithms are based on Krylov-subspace techniques and congruent transformation.

Krylov subspace methods are based on the computation of the leading eigenvalues (those with the largest magnitude). By making use of numerically well conditioned algorithms such as the Arnoldi process, Krylov methods are much better behaved than the direct MMT.

Krylov methods can capture hundreds of poles at a single expansion (the direct methods are limited to 6-8 poles from a single expansion). Furthermore, while direct methods do not guarantee passivity, reduced order models obtained using Krylov subspace methods were proven to be passive [41], [60].

2.2.9 The Arnoldi Algorithm

The Arnoldi algorithm was suggested in the literature to overcome the ill-conditioning problem associated with the Padé approximation. This algorithm [65] first starts by computing a subspace which spans the same space as the moments of the system of equations (2.12). This subspace is called the Krylov-subspace and is denoted by $\kappa(A, R, q)$. In other words, any vector that is a linear combination of the moments of the system, can be expressed as a linear combination of the columns of $\kappa(A, R, q)$. This can be mathematically expressed as

$$\kappa(A, R, q) = \text{colsp}(\mathbf{K}) \tag{2.21}$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{M}_0 & \mathbf{M}_1 & \mathbf{M}_2 & \dots & \mathbf{M}_{q-1} \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{A}\mathbf{R} & \mathbf{A}^2\mathbf{R} & \dots & \mathbf{A}^{q-1}\mathbf{R} \end{pmatrix} \quad (2.22)$$

The matrix $\mathbf{K}$ by itself is likely to be ill-conditioned since the columns of $\mathbf{K}$ are formed based on the sequence $\mathbf{A}^i\mathbf{R}$, which quickly converges to the eigenvector of the largest eigenvalue [65]. To overcome this problem (which is the ill-conditioning problem), the Arnoldi algorithm computes another orthogonal matrix $\mathbf{Q}$ such that

$$\text{colsp}(\mathbf{Q}) = \text{colsp}(\mathbf{K}) \quad (2.23)$$

In contrast to the matrix $\mathbf{K}$, the matrix $\mathbf{Q}$ has the following advantages

1. $\mathbf{Q}$ is well conditioned.
2. $\mathbf{Q}$ is easy to invert since $\mathbf{Q}^{-1} = \mathbf{Q}^T$.
3. As many leading columns of $\mathbf{Q}$ as needed can be constructed to get an accurate solution.

The matrix $\mathbf{Q}$ can be computed using techniques such as the *modified Gram-Schmidt algorithm* [65]. A simple implementation of the Arnoldi algorithm which uses modified Gram-Schmidt is given below. [52], [61], [62]

function *Arnoldi*

```

 $\mathbf{q}_1 = \mathbf{R} / \|\mathbf{R}\|_2$  // Get the direction of  $\mathbf{q}_1$ 
for  $j = 1$  to  $k$  //  $k$  is the desired number of columns of the  $\mathbf{Q}$  matrix
     $\mathbf{z} = \mathbf{A}\mathbf{q}_j$ ;
    for  $i = 1$  to  $j$ 
```

```

         $h_{ij} = \mathbf{q}_i^T \mathbf{z};$ 
         $\mathbf{z} = \mathbf{z} - h_{ij} \mathbf{q}_i;$ 
    end for
     $h_{j+1,j} = \|\mathbf{z}\|;$  if  $h_{j+1,j} = 0$ , quit
     $\mathbf{q}_j = \mathbf{z}/h_{j+1,j};$ 
end for
end function

```

The matrix $\mathbf{Q}$ consists of the Arnoldi vectors, which are the columns $\mathbf{q}_j$ computed by the Arnoldi algorithm. The vectors $\mathbf{q}_j$ are computed one at a time, providing the advantage of computing only as many leading columns of $\mathbf{Q}$ as needed.

2.2.10 Model-Reduction through Arnoldi

After the orthonormal well-conditioned matrix $\mathbf{Q}$ is computed, a congruent transformation is performed on (2.12) and the resulting reduced system is expressed as

$$\hat{\mathbf{C}} \frac{d\hat{\mathbf{T}}(t)}{dt} + \hat{\mathbf{G}} \hat{\mathbf{T}}(t) = \hat{\mathbf{P}}(t) \quad (2.24)$$

where $\hat{\mathbf{C}}$ and $\hat{\mathbf{G}}$ represent the reduced capacitance and conductance matrices and $\hat{\mathbf{P}}$ is the reduced input vector obtained by

$$\hat{\mathbf{C}} = \mathbf{Q}^T \mathbf{C} \mathbf{Q}; \quad \hat{\mathbf{G}} = \mathbf{Q}^T \mathbf{G} \mathbf{Q}; \quad \hat{\mathbf{P}} = \mathbf{Q}^T \mathbf{P} \quad (2.25)$$

The size of the reduced system is much smaller than the original size, enabling the reduced system to be solved quickly and generate $\hat{\mathbf{T}}$ (the temperatures in the reduced space). $\hat{\mathbf{T}} \in \mathbb{R}^q$ and q is the size of the reduced vector of unknowns ($q \ll N$). The

solution in the reduced space is then mapped back to the original space, obtaining the full solution by

$$\mathbf{T}(t) = \mathbf{Q}\hat{\mathbf{T}}(t) \quad (2.26)$$

An indicator for the accuracy of the response of the reduced system is the total number of moments it can preserve (match) for a given order of reduction q . A Padé approximation of order q matches the first $2q$ moments [52]. However, an Arnoldi-based reduction of order q matches only the first q moments of the original system [41]. Essentially, this means that for a comparable accuracy, the reduced-model from Arnoldi will have double the size of the reduced model from direct Padé based approximation. On the other hand, since the Arnoldi approximation does not directly use the moments to obtain the reduced model, it does not suffer from the same numerical ill-conditioning that is associated with direct MMT. The accuracy of the Arnoldi approximation increases as the order q is increased since more moments of the original system are matched.

In an alternative approach, the reduction based on Lanczos algorithm can preserve double the moments ($2q$) when compared to Arnoldi for the same order of q [43], [59]. The difference between Arnoldi and Lanczos algorithms is that in Lanczos algorithm the matrix $\mathbf{A}$ is transformed to a tridiagonal matrix $\mathbf{T}_q$. In addition, Lanczos algorithm uses two biorthogonal Krylov-subspaces to recursively compute $\mathbf{T}_q$ [59]. However, reduced models formed using the Lanczos algorithm are not guaranteed to be passive.

In this thesis, the model order reduction process was performed using a Krylov-subspace method combined with the Arnoldi algorithm.

Chapter 3

Thermal Applications of Multi-Dimensional Model Reduction

3.1 Introduction

As discussed in section 2.2 several algorithms were proposed to efficiently simulate large systems of equations by model order reduction. However, all of them perform model reduction with respect to only one parameter (frequency).

Multi-Dimension Model Reduction (MDMR) is a reduction technique based on Krylov-subspace technique extended to multiple dimensions. It can form reduced models of large linear systems with respect not only to frequency but also to any other design parameters of the system.

The use of MDMR for thermal simulation can produce reduced models which are

independent of the boundary conditions as well as the initial conditions applied to the model (including external thermal resistance representing a mixed BC's). This result is highly important since it allows for multiple simulations (with different applied BC's) from a single model-reduction.

In this chapter, transient solution of different packages is obtained using the following procedure:

1. Generate the thermal network equations using *Atar*.
2. Perform an MDMR on the equations and produce a BICI transient reduced model.
3. Obtain the transient solution using Eigen Decomposition on the reduced model.
4. Map the reduced solution back to nodal temperatures and heat flows in the original *Atar* model.

This chapter proceeds as follows: Section 3.2 describes the use of *Atar* simulator to generate a thermal model with an explicit incorporation of the BC's in the discretized thermal equations. Section 3.3 outlines the theory behind the MDMR technique. Section 3.4 describes the Eigen-Decomposition process used to generate a transient solution of the reduced equations and section 3.6 presents the results of applying MDMR-ED to different models.

3.2 Numerical Generation of Thermal Network Equations

To solve for the conductive heat flow in an IC or package structure requires solving the following partial differential equation [66],

$$\nabla(\kappa(T)\nabla T) + g(x, y, z, t) = \rho C \frac{dT}{dt} \quad (3.1)$$

where $\kappa(T)$ is the temperature-dependent thermal conductivity (W/mK), C the thermal capacity of the material (J/kgK), ρ the material density (kg/m³) and $g(x,y,z)$ volumetric heat generation (W/m³). In this work, the temperature dependency of κ was not taken into consideration, and κ was set to be constant. Equation 3.1 can be written as

$$\kappa(\nabla^2 T) + g(x, y, z, t) = \rho C \frac{dT}{dt} \quad (3.2)$$

The first step in solving (3.2) is to discretize the model to be simulated. For this purpose, *Atar* which is a numerical simulator developed at Carleton University [29], [48], [50] was used. It is a thermal simulation tool that uses a technology description and layout/geometry information to automatically generate a full 3D model using an unstructured mesh. A thermal model of the structure consisting of resistors, capacitors and power sources is then extracted and the BC's applied.

For linear problems with temperature independent material properties, the thermal model of a structure of many blocks consisting of resistors, capacitors and power

sources can be expressed as:

$$\mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) = \mathbf{P}(t), \quad (3.3)$$

$\mathbf{C} \in \mathbb{R}^{N \times N}$ is the thermal capacitance matrix where N is the size of the model

$\mathbf{G} \in \mathbb{R}^{N \times N}$ is the thermal conductance matrix

$\mathbf{P} \in \mathbb{R}^N$ is the source vector representing internal power generation and applied boundary conditions

$\mathbf{T} \in \mathbb{R}^N$ is the vector of nodal temperatures in the model

Equation 3.3 has the boundary conditions implicitly expressed in the matrices $\mathbf{C}$, $\mathbf{G}$ and $\mathbf{P}$.

Mixed boundary conditions (which, in this work represent Robin boundary conditions) are applied to the model with the heat flow off the surface $P = h(T_s - T_f) + \beta$ specifying the local heat flow as a function of the surface temperature T_s . The parameters T_f (fixed BC's temperature), h (heat transfer coefficient) and β (outside heat flow) can all be specified for entire surface or for individual surface elements.

Atar in this work is used to build a detailed model, generate the thermal network equations as well as to numerically simulate and solve for the temperature in the detailed model. As previously mentioned, FDM/FEM programs such as *Atar* provide a very detailed description of the heat flow and temperature field in the model, suitable for detailed design, but are computationally intensive. Computation time increases dramatically with model size because of its dependence on the number of nodes (elements) in the model, limiting the size of a model and forcing model simplifications.

To overcome this barrier, the quick model building capability of *Atar* is combined with the MDMR method described in Section 3.3 creating a more practical design tool useful for simulation of larger models.

3.2.1 Explicit Temperature Boundary Conditions

For electronic components and packages the simple application of fixed temperatures on model surfaces does not appropriately capture the presence of external heat drops caused by convective and conductive cooling. When modeling a package it is often needed to use mixed BC's for some of the model surfaces. A convenient way to represent these mixed BC's is to use an external thermal resistance. The resistors address the relation between the model and the surrounding by connecting a model surface to a reference temperature, capturing the convective and conductive cooling. According to Newton's law of Cooling the thermal transport from a surface to the surrounding can be expressed in the form $P_s = hA_s(T_s - T_m)$ where P_s is the heat flow transferred from a surface at uniform temperature T_s to an ambient with a reference temperature T_m . h is the heat transfer coefficient and A_s is the surface area.

Fig. 3.1a shows a characterization of a surface BC's as a fixed surface temperature T_f and mixed heat flow represented by an external resistor and an ambient temperature T_m . A depiction of both fixed and mixed BC's present in a package model is shown in fig. 3.1b.

The basic philosophy of the model building in *Atar* is to create a complex mesh geometry using rectangular blocks of varying sizes. Fig. 3.2 shows the topology of a single characteristic block. At the center of each block is a temperature node. The internal heat flows between adjacent blocks are represented by thermal resistors

and the heat flow associated with the heat capacity of the block is represented by a capacitor (C_i). Thermal resistors are therefore connected from this node to nodes in all adjacent blocks. The value of R_{ij} , the resistor between nodes i and j , is calculated using the common cross-section area, the size of the two blocks and the local thermal conductivity. The thermal capacitance of the material is used to calculate the value of a capacitor connected from the block to the ground.

A number of boundary conditions can be placed on the model. Surfaces or regions of the model can be designated to be at a fixed temperature and mixed BC's can be applied to external surfaces. As shown in [67] the relation between the external thermal resistance (R_{ext}) and the coefficient of heat transfer from the surface (h) can be expressed as $R_{ext} = 1/(hA_s)$, [K/W]. This relationship is easily implemented in *Atar* by applying mixed BC's and allowing h to be used as a parameter in characterizing the external thermal resistance. It should be mentioned here that all the external resistances are assumed to be constant and no variation with surface temperature is taken into account. The block shown in Fig. 3.2 has two exposed surfaces. A mixed BC's has been applied to the top surface using an external resistance R_{ext} and an ambient temperature of T_m . A fixed temperature BC's has been applied to the front surface of the block and a fixed temperature node is present.

If the fixed and mixed BC's present in the model are parameterized using heat transfer coefficients and surface and ambient temperatures, *Atar* will produce the

Figure 3.1: Boundary Conditions

a) Fixed and mixed BC's applied on a surface b) Fixed and mixed BC's present in package model

Figure 3.2: A depiction of the thermal model of single *Atar* block. The block has a mixed BC's on the top face and fixed BC's on the front face. All other faces are connected to adjacent blocks.

matrices for the following equation which explicitly contains the BC's:

$$\begin{aligned} \mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) + \sum_{i=1}^n h_i \mathbf{A}_{h_i} \mathbf{T}(t) = \\ \mathbf{P}(t) + \sum_{i=1}^n h_i T_{m_i} \mathbf{A}_{s_i} + \sum_{k=1}^m T_{f_k} \mathbf{G}_{f_k} + \sum_{j=1}^l P_j \mathbf{V}_{p_j} \end{aligned} \quad (3.4)$$

$\mathbf{C} \in \mathbb{R}^{N \times N}$ is the thermal capacitance matrix where N is the size of the model

$\mathbf{G} \in \mathbb{R}^{N \times N}$ is the thermal conductance matrix

$\mathbf{P} \in \mathbb{R}^N$ is the source vector representing internal power generation

$\mathbf{T} \in \mathbb{R}^N$ is the vector of nodal temperatures

$\mathbf{A}_{h_i} \in \mathbb{R}^{N \times N}$ with elements $i \in (1, 2, \dots, n)$ represents the conductance of the mixed BC surface nodes, n is the number of mixed BC's

$\mathbf{A}_{s_i} \in \mathbb{R}^N$ with elements $i \in (1, 2, \dots, n)$ represents the power flow from mixed BC's surface nodes

$\mathbf{G}_{f_k} \in \mathbb{R}^N$ with elements $k \in (1, 2, \dots, m)$ represents the power flow from fixed BC's surface nodes, m is the number of fixed BC's

$\mathbf{V}_{p_j} \in \mathbb{R}^N$ with elements $j \in (1, 2, \dots, l)$ represents the input power of the nodes, l is the number of power sources

The mixed BC's surface nodes are selected using the matrices $\mathbf{A}_{h_i}$ and vectors $\mathbf{A}_{s_i}$. Each mixed Boundary Condition is characterized by heat transfer coefficient h_i and a fixed temperature T_{m_i} . Likewise, each fixed temperature surface is characterized by

a selection vector $\mathbf{G}_{f_k}$ and a temperature T_{f_k} . Individual source volumes are selected using the $\mathbf{V}_{p_j}$ vectors and a parameter P_j sets the input power in each heat source.

Although this formulation can be used to determine a transient response it suffers from a major drawback due to the large size of the matrices. Integrating this large set of equations over time requires large computational capacities and is time consuming. In the next section the Multi-Dimensional Model Reduction method is presented to solve this problem.

3.3 Multi-Dimensional Model Reduction

This section outlines the mathematical techniques used to perform the Multi-Dimensional Model Reduction of a linear thermal network. In order to find the multi-dimensional macro-model of (3.4) the *multi-dimensional subspace* should be evaluated. Next, the reduced system is diagonalized by performing an eigen decomposition. The poles and residues of the reduced systems are then used to generate the time domain solution in the reduced variable space.

3.3.1 Krylov-based Model Reduction Technique

Krylov-subspace based model reduction technique is an indirect moment matching technique that uses the fact that the time-domain behavior of a large linear system is governed mostly by its dominant poles. Dominant poles are the poles that are close to the imaginary axis and significantly influence the frequency characteristics of the system.

Krylov subspace methods are based on the computation of the leading eigenvalues

(those with the largest magnitude). By making use of numerically well conditioned algorithms such as the Arnoldi process, Krylov methods are much better behaved than the direct MMT (such as Pade approximation).

Krylov methods can capture hundreds of poles at a single expansion (the direct methods are limited to 6-8 poles from a single expansion). Furthermore, while direct methods do not guarantee passivity, *Atar* produces the $\mathbf{C}$ and $\mathbf{G}$ matrices such that $\mathbf{C}$ is non-negative definite and $\mathbf{G}$ is positive definite, so the reduced order models obtained using Krylov subspace methods are passive [41], [60].

Consider a model with one set of boundary conditions

$$\mathbf{C} \frac{d\mathbf{T}(t)}{dt} + (\mathbf{G} + h\mathbf{A}_h)\mathbf{T}(t) = \mathbf{P}(t) + hT_m\mathbf{A}_s \quad (3.5)$$

The matrices $\mathbf{C}$ and $\mathbf{G}$ represent the discretized thermal capacitance and conductance respectively. $\mathbf{P}$ is the excitation vector which includes any initial power in the model. The heat transfer coefficient h and the fixed temperature T_m represent a mixed boundary condition. The mixed boundary condition surface nodes are selected using the matrix $\mathbf{A}_h$ and vector $\mathbf{A}_s$.

The moments of $\mathbf{T}$ with respect to (w.r.t.) frequency (denoted by $\mathbf{Q}_s$) are computed using the procedure described in section 2.2. The moments of $\mathbf{T}$ w.r.t. h (where the k^{th} moment of $\mathbf{T}$ w.r.t. h is represented as $\mathbf{M}_h^k$) are computed using the equations:

$$\mathbf{D}\mathbf{M}_h^0 = \mathbf{F} \quad (3.6)$$

$$\mathbf{D}\mathbf{M}_h^1 = T_{m0}\mathbf{A}_s - \mathbf{A}_h\mathbf{M}_h^0 \quad (3.7)$$

$$\mathbf{D}\mathbf{M}_h^2 = -\mathbf{A}_h\mathbf{M}_h^1 \quad (3.8)$$

$$\vdots$$

$$\mathbf{D}\mathbf{M}_h^q = -\mathbf{A}_h\mathbf{M}_h^{q-1} \quad (3.9)$$

where $\mathbf{D} = \mathbf{G} + h\mathbf{A}_h$, $\mathbf{F} = \mathbf{P} + h_0T_{m0}\mathbf{A}_s$. h_0 , T_{m0} are the chosen expansion points of h and T_m and q is the number of moments to be preserved w.r.t. h .

$\mathbf{M}_h^0$ is the 0^{th} moment of the system w.r.t. h , and is identical to the 0^{th} moment of the system w.r.t. frequency, which is already included in the matrix $\mathbf{Q}_h$. The first moment w.r.t. h ($\mathbf{M}_h^1$) is computed explicitly, and the rest of the moments ($\mathbf{D}\mathbf{M}_h^2 - \mathbf{D}\mathbf{M}_h^q$) are computed using the Arnoldi algorithm.

The subspace of the moments of $\mathbf{T}$ w.r.t. h is given as

$$\text{colsp}(\mathbf{Q}_h) = \text{colsp} \begin{pmatrix} \mathbf{M}_h^1 & \mathbf{Q}_{Arnoldi} \end{pmatrix} \quad (3.10)$$

where

$$\mathbf{Q}_{Arnoldi} = \begin{pmatrix} \mathbf{M}_h^2 & \mathbf{M}_h^3 & \dots & \mathbf{M}_h^q \end{pmatrix} \quad (3.11)$$

Only one moment of $\mathbf{T}$ with respect to T_m exists. It is calculated by

$$\mathbf{D}\mathbf{M}_{T_m} = h_0\mathbf{A}_s \quad (3.12)$$

Once all the required moments are evaluated, the multi-dimensional subspace (de-

noted by $\mathbf{Q}$) is computed such that

$$\text{colsp}(\mathbf{Q}) = \text{colsp}(\mathbf{K}) \quad (3.13)$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{Q}_s & \mathbf{Q}_h & \mathbf{M}_{T_m} \end{pmatrix} \quad (3.14)$$

This can be achieved by using a standard QR decomposition [65].

Cross Derivatives are important when a change in a certain parameter of the system affects another. In this case, cross derivatives are needed to capture the dependency between those two parameters. To simplify, consider the system

$$\mathbf{Y}(s, h)\mathbf{T}(s, h) = \mathbf{P}(s) \quad (3.15)$$

where

$$\mathbf{Y}(s, h) = s\mathbf{C} + \mathbf{G}(h) \quad (3.16)$$

Cross derivatives are computed with respect to frequency (s) and heat coefficient (h) along with their cross terms using a recursive relationship [46]

$$\mathbf{Y}(s_i, h_j)\mathbf{M}^{(m,n)} = -\sum_{r=1}^n \frac{1}{r!} \sum_{q=0}^m \frac{\mathbf{Y}^{(q,r)}\mathbf{M}^{(m-q,n-r)}}{q!} - \sum_{q=1}^m \frac{\mathbf{Y}^{(q,0)}\mathbf{M}^{(m-q,n)}}{q!} \quad (3.17)$$

$$\mathbf{Y}(s_i, h_j)\mathbf{M}^{(0,0)} = \mathbf{P} \quad (3.18)$$

where $M^{(m,n)}$ is the m^{th} and n^{th} moment with respect to s and h respectively, s_i and h_j are the i^{th} and j^{th} expansion points of s and h and $Y^{(q,r)}$ is the q^{th} and r^{th} derivative with respect to s and h , evaluated at $s = s_i$ and $h = h_j$.

The multi-dimensional subspace that includes the cross derivatives (denoted by M_x) is

$$colsp(Q) = \begin{pmatrix} K & M_x \end{pmatrix} = \begin{pmatrix} Q_s & Q_h & M_{T_m} & M_x \end{pmatrix} \quad (3.19)$$

For the general case, where multiple BC's are applied to the model equation (3.5) can be written as

$$\begin{aligned} C \frac{dT(t)}{dt} + GT(t) + \sum_{i=1}^n h_i A_{h_i} T(t) = \\ P(t) + \sum_{i=1}^n h_i T_{m_i} A_{s_i} + \sum_{k=1}^m T_{f_k} G_{f_k} + \sum_{j=1}^l P_j V_{p_j} \end{aligned} \quad (3.20)$$

The next step after computing the moments with respect to frequency, is to evaluate the moments with respect to each heat coefficient h . Each heat coefficient represents a set of BC's applied to the reduced model. The advantage of MDMR over traditional model reduction techniques is that the reduction procedure is only performed once, generating a BCI reduced model. The boundary conditions are then explicitly incorporated into the reduced model. This allows for a quick calculation of the transient response for a variety of BC's for a single model.

The moments of T with respect to h_i (where the k^{th} moment of T w.r.t. h_i is represented as $M_{h_i}^k$) are computed using the equations:

$$\mathbf{DM}_{h_i}^0 = \mathbf{F} \quad (3.21)$$

$$\mathbf{DM}_{h_i}^1 = T_{m_{i0}} \mathbf{A}_{s_i} - \mathbf{A}_{h_i} \mathbf{M}_{h_i}^0 \quad (3.22)$$

$$\mathbf{DM}_{h_i}^2 = -\mathbf{A}_{h_i} \mathbf{M}_{h_i}^1 \quad (3.23)$$

$$\vdots$$

$$\mathbf{DM}_{h_i}^{q_{h_i}} = -\mathbf{A}_{h_i} \mathbf{M}_{h_i}^{q_{h_i}-1} \quad (3.24)$$

where $\mathbf{D} = \mathbf{G} + \sum_{i=1}^n h_{i0} \mathbf{A}_{h_i}$, $\mathbf{F} = \mathbf{P} + \sum_{i=1}^n h_{i0} T_{m_{i0}} \mathbf{A}_{s_i} + \sum_{k=1}^m T_{f_{k0}} \mathbf{G}_{f_k} + \sum_{j=1}^l P_{j0} \mathbf{V}_{p_j}$. $h_{i0}, T_{m_{i0}}, T_{f_{k0}}$ and P_{j0} are the chosen expansion points of h_i, T_{m_i}, T_{f_k} and P_j respectively and q_{h_i} is the number of moments to be preserved with respect to h_i . The subspace of the moments of $\mathbf{T}$ with respect to h_i is given as

$$\text{colsp}(\mathbf{Q}_{h_i}) = \text{colsp} \begin{pmatrix} \mathbf{M}_{h_i}^1 & \mathbf{Q}_{Arnoldi_i} \end{pmatrix} \quad (3.25)$$

where

$$\mathbf{Q}_{Arnoldi_i} = \begin{pmatrix} \mathbf{M}_{h_i}^2 & \mathbf{M}_{h_i}^3 & \dots & \mathbf{M}_{h_i}^q \end{pmatrix} \quad (3.26)$$

Note that the 0^{th} moment of all h_i 's is the same and is denoted by $\mathbf{M}_h^0$. It is already included as the first moment with respect to frequency in the matrix $\mathbf{Q}_s$. Moments of $\mathbf{T}$ with respect to T_{m_i}, T_{f_k} and P_j (for which only one moment exists) are computed using

$$\mathbf{DM}_{T_{m_i}} = h_{i0} \mathbf{A}_{s_i} \quad (3.27)$$

$$\mathbf{DM}_{T_{f_k}} = \mathbf{G}_{f_k} \quad (3.28)$$

$$\mathbf{DM}_{P_j} = \mathbf{V}_{p_j} \quad (3.29)$$

Once all the required moments are evaluated, the multi-dimensional subspace (denoted by $\mathbf{Q}$) is computed such that

$$\text{colsp}(\mathbf{Q}) = \text{colsp}(\mathbf{K}) \quad (3.30)$$

where,

$$\mathbf{K} = \begin{pmatrix} \mathbf{Q}_s & \mathbf{Q}_{h_1} & \dots & \mathbf{Q}_{h_n} & \mathbf{M}_{T_{m_1}} & \dots & \mathbf{M}_{T_{m_n}} \\ \mathbf{M}_{T_{f_1}} & \dots & \mathbf{M}_{T_{f_m}} & \mathbf{M}_{P_1} & \dots & \mathbf{M}_{P_l} \end{pmatrix} \quad (3.31)$$

In the case when cross-derivatives are required to preserve the network response the multi-dimensional subspace is modified to include the cross-derivatives (denoted by $\mathbf{M}_x$) as

$$\text{colsp}(\mathbf{Q}) = \text{colsp} \left(\begin{pmatrix} \mathbf{K} & \mathbf{M}_x \end{pmatrix} \right) \quad (3.32)$$

In general it is not essential to include cross-derivatives in-order to preserve the accuracy of typical thermal networks. It is to be noted that once computed, the multi-

dimensional subspace $\mathbf{Q}$ is a constant real matrix independent of frequency and heat coefficients. We then define:

$$\mathbf{T}(t) = \mathbf{Q}\hat{\mathbf{T}}(t) \quad (3.33)$$

where $\hat{\mathbf{T}}(t) \in \mathbb{R}^q$ and q is the size of the reduced vector of unknowns ($q \ll N$).

Using (3.33) a congruent transformation is performed on (3.20) to give

$$\begin{aligned} \hat{\mathbf{C}} \frac{d\hat{\mathbf{T}}(t)}{dt} + \hat{\mathbf{G}}\hat{\mathbf{T}}(t) + \sum_{i=1}^n h_i \hat{\mathbf{A}}_{h_i} \hat{\mathbf{T}}(t) = \\ \hat{\mathbf{P}}(t) + \sum_{i=1}^n h_i T_{m_i} \hat{\mathbf{A}}_{s_i} + \sum_{k=1}^m T_{f_k} \hat{\mathbf{G}}_{f_k} + \sum_{j=1}^l P_j \hat{\mathbf{V}}_{p_j} \end{aligned} \quad (3.34)$$

where,

$$\begin{aligned} \hat{\mathbf{C}} = \mathbf{Q}^T \mathbf{C} \mathbf{Q}; \quad \hat{\mathbf{G}} = \mathbf{Q}^T \mathbf{G} \mathbf{Q}; \quad \hat{\mathbf{A}}_{h_i} = \mathbf{Q}^T \mathbf{A}_{h_i} \mathbf{Q} \\ \hat{\mathbf{P}} = \mathbf{Q}^T \mathbf{P}; \quad \hat{\mathbf{A}}_{s_i} = \mathbf{Q}^T \mathbf{A}_{s_i}; \quad \hat{\mathbf{G}}_{f_k} = \mathbf{Q}^T \mathbf{G}_{f_k}; \quad \hat{\mathbf{V}}_{p_j} = \mathbf{Q}^T \mathbf{V}_{p_j} \end{aligned} \quad (3.35)$$

The size of the reduced network (3.34) is very small compared to the original network (3.20). The response at any node of the original network can be computed using (3.33) once the solution of the reduced system (3.34) is known.

It can be proven that the time-domain reduced network formed preserves the dominant eigen-values of the original network (3.20) [41]. In addition if the original network is passive, it can also be proven that the macro-model (3.34) is also passive [51].

3.4 Eigen Decomposition of the Reduced Multi-Dimensional System

This section is based on the fact that the poles of a system written as

$$\mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) = \mathbf{P}(t) \quad (3.36)$$

are the inverse of the eigenvalues of $\mathbf{A} = -\mathbf{G}^{-1}\mathbf{C}$. Since the Krylov-subspace Model Reduction captures the dominant poles of the original system, the time domain response can be calculated directly from the eigen values of the reduced system.

In this chapter, the Eigen Decomposition is performed on the equations

$$\begin{aligned} \hat{\mathbf{C}} \frac{d\hat{\mathbf{T}}(t)}{dt} + \hat{\mathbf{G}}\hat{\mathbf{T}}(t) + \sum_{i=1}^n h_i \hat{\mathbf{A}}_{h_i} \hat{\mathbf{T}}(t) = \\ \hat{\mathbf{P}}(t) + \sum_{i=1}^n h_i T_{m_i} \hat{\mathbf{A}}_{s_i} + \sum_{k=1}^m T_{f_k} \hat{\mathbf{G}}_{f_k} + \sum_{j=1}^l P_j \hat{\mathbf{V}}_{p_j} \end{aligned} \quad (3.37)$$

In order to simplify the computations in this section the total conductance matrix is defined as

$$\hat{\mathbf{G}}_t = \hat{\mathbf{G}} + \sum_{i=1}^n h_i \hat{\mathbf{A}}_{h_i} \quad (3.38)$$

and the total power vector

$$\hat{\mathbf{P}}_t = \hat{\mathbf{P}}(t) + \sum_{i=1}^n h_i T_{m_i} \hat{\mathbf{A}}_{s_i} + \sum_{k=1}^m T_{f_k} \hat{\mathbf{G}}_{f_k} + \sum_{j=1}^l P_j \hat{\mathbf{V}}_{p_j} \quad (3.39)$$

writing (3.37) in the frequency domain gives:

$$(s\hat{\mathbf{C}} + \hat{\mathbf{G}}_t)\hat{\mathbf{T}}(s) = \frac{\hat{\mathbf{P}}_t^0}{s} + \hat{\mathbf{T}}^0 \quad (3.40)$$

where $\hat{\mathbf{P}}_t(t) = \hat{\mathbf{P}}_t^0 \cdot u(t)$ (a step function power transition is modeled) and the original temperature distribution in the reduced space is $\hat{\mathbf{T}}^0$.

Solving for $\hat{\mathbf{T}}(s)$,

$$\hat{\mathbf{T}}(s) = (\hat{\mathbf{G}}_t + s\hat{\mathbf{C}})^{-1}(\frac{\hat{\mathbf{P}}_t^0}{s} + \hat{\mathbf{T}}^0)$$

or

$$\hat{\mathbf{T}}(s) = (I + s\hat{\mathbf{G}}_t^{-1}\hat{\mathbf{C}})^{-1}\hat{\mathbf{G}}_t^{-1}(\frac{\hat{\mathbf{P}}_t^0}{s} + \hat{\mathbf{T}}^0) = (I - s\mathbf{A}_t)^{-1}\hat{\mathbf{G}}_t^{-1}(\frac{\hat{\mathbf{P}}_t^0}{s} + \hat{\mathbf{T}}^0)$$

where I is the identity matrix and $\mathbf{A}_t = -\hat{\mathbf{G}}_t^{-1}\hat{\mathbf{C}}$. Using Eigenvalue decomposition $\mathbf{A}_t$ can be written as

$$\mathbf{A}_t = -\hat{\mathbf{G}}_t^{-1}\hat{\mathbf{C}} = \mathbf{V}\Lambda\mathbf{V}^{-1}$$

where $\Lambda = \text{diag}(\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_j)$. The final expression for $\hat{\mathbf{T}}(s)$ becomes

$$\begin{aligned} \hat{\mathbf{T}}(s) &= \mathbf{V}(I - s\Lambda)^{-1}\mathbf{V}^{-1}\hat{\mathbf{G}}_t^{-1}(\frac{\hat{\mathbf{P}}_t^0}{s} + \hat{\mathbf{T}}^0) = \\ &\mathbf{V} \begin{pmatrix} \frac{1}{1-s\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{1-s\lambda_j} \end{pmatrix} \mathbf{V}^{-1}\hat{\mathbf{G}}_t^{-1}(\frac{\hat{\mathbf{P}}_t^0}{s} + \hat{\mathbf{T}}^0) \end{aligned} \quad (3.44)$$

Every row in the vector $\hat{\mathbf{T}}(s)$ can then be expressed as a function of its poles and residues

$$\hat{T}(s) = c + \sum_{i=1}^j \frac{k_i}{s - P_i} + \sum_{i=1}^j \frac{k_i^0}{s - P_i}$$

where c is a constant, j is the number of poles P_i of the reduced model, k_i are residues associated with the input power $\hat{P}_t^0$ and k_i^0 the residues associated with the initial temperature distribution $\hat{T}^0$. Using these poles and residues the time domain solution of the reduced system can be easily calculated. The specific solution developed here was for a step response. The solution to other inputs (such as a pulse or a train of pulses) can be developed as well. Each row in the vector $\hat{T}(t)_i$ is of the form:

$$\hat{T}(t)_i = A_i - (a_{i1}e^{\bar{\lambda}_1 t} + a_{i2}e^{\bar{\lambda}_2 t} + \dots + a_{ij}e^{\bar{\lambda}_j t}) + (a_{i1}^0 e^{\bar{\lambda}_1 t} + a_{i2}^0 e^{\bar{\lambda}_2 t} + \dots + a_{ij}^0 e^{\bar{\lambda}_j t})$$

where $a_{i1} \dots a_{ij}$ and $a_{i1}^0 \dots a_{ij}^0$ represent the residues associated with the input power and initial temperature distribution respectively and $A_i = \sum_{m=1}^j a_{im}$. At steady-state the temperature is

$$\hat{T}(\infty)_i = A_i$$

and at $t = 0$

$$\hat{T}(0)_i = (a_{i1}^0 + a_{i2}^0 + \dots + a_{ij}^0) \equiv \hat{T}_i^0$$

This solution does not involve a use of a circuit simulator, but uses an explicit analytical solution.

The time domain solution of the original system is calculated by

$$T(t) = Q\hat{T}(t)$$

For each new set of BC's, the eigen decomposition of the reduced system is the only process that needs to be re-calculated. Since the reduced system is much smaller than the original one, this process is very quick. This method enables generating the time

domain solution for any BC's at any desired time point.

3.5 Computational Issues

The computational cost for this method consists of three major parts. The first is the LU decomposition, the second is the QR process and the third is the Eigen Decomposition of the reduced system. Table 3.1 presents the cost of each process.

Process	LU	QR	ED
Cost	n^α	$4q^2n$	q^3

Table 3.1: Computational cost

where n is the size of the original model, α is a constant of value 1.2-1.4 and q is the size of the reduced system. The order of magnitude of the overall cost depends on the sizes n and q . For large systems, where $q \ll n$, the overall cost is of order n . However, for applications that require a large q , the cost of the eigen decomposition of the reduced system becomes significant. For the models simulated in this chapter, the size of q was 12-20, while n was between 200-62,096. Overall, the cost was of order n .

3.6 Results

As mentioned earlier, the MDMR-ED (Multi-Dimensional Model Reduction - Eigen Decomposition) procedure used to obtain a transient response consists of the following steps:

1. Build the model in *Atar* and generate the thermal equations describing the model.

2. Reduce the thermal equations using MDMR.
3. Perform an Eigen Decomposition of the reduced model and use their poles and residues to calculate the transient response in the reduced space.
4. Map the solution back to nodal temperatures and heat flows in the original *Atar* model.

The MDMR-ED procedure was first tested on a set of five rectangular models with a single power source, a fixed temperature BC's surface and a mixed BC's surface. The rectangular models were $160 \times 160 \times 20 \text{ } (\mu\text{m}^3)$, with applied power of 0.05 W uniformly distributed over a region of $20 \times 8 \times 4 \text{ } (\mu\text{m}^3)$. The right side of the models was fixed at 300K and the mixed boundary conditions were applied on the bottom side as shown in fig. 3.3. Adiabatic boundary conditions ($h=0$) were applied to all the other surfaces of the model.

Figure 3.3: Dimension and boundary conditions

Figure 3.4: Different mesh grids
a) 200 block model b) 12,856 block model

Two primary concerns are to be addressed using this model, one is the accuracy of the MDMR-ED method and the second is the relative speed of the method compared to existing numerical techniques. To address the first concern an analytical solution of the heat flow in the model was derived and the MDMR-ED analysis compared to it.

Each of the five rectangular models consisted of a different number of nodes (elements) as a result of its mesh. The sizes of the models simulated were 200, 1,444, 12,856, 30,480 and 62,096 nodes. An example of the different meshes used is presented in Fig. 3.4a and b.

It can be expected that as the mesh becomes finer (number of blocks in the model increases), the discretization error will decrease.

Three different methods were used to generate the transient solution: TLM, MDMR-ED and the analytical one. The error between each two of them was calculated. The error was defined as

$$error = \frac{|T_1 - T_2|}{T_{ss}} \times 100 \quad (3.50)$$

where T_{ss} is the steady state temperature-rise and T_1 and T_2 represent the two methods compared.

The error between the TLM and the analytical solutions will be referred to as E_{TLM-A} . The error between the MDMR-ED and the analytical solutions is E_{MR-A} and the error between MDMR-ED and TLM solutions is E_{MR-TLM} .

The time dependent analytical solution for this problem can be developed [68] using a Green's function approach. This leads to the following expression for the temperature:

$$\begin{aligned}
T(x, y, z, t) = & \frac{g\alpha}{\kappa} \sum_{m=1}^{\infty} 1 - e^{-\alpha\beta_m^2 t} \left(\frac{2 \sin(\beta_m x)}{w_x} \right) \left(\frac{\cos(\beta_m x_1) - \cos(\beta_m x_2)}{\beta_m} \right) \\
& \left[\frac{y_2 - y_1}{w_y} + \sum_{n=1}^{\infty} 1 - e^{-\alpha\gamma_n^2 t} \left(\frac{2 \cos(\gamma_n y)}{w_y} \right) \left(\frac{\sin(\gamma_n y_2) - \sin(\gamma_n y_1)}{\gamma_n} \right) \right] \\
& \sum_{p=1}^{\infty} 1 - e^{-\alpha\eta_p^2 t} \left(\frac{2(\eta_p^2 + H^2)}{l(\eta_p^2 + H^2) + H} \right) (\cos(\eta_p(l - z))) \\
& \left(\frac{\sin(\eta_p(l - z_1)) - \sin(\eta_p(l - z_2))}{\eta_p} \right)
\end{aligned}$$

where w_x , w_y and l are the model dimensions, as shown in Fig. 3.3. κ is thermal conductivity, C is the thermal capacity of the material, ρ is the material density, $\alpha = \kappa/C\rho$, g is the power density over the region defined by the corner points (x_1, y_1, z_1) and (x_2, y_2, z_2) , $\beta_m = m\pi/w_x$, $\gamma_n = n\pi/w_y$, and η_p are the positive roots of the equation $\eta_p \tan(\eta_p l) = H$. H is defined by the heat transfer coefficient h divided by the thermal conductivity κ .

3.6.1 Determination of Model Parameters

Model parameters such as expansion point, number of moments and the necessity of cross derivatives have to be evaluated in the early stage of the transient analysis of a model since they directly affect the accuracy of the solution.

For this purpose the 62,096 block model was simulated numerous times, each with different values for those parameters, until an optimal value was found.

The transient responses presented in this chapter are those of the maximum temperature node in the model.

Expansion Points

When simulating a model with a wide range of heat coefficient values, choosing an expansion point for h while performing model reduction is crucial.

The effect of different expansion points on the accuracy of the transient response of the reduced model is demonstrated in fig. 3.5.

Figure 3.5: Transient response for different h values at three expansion points

In order to find the optimal expansion point, multiple points have to be examined. The model was simulated with various expansion points for three heat coefficient values. The purpose of this process was to determine which expansion point gives the most accurate transient response for the widest range of h values.

Fig. 3.5 presents the transient response of a single node simulated with three different expansion points ($h_{exp} = 10^{-12}$, 10^{-5} and 10^{-3}). These values were chosen since

Figure 3.6: Error measurements for $h = 10^{-12}$

Figure 3.7: Error measurements for $h = 10^{-5}$

Figure 3.8: Error measurements for $h = 10^{-3}$

the model was about to be simulated with different heat coefficient values ranging from 0 to 10^{-3} .

The model was simulated three times for each expansion point, each time with a different heat coefficient applied to it: $h_1 = 10^{-12}$, $h_2 = 10^{-5}$ and $h_3 = 10^{-3}$.

The analytical solution is presented in this figure as well. Next, the accuracy of each expansion point was measured, by simply comparing the response of each expansion point at each h to the analytical one.

Fig. 3.6 shows the error for $h = 10^{-12}$. The three responses (each generated with a different expansion point) were compared to the analytical one and the error was measured. It is clearly seen that the largest error for this case is when the expansion point is 10^{-3} (21.53%), where the values of the maximum errors for the two other expansion points (10^{-12} and 10^{-5}) are 0.645% and 2.03% respectively.

The results for the case where $h = 10^{-5}$ are shown in fig. 3.7. The largest error is when the expansion point is 10^{-12} (2.83%) and smallest when the expansion point is 10^{-5} (0.57%).

When $h = 10^{-3}$ the largest error is 2.9% for expansion point 10^{-12} , and the smallest is 1.06% when the point is 10^{-3} . Those results are shown in fig. 3.8.

From these simulations, it is clear that the response is most accurate when h has the value of the expansion point. The error increases as the evaluated h departs away from the expansion point.

Since the models were about to be simulated for a range of h values, an optimal expansion point had to be chosen. The optimal expansion point for the rectangular models was found to be around 10^{-5} . For this point, the error is minimal for the widest range of heat coefficient values.

Number of Moments

The number of moments with respect to each parameter is an important issue. Insufficient number of moments will yield an inaccurate result and too many moments will increase the simulation time.

Since heat coefficient (h) is applied to the model, moments with respect to h should be calculated, as well as moments with respect to frequency.

The 62,096 block model was simulated a few times with heat coefficient of $h = 10^{-10}$, with different combinations of number of moments w.r.t. frequency and w.r.t. h in order to determine the optimal number of overall moments. Each transient solution was then compared to the analytical one and the error was measured. The results are shown in fig. 3.9a and b.

a)

b)

Figure 3.9: Number of Moments

a) Transient response for different combinations of moments (5-4: 5 moments w.r.t. frequency and 4 w.r.t. h , 3-3: 3 moments w.r.t. each parameter, 1-1: a single moment w.r.t. to each parameter) b) Error measurements

Fig. 3.9a shows the transient response of a specific node on the model, generated with three different combinations of number of moments. The first combination is 1-1 which means 1 moment w.r.t. frequency and 1 w.r.t. h . The second combination is 3-3 (3 moments w.r.t. each parameter) and the last one is 5-4 (5 w.r.t frequency and 4 w.r.t h).

Fig. 3.9b shows the error of each combination w.r.t the analytical solution. From these two figures, it is seen that computing a low number of moments is not enough to capture the behavior of the original model. The error for the 1-1 combination is large, and reaches 3.78 % at SS. As the number of moments increases, the error decreases significantly. The maximum error for the 3-3 case is 2.46% and the SS error stabilizes on 0.11%, where in the 5-4 combination the maximum error drops to 0.62% and the SS error is 0.024% only.

The number of moments has a limit above which adding more moments will not contribute any more information about the original model. For the rectangular model simulated here, this combination was found to be 5-4. (5 moments w.r.t frequency and 4 w.r.t. h).

Cross Derivatives

As mentioned in section 3.3.1, cross derivatives will contribute to the response only when a change in a certain parameter will affect another. Cross derivatives will capture the dependency between those two parameters, and will be valuable to the solution. The simulations in the previous sections were done without cross derivatives. In this section, the effect of cross derivative on the transient solution will be discussed. Five cross derivatives were computed M_{h-f} (derivatives w.r.t heat coefficient and

a)

b)

Figure 3.10: Cross Derivatives

a) Transient response with and without cross derivatives b) Error measurements

frequency), $\mathbf{M}_{h-T_m}$ (derivative w.r.t. heat coefficient and T_m), $\mathbf{M}_{f-T_f}$ (derivative w.r.t. frequency and T_f), $\mathbf{M}_{f-T_m}$ (derivative w.r.t. frequency and T_m) and $\mathbf{M}_{f-P}$ (derivative w.r.t. frequency and P). The transient solution was generated with those derivatives and compared to the solution derived without them.

Fig. 3.10a presents the solution for two combinations of number of moments (1-1 and 5-4) with and without the cross derivatives. The error is presented in fig. 3.10b.

From the error graph, it is seen that the effect of the cross derivatives is smaller as the number of moments increases. The maximum error for the 1-1 combination is 0.12 % and for the 5-4 is 0.004 %.

Since the optimal number of moments was found to be 5 and 4 (w.r.t frequency and h , respectively) cross derivatives are not needed in this case.

The reduced system will result in high accuracy provided that the number of moments and the expansion points for the different heat coefficients are chosen carefully.

Initial Conditions and Boundary Conditions Variation

In this section, the effect of varying the initial conditions and the fixed BC's is presented. The 62,096 block model was simulated with the following conditions: the right side of the model was fixed at 300K, the rear side was fixed at 320K and mixed BC's were applied to the bottom side ($h = 10^{-3}$) as well as initial conditions of 320K. Diabatic boundary conditions were applied to all the other surfaces of the model. In order to verify the accuracy of the transient solution, the analytical solution for this problem was derived using the Green function approach [68]. This leads to the following expression for the temperature:

$$T(x, y, z, t) = T_0 + T_g + T_{BC} \quad (3.51)$$

where T_0 represent the contribution of the initial conditions on the temperature distribution, T_g is the contribution of the energy generation and T_{BC} is the contribution of the nonhomogeneous terms of the boundary conditions on the temperature.

$$\begin{aligned} T_0(x, y, z, t) = & F \sum_{m=1}^{\infty} e^{-\alpha \beta_m^2 t} \left(\frac{2 \sin(\beta_m x)}{w_x} \right) \left(\frac{1 - \cos(\beta_m w_x)}{\beta_m} \right) \\ & \left[1 + \sum_{n=1}^{\infty} e^{-\alpha \gamma_n^2 t} \left(\frac{2 \cos(\gamma_n y)}{w_y} \right) \left(\frac{\sin(\gamma_n w_y)}{\gamma_n} \right) \right] \\ & \sum_{p=1}^{\infty} e^{-\alpha \eta_p^2 t} \left(\frac{2(\eta_p^2 + H^2)}{l(\eta_p^2 + H^2) + H} \right) (\cos(\eta_p(l - z))) \\ & \left(\frac{\sin(\eta_p l)}{\eta_p} \right) \end{aligned}$$

$$\begin{aligned} T_g(x, y, z, t) = & \frac{g\alpha}{\kappa} \sum_{m=1}^{\infty} 1 - e^{-\alpha \beta_m^2 t} \left(\frac{2 \sin(\beta_m x)}{w_x} \right) \left(\frac{\cos(\beta_m x_1) - \cos(\beta_m x_2)}{\beta_m} \right) \\ & \left[\frac{y_2 - y_1}{w_y} + \sum_{n=1}^{\infty} 1 - e^{-\alpha \gamma_n^2 t} \left(\frac{2 \cos(\gamma_n y)}{w_y} \right) \left(\frac{\sin(\gamma_n y_2) - \sin(\gamma_n y_1)}{\gamma_n} \right) \right] \\ & \sum_{p=1}^{\infty} 1 - e^{-\alpha \eta_p^2 t} \left(\frac{2(\eta_p^2 + H^2)}{l(\eta_p^2 + H^2) + H} \right) (\cos(\eta_p(l - z))) \\ & \left(\frac{\sin(\eta_p(l - z_1)) - \sin(\eta_p(l - z_2))}{\eta_p} \right) \end{aligned}$$

$$\begin{aligned}
T_{BC}(x, y, z, t) = & \frac{T_f \alpha}{\kappa} \sum_{m=1}^{\infty} 1 - e^{-\alpha \beta_m^2 t} \left(\frac{2 \sin(\beta_m x)}{w_x} \right) \left(\frac{1 - \cos(\beta_m w_x)}{\beta_m} \right) \\
& \left[1 + \sum_{n=1}^{\infty} 1 - e^{-\alpha \gamma_n^2 t} \left(\frac{2 \cos(\gamma_n y)}{w_y} \right) (\cos(\gamma_n w_y)) \right] \\
& \sum_{p=1}^{\infty} 1 - e^{-\alpha \eta_p^2 t} \left(\frac{2(\eta_p^2 + H^2)}{l(\eta_p^2 + H^2) + H} \right) (\cos(\eta_p(l - z))) \\
& \left(\frac{\sin(\eta_p l)}{\eta_p} \right)
\end{aligned}$$

where w_x , w_y and l are the model dimensions, as shown in Fig. 3.3. κ is thermal conductivity, C is the thermal capacity of the material, ρ is the material density, $\alpha = \kappa/C\rho$, $\beta_m = m\pi/w_x$, $\gamma_n = n\pi/w_y$, and η_p are the positive roots of the equation $\eta_p \tan(\eta_p l) = H$, $H = h/\kappa$.

F is the initial condition function (in our case it is a constant of value 320), g is the power density over the region defined by the corner points (x_1, y_1, z_1) and (x_2, y_2, z_2) and T_f is the nonhomogeneous BC's (a constant of 320).

The model was simulated with different combinations of number of moments, and the error for each combination was measured. The transient responses and the error curves are presented in figs. 3.11a and b, respectively.

The first combination was 5 moments w.r.t frequency, 4 moments w.r.t h and no cross derivatives, as this combination was found to be optimal in section 3.6.1. This combination was referred to as 5-4-0. As clearly seen from fig. 3.11b, this combination results in a high error (12.2% in the beginning of the response and 3.15% at SS). The next combination was 7-8-4 (7 moments w.r.t. frequency, 8 w.r.t. h and 4 cross derivatives $\mathbf{M}_{h-T_m}$ w.r.t. h and T_m). This combination significantly reduced the

a)

b)

Figure 3.11: Initial conditions and BC's variation

a) Transient response for different combinations of moments b) Error measurements

error (both at $t = 10^{-10}$ and at SS), but the response was not accurate enough. Next, a combination of 5-1-40 was chosen (5 moments w.r.t. frequency, 1 w.r.t. h and 40 cross derivatives). The large number of cross derivatives and the low number of moments w.r.t. h were intentionally chosen in order to learn about the significance of cross derivatives in this case. Indeed, this combination results in a low error as seen in fig. 3.11b, but the cost is a relatively large reduced model. The optimal transient response (with the smallest error) was achieved when simulating the model with the 7-15-5 combination. This results both in a small reduced model and an accurate response. The peak error for this case is 1.53% at $t = 9.1 * 10^{-8}s$ and the SS error is only 0.1%.

These simulations show that both boundary conditions and initial conditions can be explicitly incorporated into the reduced model's equations using the MDMR-ED technique, resulting in a Boundary and Initial Conditions Independent (BICI) reduced model. It means that multiple simulations, with different initial and boundary conditions, can be performed from a single MDMR process. Only the Eigen Decomposition process has to be repeated for each new set of initial conditions or BC's.

Error due to Discretization

The effect of discretization on the accuracy of the transient response will be discussed in this section.

Since the size of the matrices and the vectors of 3.20 is based directly on the number of blocks the original model is divided into, the accuracy is expected to be higher as the mesh is finer.

The five models simulated in this chapter consisted of 200, 1,444, 12,865, 32,480

and 62,096 blocks. The difference in the block number between the lowest (200) and the highest (62,096) is large enough for us to learn about the dependency between the accuracy of the solution and the number of blocks in the model.

All five models were simulated with a heat coefficient $h = 10^{-6}$. The transient responses generated for each model were then compared to the analytical solution, and the error was measured.

Fig. 3.12a presents the transient solution calculated by MDMR-ED for the 200 and 62,096 block models (the smallest and the largest), and the analytical solution for a fixed value of h (10^{-6}). The effect of the discretization is clearly seen when examining the error presented in fig. 3.12b. The error between the two methods (analytical and MDMR-ED) in the 62,096 block model is significantly smaller than the one of the 200 block model. The peak error value for the 200 model is 2.519% where it is only 1.34% in the 62,096 one. The analytical steady-state temperature-rise is 28.1 K. It is 27.7 K for the 200 block model (error of 1.37%) and 28 K for the 62,096 block model (error of 0.24%). As expected, as the mesh becomes finer, the error decreases.

3.6.2 Rectangular Models - Heat Flow Variation

The five models were simulated with a wide range of h values, from 0 to 10^{-3} . Fig. 3.13a presents the transient solution for the 62,096 block model calculated by MDMR-ED compared with the analytical one for different values of h . As the value of the heat coefficient (h) increases, the steady-state temperature-rise decreases, until it reaches its lowest limit. For the five rectangular models the highest SS temperature-rise was 34.6 K achieved by $h = 0$ and the lowest limit was found to be 17.77 K, achieved by $h = 10^{-3}$. Simulating the model with values of h larger than 10^{-3} resulted in the same

a)

b)

Figure 3.12: 200 and 62,096 block model results

a) Temperature rise as a function of time for $h = 10^{-6}$ b) Error measurements

SS temperature-rise. Fig. 3.13b presents the error between the MDMR-ED solution to the analytical one for the various values of h . The SS error is smallest for $h = 0$ (0.008%) and largest (1.18%) for $h = 10^{-3}$. The peak error is largest when $h = 10^{-7}$ (2.05% at $t = 7.14 * 10^{-7} s$) and smallest when $h = 10^{-5}$ (0.58% at $t = 6.95 * 10^{-7} s$). The overall error is smallest when $h = 10^{-5}$. This was expected, since the expansion point chosen for this model was $h_{exp} = 10^{-5}$.

Fig. 3.14a presents a comparison of the transient solution of the three methods (TLM, analytical and MDMR-ED) for selected values of h for the 62,096 block model. Fig. 3.14b presents three error calculations for the case of $h = 10^{-6}$. The first curve presents the relative error between TLM and the analytical solution (E_{TLM-A}). The peak error value in this case is 1.306% (at $t = 3.8 * 10^{-7}$) and the SS error stabilizes on 0.25%. The second curve shows the error between the MDMR-ED and the analytical solution (E_{MR-A}). The peak error value is 1.34% (at $t = 7.7 * 10^{-7}$) and the SS error is 0.24%. Also shown is the error found between the TLM and MDMR-ED simulations (E_{MR-TLM}) which should not include the discretization error and should only contain the difference between the two numerical techniques and indeed the error is generally below 0.5%.

As discussed earlier, the heat coefficient (h) has an effect on the SS temperature-rise. For different values of h , the response stabilizes on different SS temperature-rise. The steady-state temperature-rise as a function of h is shown in Fig. 3.15, for the analytical case as well as the 200 block and 62,096 block models. It is clear that the 62,096 block model provides very accurate results and is almost identical to the analytical solution.

An important advantage of the MDMR-ED method is the ability to obtain the

a)

b)

Figure 3.13: Comparison of the analytical and MDMR-ED solutions for the 62,096 block model

a) Temperature rise as a function of time for different values of h b) Error measurements

a)

b)

Figure 3.14: Comparison of Analytical, TLM and MDMR-ED solutions for the 62,096 block model

a) Temperature rise as a function of time for different values of h b) Error measurements for $h = 10^{-6}$

Figure 3.15: Steady State temperature rise as a function of h

temperature distribution in the model at any desired time point. As an example, a temperature contour plot of the 62,096 block model at SS is shown in Fig. 3.16.

Figure 3.16: Temperature contour plot of the 62,096 block model

Simulation Efficiency

The main advantage of the MDMR-ED is the speed up it offers. Table 3.2 presents the model sizes (original and reduced), the simulation times for *Atar* (TLM) and MDMR-ED methods for the five models. For these results the TLM time stepping algorithm was used very aggressively to reduce the simulation time [29]. To allow for a fair comparison the TLM time step was increased as quickly as possible during the simulation and still allow for response accurate to within 2%. These results indicate that the MDMR-ED simulations are very accurate and that the TLM time stepping algorithm is being used as aggressively as possible to decrease the simulation time.

The primary advantage of the MDMR-ED method presents itself when a number of simulation with different BC's are performed on a single model. The total speed up

Model size	MDMR Model Size	Atar (TLM) t_{TLM}	MR t_{MR}	ED t_{ED}	speed up at $n = 300$
200	12	0.05 m	0.02 m	0.27 m	0.18
1,444	12	0.7 m	0.042 m	0.31 m	2.25
12,856	12	37 m	1.85 m	0.34 m	106.9
32,480	12	100 m	6.33 m	0.38 m	249.3
62,096	12	190 m	72 m	0.42 m	287.6

Table 3.2: Simulation times for rectangular models. All runs were done on a SUN-Blade 100 workstation

Figure 3.17: Speed Up as a function of number of simulations

(S) is a function of the number of desired simulations at different BC's and is defined as:

$$S = \frac{nt_{TLM}}{t_{MR} + nt_{ED}} \quad (3.52)$$

where n is the number of simulations. If we examine the 32,480 block case, we see that it takes *Atar* using TLM 100 m per simulation. The simulation time of the MDMR-ED consists of two parts: the MDMR process (t_{MR}) and the Eigen Decomposition one(t_{ED}). The MDMR takes 6.33 m and it is only performed once, while the eigen decomposition takes 0.38 m and is performed once per simulation of a BC's set. Fig. 3.17 presents the speed up as a function of the number of simulations for the rectangular models. For the smallest model (200 blocks) the speed up is essentially a constant line of a value less than 1. For this case the TLM solver is actually faster than the MDMR-ED and the simulation time is dominated by the time needed for the eigen decomposition. For the 1,444 block model the speed up is a constant line of a value slightly larger than 1. For the other three models the speed-up increases dramatically as the number of simulations is increased.

From this plot it can be seen that as the number of simulations increases, t_{MR} becomes negligible, and the speed up approaches a limit, which is simply the TLM simulation time divided by the time of the eigen decomposition process

$$\lim_{n \rightarrow \infty} S = \lim_{n \rightarrow \infty} \frac{nt_{TLM}}{t_{MR} + nt_{ED}} = \frac{t_{TLM}}{t_{ED}} \quad (3.53)$$

For the 1,444 block model the value is $\frac{0.7}{0.31} = 2.26$, for the 12,856 block model the value is $\frac{37}{0.34} = 108.8$, for the 32,480 block model the value is $\frac{100}{0.38} = 263.16$ and for the

62,096 block model it is $\frac{190}{0.42} = 452.38$. As expected, this limit becomes larger as the number of blocks in the model increases.

3.6.3 CBGA Model

The MDMR-ED was evaluated by simulating a complex electronic package model. A C4/CBGA package (schematic shown in Fig. 3.18a). Thermal simulation of very detailed models of C4/CBGA packages which include all the package elements is important to control package junction temperature and determine the characteristic thermal resistance of the package. The entire model incorporates three main parts: a package substrate ($33 \times 33 \times 0.86 \text{ mm}^3$), a silicon digital integrated circuit ($20 \times 20 \times 0.356 \text{ mm}^3$), and an aluminum cap. In the model shown, the ceramic BGA substrate consists of four ceramic layers, two signal layers and two ground layers. The package-to-board interconnection is an array of 1024 quadratic solder balls on 1 mm pitch. In order to take into account the important role of substrate thermal vias 576 thermal vias are included [69]. A total power of 10 W dissipated is assumed to be uniformly spread over a thin layer close to the top side of the chip. The I/O count is 4096 with a C4 grid pitch of $300 \mu\text{m}$ and bump size $200 \mu\text{m}$. A partial view of the model structure of C4/CBGA package is shown in Fig. 3.18b. The size of the *Atar* model was 53,976 blocks.

The BC's for this model include a uniform temperature at the bottom of the package (the package/board contact surface) and mixed boundary conditions characterized at the top. Adiabatic boundary conditions were applied to all the other surfaces of the model. The optimal expansion point of h was found to be around 10^{-6} in a process similar to that described in section 3.6.1.

a)

b)

Figure 3.18: Model of a low-profile capped CBGA
a) Schematic b) Sectional view showing the sublayer and other details

Fig. 3.19a presents the transient response of the CBGA model for a wide range of h values, from 0 to 10^{-5} . The MDMR-ED results as well as the TLM ones are presented. The highest SS temperature-rise is reached when $h = 0$ and is 18.7 K. The lowest SS temperature-rise is 8.9 K when $h = 10^{-5}$. Simulating the model with a heat coefficient higher than 10^{-5} will result in the same transient response. The error between MDMR-ED and TLM solutions was calculated for each simulation and is presented in Fig. 3.19b. The peak value of the error is received when $h = 10^{-8}$ and is 3.23% at $t = 0.014s$. When the model reaches steady-state, the error (for all h values) stabilizes on values below 1%.

As in the rectangular models case, MDMR-ED was found to be much faster than the TLM method. Table 3.3 presents the sizes of the original as well as the reduced model and the simulation times. A single simulation takes Atar 90 m. The MDMR process takes 27 m and each eigen-decomposition is another 0.4 m. Using eq. (3.52) the speed up is $S = 90n/(27 + 0.4n)$. As n (number of simulations) goes to infinity, the speed up reaches its upper limit which is $S_{\infty} = 90/0.4 = 225$. Fig. 3.20 presents the speed up for the CBGA model as a function of the number of simulations.

Model size	MDMR Model Size	Atar (TLM) t_{TLM}	MR t_{MR}	ED t_{ED}	speed up at $n = 300$
53,976	14	90 m	27 m	0.4 m	183.6

Table 3.3: Simulation times for CBGA model

The steady-state temperature-rise as a function of h is presented in Fig. 3.21. Both TLM and MDMR-ED results are shown and the accuracy of the MDMR-ED method can be seen to be very good.

The CBGA model was simulated with three different heat transfer coefficients applied to the top side (h_1), rear side (h_2) and right side (h_3) of it. The model was

a)

b)

Figure 3.19: CBGA model results

a) Temperature rise as a function of time for different values of h b) Error between TLM and MDMR-ED results

Figure 3.20: CBGA Speed Up

Figure 3.21: SS Temperature rise as a function of h

a)

b)

Figure 3.22: CBGA model with three h 's
 a) Temperature rise as a function of time b) Error between TLM and MDMR-ED solutions

b)

Figure 3.23: CBGA model contour plots
a) At half the Steady State temperature b) At Steady State

simulated multiple times, with different sets of h values. The results are shown in Fig 3.22a. The highest SS temperature-rise was reached when all heat coefficients were set to zero and is 18.5 K. Fig. 3.22b shows the error calculations for three simulations. The maximum error is received when $h_1 = 10^{-8}$, $h_2 = 10^{-7}$ and $h_3 = 10^{-6}$ and is 3.32% (at $t = 0.016s$). The lowest maximum error is achieved for $h_1 = 10^{-5}$, $h_2 = 10^{-6}$ and $h_3 = 10^{-7}$ and is 2.72% (at $t = 0.014s$).

Finally, as the MDMR-ED method is able to generate the temperature at all nodes at any specified time point Fig. 3.23a presents a temperature contour plot of the CBGA model at $t = 2 * 10^{-2}s$, which is the time at which the maximum temperature is around half of the SS value. The maximum temperature for this time point is 309.67K. The temperature contour plot of the steady-state is also shown in Fig. 3.23b. The maximum temperature of the steady-state is 318.5K.

3.7 Summary

A new approach for thermal transient analysis is introduced in this chapter. This approach, MDMR-ED, makes use of the MDMR technique followed by an Eigen Decomposition of the reduced model. The solution is then obtained by using the poles and residues of the reduced model. The reduced solution is easily mapped back to the original model by a transformation matrix.

By exploiting the multi-dimensional nature of the model reduction, BC's parameters as well as initial conditions, were incorporated explicitly into the reduced model, resulting in a boundary and initial conditions independent model (BICI). This allowed to perform multiple simulations with different sets of BC's applied to the reduced

model.

The MDMR-ED solution is much faster than the TLM solution it was compared to. The speed up was found to be dependent on the model size and the number of performed simulations. As the complexity of the model increases, the speed up was larger.

The transient solution generated by the MDMR-ED is very accurate. It was compared both to the analytical solution and to the TLM solution of the five rectangular models. The solutions obtained by the three methods, as well as the errors between them, were presented. MDMR-ED was also applied to a complex model of an electronic package (CBGA). The MDMR-ED solution was compared to the TLM solution, and was found to be accurate and much faster (a speed up of 225 was achieved).

A key feature of the MDMR-ED method is its capability of providing complete temperature distribution at any time point of the simulation, as shown in Fig. 3.23a and b.

The main limitation of this method is presented by the LU decomposition performed during the MDMR process. If the model size is too large, the LU process takes a long time, and sometimes even fails. The next chapter presents the creation of macro-models as a solution to this problem.

Chapter 4

Transient Macro-models for Electronic Components Using Model Reduction

4.1 Introduction

In this chapter a methodology to significantly improve the accuracy, flexibility and efficiency of transient thermal macro-models of devices and modules is presented.

The systems simulated in this chapter consist of a detailed “base-model” and one or more “component-models” linked to it. The base-model and the component-models were built using *Atar*. An important aspect to be considered in the definition of a component-model is its interface with a detailed base-model. The interface is formed of thermal ports which allow heat exchange between the base-model and the component-models. MDMR is performed on the component-models, resulting

in BICI macro-models. The *mixed method* is presented as an efficient way to link macro-models to a base-model. The mixed equations, therefore, consist of the detailed base-model equations and the macro-model ones and can be very large. To allow for the direct calculation of the transient response a second MDMR is performed on the mixed matrices, to create a very small system of equations. Once the reduced mixed matrices are generated (the size of this system is usually less than 20), the transient solution in the reduced space is easily obtained using Eigen Decomposition. The last step is to map the reduced solution back to nodal temperatures and heat flows in the original base-model and component models.

The major motivations of the proposed methodology can be summarized as follows:

1. The boundary conditions of both the base-model and the component-model are explicitly expressed, so the mixed model is BCI. The mixed method can handle boundary conditions variations both from the base-model and the component-model.
2. Hierarchical model reduction of the mixed model is feasible, resulting in a very small mixed model that can easily and quickly be solved. The solution is then mapped back to the original mixed-model by using transformation matrix $\mathbf{Q}_m$. Then, by using the transformation matrix $\mathbf{Q}_c$, all the internal temperatures of the original component-model can be achieved.

The process used to generate the transient solution consists of the following steps:

1. Generate the base-model and component-model thermal network equations using *Atar* for a particular port definition.

2. Perform an MDMR on the equations describing the component to produce a BCI transient macro-model.
3. Connect the macro-model to the base-model generating the thermal network describing the mixed-model.
4. Perform a model reduction on the mixed matrices to produce a very small system of equations.
5. Calculate the transient solution using Eigen Decomposition on the reduced mixed matrices.
6. Map back the thermal results to the original base-model and component-model nodal structure.

This chapter proceeds as follows: Section 4.2 provides thermal ports definition as well as port equations. Section 4.3 outlines the theory behind the MDMR technique applied to multi-port systems. Section 4.4 describes the process of forming the mixed-matrices. Section 4.5 presents the eigen-decomposition process used to generate a transient solution of the reduced mixed matrices and section 4.7 presents the results of applying the proposed methodology to different models.

4.2 Transient Thermal Macro-models

4.2.1 Macro-model Definition

When creating a macro-model, thermal ports must be designated on the interface or contact surface between the component-model and the base-model. These ports

allow for thermal links between models, making it possible for the macro-model to exchange heat with the base model. In Fig. 4.1 and 4.2 the procedure for taking a fully detailed model of the base and component and separating it into two pieces – a component-model (to be macro-modeled) and a base-model – is shown. The figures show how ports are defined between the two models and used to connect them. The configuration of the ports must be specified at the creation of the component-model. Once a configuration is determined, the component-model can be built (by *Atar*). If a new port configuration is desired, the model has to be re-built. The port configuration is very important in determining the behavior of a macro-model. Ports essentially impose isothermal regions on the interface between the component-model and the base-model. This constraint causes an error in the solution of the mixed-model when compared to a fully detailed model of both parts. Different port geometries will be investigated in this work. The simplest port definition is a port-per-block (PPB) configuration where each surface block in the component-model is defined to be a port to the base-model, as presented in fig. 4.1a.

The primary advantage of this configuration is that solution to the mixed-model will be identical to that of the full detailed model. The main disadvantage is that often such a definition produces a large number of ports which will make the macro-model cumbersome to use.

Other port configurations can be defined by using more than one surface block per port. In this case, surface nodes within the port for both the base-model and the component-model are at a common temperature (the port temperature to which they belong). This, as remarked above, will produce an error in the transient response. The error that each port configuration presents can be measured by simply comparing

Figure 4.1: Component-model linking to a base-model
a) Fully detailed model of both parts (or a port per block configuration) b) 2 port configuration

Figure 4.2: Macro-model linking to a base-model

a) Macro-model representation of a two port model b) A two port macro-model linked to a base-model

the response to the PPB response. A two port configuration is presented in fig. 4.1b. In this case, each block of the component-model as well as each block of the base-model is connected to one of the two ports. Each port has a certain temperature, and all the blocks connected to it are at this temperature.

Mathematically a port is characterized by its temperature $T_p(t)$ and its heat flow $I_p(t)$, analogous to port voltage and port current in an interconnected electrical macro-model. In fig. 4.2a. T_{p1} and T_{p2} represent the first and second ports' temperatures, respectively. I_{p1} and I_{p2} are the ports' heat flows. Fig. 4.2b shows a two port macro-model linked to a base-model.

4.2.2 Macro-model Equations

The basic equations for the situation when a component-model is connected to a base-model through ports are,

$$\mathbf{C}_b \frac{d\mathbf{T}_b(t)}{dt} + \mathbf{G}_b \mathbf{T}_b(t) + \mathbf{L}^T \mathbf{I}_p = \mathbf{P}_b(t) \quad (4.1)$$

$$\mathbf{T}_p = \mathbf{L} \mathbf{T}_b \quad (4.2)$$

$$\mathbf{C}_c \frac{d\mathbf{T}_c(t)}{dt} + \mathbf{G}_c(\mathbf{h}) \mathbf{T}_c(t) = \mathbf{P}_c(\mathbf{h}, t) + \mathbf{M}_p \mathbf{T}_p \quad (4.3)$$

$$\mathbf{I}_p = \mathbf{M}_p^T \mathbf{T}_c \quad (4.4)$$

$\mathbf{T}_b \in \mathbb{R}^{n_b}$, $\mathbf{T}_c \in \mathbb{R}^{n_c}$ are the vectors of block temperature of the base-model and component-model respectively. n_b and n_c are the number of blocks in the base-model and the component-model

$\mathbf{C}_b \in \mathbb{R}^{n_b \times n_b}$, $\mathbf{C}_c \in \mathbb{R}^{n_c \times n_c}$ are the thermal capacitance matrices of the base-model and component-model respectively

$\mathbf{G}_b \in \mathbb{R}^{n_b \times n_b}$, $\mathbf{G}_c \in \mathbb{R}^{n_c \times n_c}$ are the thermal conductance matrices of the base-model and component-model respectively

$\mathbf{P}_b \in \mathbb{R}^{n_b}$, $\mathbf{P}_c \in \mathbb{R}^{n_c}$ are the source vectors representing internal power generation and applied boundary conditions of the base-model and component-model, respectively

$\mathbf{I}_p, \mathbf{T}_p \in \mathbb{R}^{n_p}$ denote the port heat flow and port temperature respectively. n_p is the number of ports

$\mathbf{L} = (l_{i,j})$ with elements $l_{i,j} \in (0, 1)$ where $i \in (1, 2, \dots, n_p)$, $j \in (1, 2, \dots, n_b)$ with a maximum of one non-zero in each row or column is a selector matrix that connects the component-model port currents to the base-model

$\mathbf{M}_p = (m_{i,j})$ with elements $m_{i,j} \in (0, 1)$ where $i \in (1, 2, \dots, n_c)$, $j \in (1, 2, \dots, n_p)$ is a selector matrix that maps the port heat flow onto the component-model

The equations above describe a fully detailed base-model and a fully detailed component-model connected by ports and the simulation of this system of equations would be no faster than that of the original model. Forming a macro-model of the component-model using MDMR is the next step.

In order to perform MDMR on (4.3) and (4.4), the *multi-dimensional subspace* needs to be evaluated. The details of this procedure can be found in the next section.

4.3 Model Reduction Technique - Theoretical Approach

This section outlines the mathematical technique used to perform the multi-dimensional model reduction (MDMR) of a multi-port linear thermal network. It is an extension of section 3.3 to the multi-port case.

Consider the multi-port system

$$\mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) = \mathbf{P}(t) + \mathbf{M}_p \mathbf{T}_p \quad (4.5)$$

where N is the number of nodes in the system, $\mathbf{C} \in \mathbb{R}^{N \times N}$, $\mathbf{G} \in \mathbb{R}^{N \times N}$, $\mathbf{T} \in \mathbb{R}^N$ and $\mathbf{P} \in \mathbb{R}^N$. $\mathbf{M}_p \in \mathbb{R}^{N \times n_p}$ and n_p is the number of ports. Premultiplying (2.12) by $\mathbf{G}^{-1}$ we have

$$\mathbf{G}^{-1} \mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{T}(t) = \mathbf{G}^{-1} \mathbf{P}(t) + \mathbf{G}^{-1} \mathbf{M}_p \mathbf{T}_p \quad (4.6)$$

In the Laplace domain it can be written as

$$s\mathbf{G}^{-1} \mathbf{C} \mathbf{T}(s) + \mathbf{T}(s) = \mathbf{G}^{-1} \mathbf{P}(s) + \mathbf{G}^{-1} \mathbf{M}_p \mathbf{T}_p \quad (4.7)$$

or

$$\mathbf{T} = (\mathbf{I} + s\mathbf{G}^{-1} \mathbf{C})^{-1} (\mathbf{G}^{-1} \mathbf{P}(s) + \mathbf{G}^{-1} \mathbf{M}_p \mathbf{T}_p) \quad (4.8)$$

we define

$$\mathbf{A} = -\mathbf{G}^{-1}\mathbf{C} \quad (4.9)$$

$$\mathbf{R} = \mathbf{G}^{-1}\mathbf{M}_p \quad (4.10)$$

The block moments of $\mathbf{T}$ w.r.t. frequency are defined as the coefficients of Taylor expansion of $\mathbf{T}$ around $s=0$

$$\mathbf{T}_{(s)} = \mathbf{M}_0 + \mathbf{M}_1 s + \mathbf{M}_2 s^2 + \dots \quad (4.11)$$

where $\mathbf{M}_i \in \mathbb{R}^{N \times n_p}$. These block moments can be computed using the relation

$$\mathbf{M}_i = \mathbf{A}^i \mathbf{R} \quad (4.12)$$

The block Krylov space generated by the matrices $\mathbf{A}$ and $\mathbf{R}$ is denoted as $k(\mathbf{A}, \mathbf{R}, q)$. Any vector that is a linear combination of the moments of the system can be expressed as a linear combination of the columns of $k(\mathbf{A}, \mathbf{R}, q)$.

$$k(\mathbf{A}, \mathbf{R}, q) = \text{colsp}(\mathbf{K}) \quad (4.13)$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{R} & \mathbf{A}\mathbf{R} & \mathbf{A}^2\mathbf{R} & \dots & \mathbf{A}^{k-1}\mathbf{R} & \mathbf{A}^k\tilde{\mathbf{R}} \end{pmatrix} \quad (4.14)$$

The terms $\mathbf{R}, \mathbf{A}\mathbf{R}, \mathbf{A}^2\mathbf{R} \dots \mathbf{A}^i\mathbf{R}$ are the block moments of the system (the derivatives of $\mathbf{T}$ with respect to frequency). where $\tilde{\mathbf{R}} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \dots \ \mathbf{r}_l]$, $k = \lfloor q/n_p \rfloor$ and $l =$

$q - kn_p$. The operator $[\cdot]$ represents the truncation to the nearest integer, towards zero. For example, assume a three-port characterization involving $\mathbf{A} \in \Re^{50 \times 50}$ and let $q = 14$. In this case, $k = [14/3] = 4$, $l = 14 - 4 \times 3 = 2$ and (4.14) can be written as

$$\mathbf{K} = \begin{pmatrix} \mathbf{R} & \mathbf{A}\mathbf{R} & \mathbf{A}^2\mathbf{R} & \mathbf{A}^3\mathbf{R} & \mathbf{A}^4\mathbf{r}_1 & \mathbf{A}^4\mathbf{r}_2 \end{pmatrix} \quad (4.15)$$

The matrix $\mathbf{K}$ is likely to be ill-conditioned since the columns of $\mathbf{K}$ are formed based on the sequence $\mathbf{A}^i\mathbf{R}$, which quickly converges to the eigenvector of the largest eigenvalue [65]. To overcome the ill-conditioning problem, the Block Arnoldi algorithm computes another orthogonal matrix $\mathbf{Q}$ such that

$$\text{colsp}(\mathbf{Q}_s) = \text{colsp}(\mathbf{K}) \quad (4.16)$$

The matrix $\mathbf{Q}_s$ is well conditioned. A simple implementation of the Block Arnoldi algorithm is given below [41], [70].

```

function Block_Arnoldi
[ $q_0; \mathbf{R}$ ] = qr( $\mathbf{R}$ ) // qr factorization of  $\mathbf{R}$ 
    for j = 1 to k // k is the desired number of columns of the  $\mathbf{Q}$  matrix
         $\mathbf{z} = \mathbf{A}q_{j-1}$ ;
        for i = 1 to j-1
             $h_{ij} = q_i^T \mathbf{z}$ ;
             $\mathbf{z} = \mathbf{z} - q_i h_{ij}$ ;
        end for
        [ $q_j; \mathbf{R}$ ] = qr( $\mathbf{z}$ )
    end for
 $\mathbf{Q} = [q_0 \dots q_k]$ 
end function

```

A reduced system of order q preserves the first $[q/n_p]$ block moments of the original system [41].

4.3.1 Computation of the Multi-Dimensional Subspace

The first step of the algorithm is to compute the multi-dimensional subspace of the system. For that purpose, the block moments of $\mathbf{T}$ with respect to frequency (denoted by $\mathbf{Q}_s$) are computed using the procedure described above [41], [65] using Krylov subspace techniques and Block Arnoldi algorithm.

The next step is to compute the block moments of $\mathbf{T}$ with respect to the heat coefficients $h_1, \dots, h_i$ (denoted by the matrices $\mathbf{Q}_{h_i}$).

Consider the equation

$$\begin{aligned} \mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) + \sum_{i=1}^n h_i \mathbf{A}_{h_i} \mathbf{T}(t) = \\ \mathbf{M}_p \mathbf{T}_p + \mathbf{P}(t) + \sum_{i=1}^n h_i \mathbf{T}_{m_i} \mathbf{A}_{s_i} + \sum_{k=1}^m T_{f_k} \mathbf{G}_{f_k} + \sum_{j=1}^p P_j \mathbf{V}_{p_j} \end{aligned} \quad (4.17)$$

when a unit excitation is applied at the p^{th} port, where $p \in (1, 2, \dots, n_p)$ eq. 4.17 becomes [71]

$$\begin{aligned} \mathbf{C} \frac{d\mathbf{T}(t)}{dt} + \mathbf{G}\mathbf{T}(t) + \sum_{i=1}^n h_i \mathbf{A}_{h_i} \mathbf{T}(t) = \\ \mathbf{m}_p^p \mathbf{T}_p + \mathbf{P}(t) + \sum_{i=1}^n h_i \mathbf{T}_{m_i} \mathbf{A}_{s_i} + \sum_{k=1}^m T_{f_k} \mathbf{G}_{f_k} + \sum_{j=1}^p P_j \mathbf{V}_{p_j} \end{aligned} \quad (4.18)$$

where $\mathbf{m}_p^p$ is a binary vector with non-zero entry at p^th port (the p^{th} column of $\mathbf{M}_p$). For the p^th port excitation, the moments of T w.r.t. h_i (where the k^{th} moment of T w.r.t. h_i is represented by $\mathbf{m}_{h_i}^{(p,k)}$) are computed using the equations

$$\mathbf{D}\mathbf{m}_{h_i}^{(p,0)} = \mathbf{F}^p \quad (4.19)$$

$$\mathbf{D}\mathbf{m}_{h_i}^{(p,1)} = T_{m0}\mathbf{A}_{s_i} - \mathbf{A}_{h_i}\mathbf{m}_{h_i}^{(p,0)} \quad (4.20)$$

$$\mathbf{D}\mathbf{m}_{h_i}^{(p,2)} = -\mathbf{A}_{h_i}\mathbf{m}_{h_i}^{(p,1)} \quad (4.21)$$

$\vdots$

$$\mathbf{D}\mathbf{m}_{h_i}^{(p,q_{h_i})} = -\mathbf{A}_{h_i}\mathbf{m}_{h_i}^{(p,q_{h_i}-1)} \quad (4.22)$$

where $\mathbf{D} = \mathbf{G} + \sum_{i=1}^n h_{i0}\mathbf{A}_{h_i}$, $\mathbf{F} = \mathbf{m}_p^p + \mathbf{P} + \sum_{i=1}^n h_{i0}T_{m_{i0}}\mathbf{A}_{s_i} + \sum_{k=1}^m T_{f_{k0}}\mathbf{G}_{f_k} + \sum_{j=1}^p P_{j0}\mathbf{V}_{p_j}$, $h_{i0}, T_{m_{i0}}, T_{f_{k0}}$ and P_{j0} are the chosen expansion points of h_i, T_{m_i}, T_{f_k} and P_j respectively and q_{h_i} is the number of moments to be preserved with respect to h_i .

As for a multi-port system, block moments (a matrix rather than a vector) are introduced and the block moment $\mathbf{M}_{h_i}^j$ ($j \in 1, 2, \dots, q_{h_i}$) of T w.r.t. h_i are formed as follows

$$\mathbf{M}_{h_i}^0 = \begin{pmatrix} \mathbf{m}_{h_i}^{(1,0)} & \mathbf{m}_{h_i}^{(2,0)} & \dots & \mathbf{m}_{h_i}^{(p,0)} \end{pmatrix} \quad (4.23)$$

$$\mathbf{M}_{h_i}^1 = \begin{pmatrix} \mathbf{m}_{h_i}^{(1,1)} & \mathbf{m}_{h_i}^{(2,1)} & \dots & \mathbf{m}_{h_i}^{(p,1)} \end{pmatrix} \quad (4.24)$$

$$\vdots$$

$$\mathbf{M}_{h_i}^{q_{h_i}} = \begin{pmatrix} \mathbf{m}_{h_i}^{(1,q_{h_i})} & \mathbf{m}_{h_i}^{(2,q_{h_i})} & \dots & \mathbf{m}_{h_i}^{(p,q_{h_i})} \end{pmatrix} \quad (4.25)$$

Combining the above equations , the block moment $\mathbf{M}_{h_i}^j (j \in 1, 2, \dots, q_{h_i})$ of T w.r.t. h_i can be evaluated from

$$\mathbf{D}\mathbf{M}_{h_i}^0 = \mathbf{M}_p \quad (4.26)$$

$$\mathbf{D}\mathbf{M}_{h_i}^1 = T_{m0}[\mathbf{A}_{s_i} \dots \mathbf{A}_{s_i}] - \mathbf{A}_{h_i} \mathbf{M}_{h_i}^0 \quad (4.27)$$

$$\mathbf{D}\mathbf{M}_{h_i}^2 = -\mathbf{A}_{h_i} \mathbf{M}_{h_i}^1 \quad (4.28)$$

$$\vdots$$

$$\mathbf{D}\mathbf{M}_{h_i}^{q_{h_i}} = -\mathbf{A}_{h_i} \mathbf{M}_{h_i}^{q_{h_i}-1} \quad (4.29)$$

The subspace of the block moments of T w.r.t. h_i is given as

$$\text{colsp}(\mathbf{Q}_{h_i}) = \text{colsp} \begin{pmatrix} \mathbf{M}_{h_i}^1 & \mathbf{Q}_{Arnoldi} \end{pmatrix} \quad (4.30)$$

where

$$\mathbf{Q}_{Arnoldi} = \begin{pmatrix} \mathbf{M}_{h_i}^2 & \mathbf{M}_{h_i}^3 & \dots & \mathbf{M}_{h_i}^{q_{h_i}} \end{pmatrix} \quad (4.31)$$

Note that the 0^{th} block moment for all h_i 's is the same and is denoted by $\mathbf{M}_h^0$. It is already included as the first moment with respect to frequency in the matrix $\mathbf{Q}_s$. Moments of $\mathbf{T}$ with respect to T_{m_i}, T_{f_k} and P_j are computed using

$$\mathbf{Dm}_{T_{m_i}} = h_{i0} \mathbf{A}_{s_i} \quad (4.32)$$

$$\mathbf{Dm}_{T_{f_k}} = \mathbf{G}_{f_k} \quad (4.33)$$

$$\mathbf{Dm}_{P_j} = \mathbf{V}_{p_j} \quad (4.34)$$

Once all the required moments are evaluated, the multi-dimensional subspace (denoted by $\mathbf{Q}$) is computed such that

$$\text{colsp}(\mathbf{Q}) = \text{colsp}(\mathbf{K}) \quad (4.35)$$

where,

$$\mathbf{K} = \begin{pmatrix} \mathbf{Q}_s & \mathbf{Q}_{h_1} & \dots & \mathbf{Q}_{h_n} & \mathbf{m}_{T_{m_1}} & \dots & \mathbf{m}_{T_{m_n}} \\ & & & & \mathbf{m}_{T_{f_1}} & \dots & \mathbf{m}_{T_{f_m}} & \mathbf{m}_{P_1} & \dots & \mathbf{m}_{P_p} \end{pmatrix} \quad (4.36)$$

In the case when cross-derivatives are required to preserve the network response [46], the multi-dimensional subspace is modified to include the cross-derivatives (denoted by $\mathbf{M}_x$) as

$$\text{colsp}(\mathbf{Q}) = \text{colsp} \begin{pmatrix} \mathbf{K} & \mathbf{M}_x \end{pmatrix} \quad (4.37)$$

It is to be noted that once computed, the multi-dimensional subspace $\mathbf{Q}$ is a constant real matrix independent of frequency and and heat coefficients. We then define:

$$\mathbf{T}(t) = \mathbf{Q}\hat{\mathbf{T}}(t). \quad (4.38)$$

where $\hat{\mathbf{T}} \in \mathbb{R}^q$ and q is the reduced vector of unknowns ($q \ll N$).

Using (4.38) a congruent transformation is performed on (4.17) to give

$$\begin{aligned} \hat{\mathbf{C}} \frac{d\hat{\mathbf{T}}(t)}{dt} + \hat{\mathbf{G}}\hat{\mathbf{T}}(t) + \sum_{i=1}^n h_i \hat{\mathbf{A}}_{h_i} \hat{\mathbf{T}}(t) = \\ \hat{\mathbf{M}}_p \mathbf{T}_p + \hat{\mathbf{P}}(t) + \sum_{i=1}^n h_i T_{m_i} \hat{\mathbf{A}}_{s_i} + \sum_{k=1}^m T_{f_k} \hat{\mathbf{G}}_{f_k} + \sum_{j=1}^p P_j \hat{\mathbf{V}}_{p_j} \end{aligned} \quad (4.39)$$

where,

$$\begin{aligned} \hat{\mathbf{C}} = \mathbf{Q}^T \mathbf{C} \mathbf{Q}; \quad \hat{\mathbf{G}} = \mathbf{Q}^T \mathbf{G} \mathbf{Q}; \quad \hat{\mathbf{A}}_{h_i} = \mathbf{Q}^T \mathbf{A}_{h_i} \mathbf{Q}; \quad \hat{\mathbf{P}} = \mathbf{Q}^T \mathbf{P} \\ \hat{\mathbf{A}}_{s_i} = \mathbf{Q}^T \mathbf{A}_{s_i}; \quad \hat{\mathbf{G}}_{f_k} = \mathbf{Q}^T \mathbf{G}_{f_k}; \quad \hat{\mathbf{V}}_{p_j} = \mathbf{Q}^T \mathbf{V}_{p_j}; \quad \hat{\mathbf{M}}_p = \mathbf{Q}^T \mathbf{M}_p \end{aligned} \quad (4.40)$$

The size of the reduced network (4.39) is very small compared to the original network (4.17). The response at any node of the original network can be computed using (4.38) once the solution of the reduced system (4.39) is known.

It can be proved that the time-domain reduced network so formed preserves the

dominant eigen-values of the original network (4.17) [41]. In addition if the original network is passive, it can also be proved that the macro-model (4.39) is also passive [51].

This algorithm is used on the component-model equations to get

$$\hat{\mathbf{C}}_c \frac{d\hat{\mathbf{T}}_c(t)}{dt} + \hat{\mathbf{G}}_c \hat{\mathbf{T}}_c(t) = \hat{\mathbf{P}}_c(t) + \hat{\mathbf{M}}_p \mathbf{T}_p \quad (4.41)$$

$$\mathbf{I}_p = \hat{\mathbf{M}}_p^T \hat{\mathbf{T}}_c \quad (4.42)$$

4.4 Forming the Mixed Matrices

This section discusses the formulation of the mixed matrices, that include the equations of the detailed base-model as well as the equations of the reduced component-model [47]. After generating the macro-model matrices, the mixed matrices are formed from (4.1), (4.2), (4.41) and (4.42). For the case of a connecting a single macro-model we have,

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{C}_b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{\mathbf{C}}_c \end{pmatrix}; \quad \mathbf{G}_m = \begin{pmatrix} \mathbf{G}_b & 0 & \mathbf{L}^T & 0 \\ -\mathbf{L} & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & -\hat{\mathbf{M}}_p^T \\ 0 & -\hat{\mathbf{M}}_p & 0 & \hat{\mathbf{G}}_c \end{pmatrix}$$

$$\mathbf{T}_m = \begin{pmatrix} \mathbf{T}_b \\ \mathbf{T}_p \\ \mathbf{I}_p \\ \hat{\mathbf{T}}_c \end{pmatrix}; \quad \mathbf{P}_m = \begin{pmatrix} \mathbf{P}_b \\ 0 \\ 0 \\ \hat{\mathbf{P}}_c \end{pmatrix}$$

the mixed-model equations are then given by,

$$\mathbf{C}_m \frac{d\mathbf{T}_m(t)}{dt} + \mathbf{G}_m \mathbf{T}_m(t) = \mathbf{P}_m(t)$$

If the structure consists of more than one macro-model, MDMR is applied to each one and all are connected to the base-model using specific $\mathbf{L}$ matrices. The following example presents a system with a base-model and two macro-models,

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{C}_b & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \hat{\mathbf{C}}_{c1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \hat{\mathbf{C}}_{c2} & 0 \end{pmatrix}$$

Figure 4.3: Process flow for creating BCI reduced macro-models and forming a mixed-model.

$$\mathbf{G}_m = \begin{pmatrix} \mathbf{G}_b & 0 & 0 & \mathbf{L}_1^T & \mathbf{L}_2^T & 0 & 0 \\ -\mathbf{L}_1 & \mathbf{I} & 0 & 0 & 0 & 0 & 0 \\ -\mathbf{L}_2 & 0 & \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 & -\hat{\mathbf{M}}_{p_1}^T & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} & 0 & -\hat{\mathbf{M}}_{p_2}^T \\ 0 & -\hat{\mathbf{M}}_{p_1} & 0 & 0 & 0 & \hat{\mathbf{G}}_{c_1} & 0 \\ 0 & 0 & -\hat{\mathbf{M}}_{p_2} & 0 & 0 & 0 & \hat{\mathbf{G}}_{c_2} \end{pmatrix}$$

$$\mathbf{T}_m = \left(\mathbf{T}_b \quad \mathbf{T}_{p_1} \quad \mathbf{T}_{p_2} \quad \mathbf{I}_{p_1} \quad \mathbf{I}_{p_2} \quad \hat{\mathbf{T}}_{c_1} \quad \hat{\mathbf{T}}_{c_2} \right)^T ; \quad \mathbf{P}_m = \left(\mathbf{P}_b \quad 0 \quad 0 \quad 0 \quad 0 \quad \hat{\mathbf{P}}_{c_1} \quad \hat{\mathbf{P}}_{c_2} \right)^T$$

An important advantage of forming the mixed matrices is that all boundary conditions of the base-model as well as the macro-model can be explicitly incorporated.

As an example, a formulation of the case where a base-model with an external heat flow and a macro-model with an external heat flow and a power source is presented. The equations of the base-model with explicit boundary conditions are then,

$$\mathbf{C}_b \frac{d\mathbf{T}_b(t)}{dt} + \mathbf{G}_{b_0} \mathbf{T}_b(t) + h_b \mathbf{A}_{b_h} \mathbf{T}_b(t) + \mathbf{L}^T \mathbf{I}_p = h_b T_{m_b} \mathbf{A}_{b_s} \quad (4.44)$$

and for the macro-model,

$$\hat{\mathbf{C}}_c \frac{d\hat{\mathbf{T}}_c(t)}{dt} + \hat{\mathbf{G}}_{c_0} \hat{\mathbf{T}}_c(t) + h_c \hat{\mathbf{A}}_{c_h} \hat{\mathbf{T}}_c(t) = \hat{\mathbf{M}}_p \mathbf{T}_p + h_c T_{m_c} \hat{\mathbf{A}}_{c_s} + P_c \hat{\mathbf{V}}_c \quad (4.45)$$

Using equations (4.44) and (4.45) the mixed equations with explicit boundary conditions can be written as follows,

$$\begin{aligned} \mathbf{C}_m \frac{d\hat{\mathbf{T}}(t_m)}{dt} + \mathbf{G}_{m_0} \mathbf{T}_m(t) + h_b \mathbf{A}_{m_{h_1}} \mathbf{T}_m(t) + h_c \mathbf{A}_{m_{h_2}} \mathbf{T}_m(t) = \\ h_b T_{m_1} \mathbf{A}_{m_{s_1}} + h_c T_{m_2} \mathbf{A}_{m_{s_2}} + P_1 \mathbf{V}_{m_{p_1}} \end{aligned} \quad (4.46)$$

where

$$\mathbf{G}_{m_0} = \begin{pmatrix} \mathbf{G}_{b_0} & 0 & \mathbf{L}^T & 0 \\ -\mathbf{L} & \mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & \hat{\mathbf{M}}_p^T \\ 0 & -\hat{\mathbf{M}}_p & 0 & \hat{\mathbf{G}}_{c_0} \end{pmatrix}; \quad \mathbf{A}_{m_{h_1}} = \begin{pmatrix} \mathbf{A}_{b_h} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \quad \mathbf{A}_{m_{s_1}} = \begin{pmatrix} \mathbf{A}_{b_s} \\ 0 \\ 0 \\ 0 \end{pmatrix};$$

$$\mathbf{A}_{m_{h_2}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{\mathbf{A}}_{c_h} \end{pmatrix}; \quad \mathbf{A}_{m_{s_2}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \hat{\mathbf{A}}_{c_s} \end{pmatrix}; \quad \mathbf{V}_{m_{p_1}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \hat{\mathbf{V}}_c \end{pmatrix}$$

Prior to the final simulation of the mixed-model all BC's must be specified and the system written as,

$$\mathbf{C}_m \frac{d\mathbf{T}_m(t)}{dt} + \mathbf{G}_m \mathbf{T}(t) = \mathbf{P}_m(t) \quad (4.47)$$

This system of equations is approximately as large as the base-model and if this is of moderate size the transient could be found by numerical integration. However, if the base is large or if more direct simulation techniques are to be used it is useful to perform a second model reduction on (4.47) and obtain,

$$\hat{\mathbf{C}}_m \frac{d\hat{\mathbf{T}}_m(t)}{dt} + \hat{\mathbf{G}}_m \hat{\mathbf{T}}(t) = \hat{\mathbf{P}}_m(t) \quad (4.48)$$

where $\mathbf{T}_m = \mathbf{Q}_m \hat{\mathbf{T}}_m$. As this model reduction is only done with a small number of moments the size of the reduced model is very small typically less than 20. This size is sufficiently small to allow for the direct calculation of the transient using a pole/zero representation. Which is the subject of the next section.

Fig. 4.3 summarizes the process of creating a BCI reduced macro-models and forming a mixed-model.

4.5 Eigen Decomposition of the Mixed Matrices

Eigen Decomposition is performed on the equations

$$\hat{\mathbf{C}}_m \frac{d\hat{\mathbf{T}}_m(t)}{dt} + \hat{\mathbf{G}}_m \hat{\mathbf{T}}(t) = \hat{\mathbf{P}}_m(t) \quad (4.49)$$

The process is identical to the one described in section 3.4.

Each row in the vector $\hat{\mathbf{T}}_m(t)_i$ is of the form:

$$\hat{T}_m(t)_i = A_i - (a_{i1}e^{\frac{-t}{\bar{\lambda}_1}} + a_{i2}e^{\frac{-t}{\bar{\lambda}_2}} + \dots + a_{ij}e^{\frac{-t}{\bar{\lambda}_j}}) + (a_{i1}^0e^{\frac{-t}{\bar{\lambda}_1}} + a_{i2}^0e^{\frac{-t}{\bar{\lambda}_2}} + \dots + a_{ij}^0e^{\frac{-t}{\bar{\lambda}_j}})$$

where $a_{i1} \dots a_{ij}$ and $a_{i1}^0 \dots a_{ij}^0$ represent the residues associated with the input power and initial temperature distribution respectively and $A_i = \sum_{m=1}^j a_{im}$.

The time domain solution of the original system is calculated by

$$\mathbf{T}_m(t) = \mathbf{Q}_m \hat{\mathbf{T}}(t)$$

where

$$\mathbf{T}_m(t) = \begin{pmatrix} \mathbf{T}_b \\ \mathbf{T}_p \\ \mathbf{I}_p \\ \hat{\mathbf{T}}_c \end{pmatrix}$$

and $\mathbf{Q}_m$ is the orthonormal matrix generated by performing MDMR on the mixed-model equations.

$\mathbf{T}_b$ is the vector of the base-model temperatures, $\mathbf{T}_p$ is the vector of ports temperatures, $\mathbf{I}_p$ is the vector of ports heat flow and $\hat{\mathbf{T}}_c$ is the reduced vector of the macro-model temperatures. $\hat{\mathbf{T}}_c$ can be easily mapped back to the original component-model

temperature space by using,

$$T_c = Q_c \hat{T}_c$$

where Q_c is the orthonormal matrix generated by performing MDMR on the component-model equations.

For a simulation of a new set of BC's (on either the base-model or the component-models), the eigen decomposition of the reduced system is the only process that needs to be re-calculated. Since the reduced system is much smaller than the original one, this process is very quick. This method enables us to generate the time domain solution for any BC's at any desired time point very efficiently.

4.6 Computational Issues

As discussed in section 3.5, the computational cost for this method consists of three major parts: LU decomposition, QR process and Eigen Decomposition of the reduced system. Since hierarchal MDMR is performed, the LU decomposition and the QR process are done twice (once for each component-model and once for the mixed-model). The ED process is done on the reduced mixed model only. Table 4.1 presents the cost of each process.

Process	LU Component	QR Component	LU Mixed	QR Mixed	ED
Cost	n_c^α	$4q_c^2 n_c$	n_m^α	$4q_m^2 n_m$	q_m^3

Table 4.1: Computational cost

where n_c is the size of the original component-model, q_c is the size of the reduced component-model, n_m is the size of the mixed-model, q_m is the size of the reduced

mixed model and α is a constant of value 1.2-1.4.

The computational cost of the MDMR performed on the component-model depends on the sizes n_c and q_c . The size of q_c directly depends on the number of ports. As the number of ports is increased, q_c becomes larger, and might dominate the cost of this process. As for the mixed model, q_m is usually very small (size less than 20) and the computational cost of the MDMR of the mixed-model is of order n_m .

4.7 Results

In this section the accuracy and efficiency of the macro-modeling methodology outlined above will be established. The first model simulated will be very simple and an analytical solution is available. For more complex models three different methods were used to generate the transient solution: the *Atar* TLM solver, a direct transient solution using model reduction of the full detailed model (referred to as a “detailed-model”) and direct solution of the “mixed-model” incorporating macro-models using model reduction. For both of the later solutions the transient response was calculated using an Eigen Decomposition of the reduced system.

To determine the relative errors of the methods the error was defined as,

$$error = \frac{|T_1 - T_2|}{T_{ss}} \times 100 \quad (4.54)$$

where T_{ss} is the steady-state temperature and T_1 and T_2 represent the two methods compared. If an analytical solution is available the error will be calculated with respect to it. However, for the more complicated examples the error is determined with respect to the fully detailed solution. This allows to determine the error introduced

by the macro-modeling process due to either the use of ports and/or MDMR.

4.7.1 Example 1: Simple Square Model

In order to verify the accuracy of the proposed method, it was first tested on a rectangular model consisting of a substrate (base-model) and a single component-model, both of size $32 \times 32 \times 16$ (μm^3). A heat source of 0.02 W uniformly distributed over a region of $8 \times 8 \times 2$ (μm^3) was applied to one corner of the component-model. The back side of the base-model was fixed at 300K and mixed boundary conditions were applied on the top of the component-model. Adiabatic boundary conditions were applied to all the other surfaces of the model. The size of the base-model was 290 blocks and the component-model size was 340 blocks. Fig. 4.4a shows the structure and mesh of the model as well as the position of the heat source. The port configuration for this model was PPB and therefore the solution of the mixed-model will not introduce any error due to the port definition.

The time dependent analytical solution for this problem can be developed [68] using a Green's function approach. This leads to the following expression for the temperature:

$$\begin{aligned}
T(x, y, z, t) = & \frac{g\alpha}{\kappa} \sum_{m=1}^{\infty} 1 - e^{-\alpha\beta_m^2 t} \left(\frac{2 \sin(\beta_m x)}{w_x} \right) \left(\frac{\cos(\beta_m x_1) - \cos(\beta_m x_2)}{\beta_m} \right) \\
& \left[\frac{y_2 - y_1}{w_y} + \sum_{n=1}^{\infty} 1 - e^{-\alpha\gamma_n^2 t} \left(\frac{2 \cos(\gamma_n y)}{w_y} \right) \left(\frac{\sin(\gamma_n y_2) - \sin(\gamma_n y_1)}{\gamma_n} \right) \right] \\
& \sum_{p=1}^{\infty} 1 - e^{-\alpha\eta_p^2 t} \left(\frac{2(\eta_p^2 + H^2)}{l(\eta_p^2 + H^2) + H} \right) (\cos(\eta_p(l - z))) \\
& \left(\frac{\sin(\eta_p(l - z_1)) - \sin(\eta_p(l - z_2))}{\eta_p} \right)
\end{aligned}$$

where w_x , w_y and l are the model dimensions, ($32 \times 32 \times 16$), $\alpha = \kappa/C\rho$ where κ is the thermal conductivity, ρ is the material density and C is thermal capacity of the material. g is the power density over the region defined by the corner points (x_1, y_1, z_1) and (x_2, y_2, z_2) , $\beta_m = m\pi/w_x$, $\gamma_n = n\pi/w_y$, and η_p are the positive roots of the equation $\eta_p \tan(\eta_p l) = H$. H is defined by the heat transfer coefficient h divided by the thermal conductivity κ .

Fig. 4.5a shows the transient solution of the maximum temperature point on the component-model, generated using the four methods mentioned above. The SS temperature-rise is 16.37 K. Fig. 4.5b presents the error of the TLM, full numerical solution and the mixed-model with respect to the analytical solution. It is clearly seen that the error of both using either a detailed-model or a mixed-model is very small and below 1% – this error is attributed to the discretization of the model. The TLM solution does show some error due to the explicit integration used to obtain the transient response. This error could be reduced, but at the expense of longer simulation times.

An advantage of the new method is the ability to generate the temperature at all

b)

Figure 4.4: Square model
a) Mesh b) Temperature contour plot at SS

a)

b)

Figure 4.5: Results for a square model

a) Temperature rise as a function of time using different solution methods b) Error calculation w.r.t. the analytical solution

nodes at any specified time point. Fig. 4.4b presents a temperature contour plot of both the base-model and the component-model at steady-state. It was found that the proposed method is very accurate for all model nodes.

4.7.2 Example 2: Simple Rectangular Model

Next, the method was applied to a more complex model to evaluate the BCI of the macro-modeling technique. The model consists of a substrate of dimensions $128 \times 64 \times 16$ (μm^3) and a component-model of $64 \times 64 \times 16$ (μm^3). A power source of 0.04 W is positioned in the center of the component-model, and it is uniformly distributed over a region of $16 \times 16 \times 2$ (μm^3). Fig. 4.6a shows the structure and mesh of this model, as well as the position of the power source. For this model all the simulations were done with 8x8 port structure defined on the bottom of the component-model, which is a port-per-block (PPB) geometry for this model. This allows to isolate the error due to capturing the BC's using MDMR. Solution of the mixed-model was evaluated with respect to the solution for a detailed-model.

Two heat coefficients were applied to this model. One to the top of the component-model (referred to as h_1) and the other to the bottom of the substrate (h_2). The right side of the substrate was fixed at 300 K. Adiabatic boundary conditions were applied to all the other surfaces of the model. The model was simulated with various values of h_1 and h_2 , ranging from 0 to 10^{-3} .

The transient response was calculated for two specific points in the model. The first is the maximum temperature point of the component-model, and the second is the maximum temperature point of the substrate. Fig. 4.7a shows the transient response of the maximum temperature point on the component-model, for different values of

Figure 4.6: Mesh and contour plot for a rectangular model a) Mesh. The base-model consists of 385 blocks and the component-model of 5986 blocks b) Temperature contour plot at SS for the full model

a)

b)

Figure 4.7: Maximum temperature-rise as a function of time
a) component-model b) substrate

a)

b)

Figure 4.8: Component-Model Error

a) Relative error b) Absolute error

a)

b)

Figure 4.9: Base-Model Error

a) Relative error b) Absolute error

the heat coefficients. The maximum steady-state value is achieved when both heat coefficients are zero. This value is 79.61 K. As the heat coefficients values rise, the SS temperature-rise decreases, until it reaches its minimum value. For this model the value is 6.95 K, and it is achieved when $h_1 = h_2 = 10^{-3}$. For heat coefficients of a value higher than that, the SS temperature-rise does not change anymore.

Fig. 4.7b shows the transient response of the maximum temperature point on the substrate, for the same values of the heat coefficients as in Fig. 4.7a. As expected, the steady-state temperature-rise of this point is much lower than the temperature-rise of the point on the component-model, since the heat source is in the component-model itself. As before, the maximum SS value (31.1 K) is achieved when both heat coefficients are zero and the minimum value (0.5 K) is achieved when $h_1 = h_2 = 10^{-3}$.

Fig. 4.8a presents the error in the transients of the component-model as function of time for different values of h_1 and h_2 . From the error graph, it is clear that as the values of the heat coefficients rise, the error increases. For the case where $h_1 = h_2 = 0$, the peak error reaches 0.68 % at $t = 9 \times 10^{-8} s$ and stabilizes on $1.025 \times 10^{-3} \%$ at SS. For the $h_1 = 10^{-6}$ and $h_2 = 10^{-8}$ the peak error is 0.87% at $5 \times 10^{-8} s$ and stabilizes on $1.2 \times 10^{-3} \%$ at SS. The peak error when $h_1 = h_2 = 10^{-3}$ is 3.28% at $10^{-7} s$ and $2.3 \times 10^{-2} \%$ at SS.

Fig. 4.9a presents the error of the transient response for the substrate point. The error in the transient solution for the maximal heat flow case is somewhat large, this is in part due the very small SS temperature-rise which is used to normalize the error. Figures 4.8b and 4.9b show that the absolute error is quite small and of almost the same magnitude for all cases. It is possible to reduce the relative and absolute error by incorporating more moments into the reduced model.

Fig. 4.6b shows the temperature contour plot of the full model at SS derived from the mixed-model.

4.7.3 Effect of the Port Configuration

In this section the effect of different port configurations on the model used in the previous section will be presented. In the previous section, the model was simulated with a port-per-block configuration and each block at the bottom of the sub-model formed a port. The error measured was due to the model-reduction process and not presence of ports.

For this section, the same model was simulated with different values of heat coefficients and 5 different port configurations. The first configuration was 1×1 (a single port). In this configuration all the nodes at the bottom of the component-model formed one port (they were all kept at the same temperature). The next port configuration was 2×2 , meaning the x-axis was divided into two ports, and so was the y-axis. Together, they formed 4 ports at the bottom of the component-model. The 4×4 port configuration formed four ports on each axis, and 16 all together. The 4×8 configuration divided the x-axis into 4 ports and the y-axis into 8, forming 32 ports at the bottom of the component-model and the last configuration, the 8×8 formed 64 ports, which is exactly the port-per-block case.

The transient responses for the same two points (maximum temperature points for the component-model and substrate) as previously were generated. Fig. 4.10a shows the response of the component-model point for different values of heat coefficients and port configurations. The upper part of the graph presents the response for the 5 port configuration for $h_1 = h_2 = 0$. Each port configuration predicts a different

a)

b)

Figure 4.10: Different Port Configurations

a) Temperature rise as a function of time for a node on the component-model b) Temperature rise as a function of time for a node on the substrate

a)

b)

Figure 4.11: Error for the component-model node
a) $h_1 = h_2 = 0$ b) $h_1 = h_2 = 10^{-3}$

steady-state value. The 1×1 configuration predicts the lowest temperature-rise, 68.23 K and the 8×8 (port-per-block) produces 79.61 K.

The error of a particular port configuration is determined with respect to the detailed-model. For the case of $h_1 = h_2 = 0$ the error is presented in Fig. 4.11a. The error value range is from 14.3 % for the 1×1 port configuration to 0.76 % for the 4×8 port configuration. This result is not surprising, since the number of ports is actually the number of fixed temperature regions at the bottom of the component-model. For the 1×1 configuration, the bottom of the component-model is kept at the same temperature, so we expect to get a large error. As we gradually increase the number of ports, we expect the error to decrease, because the bottom of the component-model is divided to more and more sections, allowing for a smooth transition in the heat flow through it.

The middle set of data of Fig. 4.10a presents the transient response for the case of $h_1 = 10^{-6}$ and $h_2 = 10^{-8}$. In this case, the steady-state temperature-rise varies from 40.62 K for the 1×1 port configuration to 42.36 K for the port-per-block configuration (8×8). The maximum error is 4.2 %, which is much smaller than the error calculated for the $h_1 = h_2 = 0$ case.

The lower set of data of Fig. 4.10a presents the transient response for the case of $h_1 = h_2 = 10^{-3}$. The SS temperature-rise range is 6.91 K for the 1×1 port configuration up to 6.953 K for the PPB configuration. The error for this case is presented in Fig. 4.11b. The maximum SS error value is 0.62 % for the 1×1 port configuration and the minimum is 0.0083 % for the 4×8 port configuration.

Fig. 4.10b presents the transient responses for different heat coefficients and different port configurations for the maximum temperature point of the substrate. Again,

three combinations of heat coefficients are presented, and for each one the responses of 5 port configurations are shown. The SS temperature-rise when $h_1 = h_2 = 0$ ranges from 15.9 K to 30.18 K for the 1×1 and 8×8 port configurations, respectively. For the $h_1 = 10^{-6}$ and $h_2 = 10^{-8}$ case, the SS temperature-rise range is from 4.29 K to 8.23 K and when $h_1 = h_2 = 10^{-3}$ the range is 0.27 K to 0.34 K.

It should be noted that the error determined in all these cases was not due to the macro-modeling methodology. It is due to the presence of ports and would be found when ever any type of macro or compact model was used.

4.7.4 Large PCB and Package Example

In order to demonstrate the advantages the proposed method offers, the macro-model method was applied to a large scale model of a printed circuit board (PCB) and a number of electronic packages. The PCB substrate is of dimensions $11.5 \times 11.5 \times 0.5$ (cm^3) and 16 identical component-models, of $1.2 \times 1.2 \times .3$ (cm^3) and was modeled with a single effective thermal conductivity. Each package model consisted of a package, a die and 64 interconnection bumps. A power source of 0.5 W was positioned in the center of the die and it is uniformly distributed over a region of $320 \times 320 \times 1$ (μm^3). Fig. 4.12a shows the base-model and the 16 component-models placed on it. Fig. 4.12b shows a cross section of a single package component-model. The right side of the base-model was fixed at 300 K and each package model was parameterized with a heat flow off the top surface. Adiabatic boundary conditions were applied to all the other surfaces of the model.

a)

b)

Figure 4.12: *Atar* models:

a) PCB substrate with 16 packages. The base-model consists of 13,876 blocks and each component-model is 20,504 blocks b) a cross section of a single package

Simulation Efficiency

To evaluate the speed-up of the macro-model methodology the time needed to generate a transient solution for this model was determined for three simulation methods. As a baseline the *Atar* TLM algorithm was used. Secondly, MDMR was performed on the detailed model consisting of the base-model and component-models, and the transient response calculated directly from the system's eigenvalues. Finally, mixed-model equations were created, consisting of reduced component-models and the base-model. After forming the mixed-model an MDMR was performed and the eigenvalues of the reduced mixed-model system used to determine the response. Simulation times for two macro-model configurations were determined. The first configuration had 4 ports (2×2) and the second 32 ports (4×8). The size of the macro-model matrices depends directly on the number of ports and for the 4 port case the reduced size is 63 while for the 32 port case it is 315.

Table 4.2 presents the model sizes for the detailed-models and mixed-models when one, five or 16 packages are connected. For the case of the detailed-model the model size grows very rapidly as each component-model consists of 20,504 blocks. The size of the mixed-model changes a little, however, as more macro-models are added. It ranges from 13,947 to 15,012 for the 4 port configuration and from 14,255 to 19,940 for the 32 port one. As macro-model matrices (of size 63 or 315) are added to the mixed matrices, the change from 1 macro-model to 16 is small. The size of the matrices remains in the same order of magnitude and the simulation time does not change significantly. The size of the final reduced model created by performing the second model reduction was, for both the detailed-model and mixed-model, 20.

Fig. 4.13 shows the time required to generate a transient solution as a function of

the number of component or macro-models connected to the base-model for the three simulation methods. It can clearly be seen from this figure that the use of model reduction is much less CPU intensive than using a method such as TLM. From this figure it can also be seen the clear advantage of using macro-modeling techniques. When simulating the detailed-model, the LU decomposition needed to perform the MR is a major concern. It fails (for a PC desktop system with 4G of RAM) when the size of the model is too large. The highest number of component-models which could be connected to detailed-model and simulation completed was found to be 5 (producing a model of size 116,396 blocks). When macro-models are used their number has a small effect on simulation time and no memory limitations were present. This emphasizes the fact that the proposed method enables large models with a high number of macro-models to be simulated. It should be emphasized that the use of the macro-model does not limit the prediction capability of the simulation, all nodal temperatures (both base and components) can be predicted.

Number of Packages	Detailed-model size	Mixed-model size	
		4 ports	32 ports
1	34,380	13,947	14,255
5	116,396	14,231	15,771
16	341,940	15,012	19,940

Table 4.2: Model Sizes for PCB base model and a variable number of packages

Package Heat Flow Variation

The use of parameterized macro-models is illustrated in this section. The heat flow coefficient h applied to the top of each one of the 16 component-models can be set independently. For the following simulations each row of 4 component-models was

Figure 4.13: Simulation time for PCB model as a function of the number of connected package models

assigned a different value of h . h_1 is the heat coefficient of the row furthest away from the fixed right side, and h_4 is the value of the closest row of component-models as seen in fig. 4.14.

Figure 4.14: Heat coefficient assigned

Initially, different heat coefficients were applied to the component-models, while using a port configuration of one port per component-model. As expected, as the value of h increases, the SS temperature-rise decreases as shown in fig. 4.15a. This figure presents the transient response of the maximum temperature block of each component-model. A heat coefficient of $h_1 = 10^{-3}$ was applied to the first row of component-models (denoted packages 1-4). Those 4 component-models have the same response, since they have the same power source, same h value and same distance from the fixed boundary condition on the right side of the base-model. Their SS

temperature-rise is the lowest possible 11.67 K. A heat coefficient of $h_2 = 10^{-6}$ was applied to the second row of component-models (denoted packages 5-8), resulting in a SS temperature-rise of 12.61 K. For the third row (packages 9-12) $h_3 = 10^{-7}$ and the SS temperature-rise is 15.23 K. The highest SS temperature-rise was achieved for the last row of component-models (packages 13-16) with $h_4 = 0$ and is 22.87 K.

Next, the port configuration was changed, while the values of $(h_1, \dots, h_4)$ were kept as mentioned above. The port configurations used are 2×2 (2 ports along the x-axis and 2 along the y-axis), 4×4 (4 ports per axis), 4×8 (4 ports along the x-axis and 8 along the y-axis) and 16×2 . The transient responses of the different port configurations are presented in fig. 4.15b. If we examine the upper part of the figure, (the packages 13-16 graph where $h_4 = 0$) we can see that for different port configurations the SS temperature-rise stabilizes on different values. The SS temperature-rise for the 1×1 port is 22.87 K while for the 16×2 it is 24.23 K. The error due to the port configuration was measured by comparing each of the responses to the 16×2 response. This port configuration is the closest to the port-per-block one, which is 16×16 for this model. The results are shown in fig. 4.16a. The highest error is caused by the 1×1 port configuration (5.6 %) and the lowest is due to the 4×8 configuration (2.55 %).

Another important point is that as the value of h increase, the error due to the different port configuration significantly decreases. This is clearly seen when we examine the packages 1-4 graph in fig. 4.15b. The error for this case (where $h_1 = 10^{-3}$) is presented in fig. 4.16b. All the responses are within a 1 % error range. The highest error is 0.92 % and the lowest is 0.27 % for the 1×1 and 4×8 port configurations, respectively.

a)

b)

Figure 4.15: Results for a large scale model
a) 4 different h values for a one port per component-model b) Different port configurations for different h values

a)

b)

Figure 4.16: Error due to Port Configurations

a) Error between the port configurations for $h = 0$ (packages 13-16) b) Error between the port configurations for $h = 10^{-3}$ (packages 1-4)

The geometry of the port configuration is an important parameter. Although the total number of the ports in the 4×8 and the 16×2 port configuration is the same (32 ports), fig. 4.15b shows that the transient response is not identical. For the packages 13-16 graph the 16×2 SS temperature-rise is a little higher (24.23 K versus 23.61 K). This shows that not only the number of ports effects the response, but also the geometry of the ports along the axis.

4.7.5 Pulse Power Simulation

Another advantage of the proposed method is the ability to simulate a model with multiple component-models, each excited with a different power source. Figs. 4.17a shows the transient responses of the maximum temperature block when each row of component-models has different excitation power. In this case, the first row of component-models (packages 1-4) were excited using a power vector $\mathbf{P}_0$ the second row (packages 5-8) with twice the power ($2\mathbf{P}_0$), the third (packages 9-12) with $3\mathbf{P}_0$ and the fourth (packages 13-16) with $4\mathbf{P}_0$.

The substitution of the different power vectors is done during the formation of the mixed matrices. Only the power vector $\mathbf{P}_{m_0}$ has to be altered. The model does not have to be re-built and the MDMR of the component-model does not have to be repeated. This allows for a quick simulation for various initial power excitations.

Using the proposed method, it is possible to simulate the model when each component-model is excited independently. Again, the only change is done while forming the mixed matrices. Since the MDMR process is linear super-position was used to generate the transient responses at various points on the model for different excitation times. For these simulations all the heat coefficients were set to 0, and the transient

a)

b)

Figure 4.17: a) Temperature rise for each set of packages with varying power in each set b) Temperature rise for Package 6 for three different excitations. The lower part of the plot illustrates when particular packages are "on" for each power configuration

Figure 4.18: Results for pulse input
a) Temperature distribution at t_1 b) Temperature distribution at t_2

response was generated 4 times. Each time only one row of 4 component-models was excited, while the rest were kept off. The result was 4 separate transient responses. In order to generate the response of a certain point for a pulse in each group of component-models the initial response was shifted and inverted according to the desired pulse rise and fall times.

Fig. 4.17b shows the transient response of a point in package 6 for three different pulsed power configurations. Clearly, the temperature on a certain component-model depends not only on the excitation of the package it models, but on the surrounding component-models as well. The response is a pulse because the surrounding component-models are turned on and off at different times.

Figs. 4.18a and b show a temperature contour plot of all nodes of the model at two different time points during the pulse. At the first time point, t_1 , the first two rows of component-models are fully excited, while the last two are off. At the second time point, t_2 , the first row of component-models is still on, but the second row is off. In addition, the third row of component-models has been excited. These contour plots demonstrate the ability of this method to easily generate the nodal temperatures for both the base-model and component-models at any desired time point for any excitation of the component-models.

4.8 Summary

A new methodology to significantly improve the accuracy, flexibility and efficiency of transient thermal macro-models of devices and modules was successfully presented. This methodology uses the mixed method and hierarchical MDMR to generate the

transient solution of different electronic systems.

Different BC's were applied, both to the component-models and the base-model, demonstrating the BCI nature of this method.

The effect of different port configurations was presented, and the accuracy due to those configurations was measured.

The accuracy of the proposed methodology was found to be very good. It was compared to the analytical solution of a square model, and to the detailed solution of more complex models.

Among the advantages of this method, is the ability to easily generate the nodal temperatures for both the base-model and component-models at any desired time point for various boundary conditions as well as any power excitation of the component-models.

The proposed methodology offers not only great speed-up (when compared to the detailed solution) but it also enables transient simulations of complex systems, consisting of many component-models. These systems cannot be solved using existing methods due to their size and complexity.

Chapter 5

Conclusions and Future Work

This chapter contains a summary of the work presented in this thesis and directions of future work.

5.1 Summary

1. The MDMR-ED technique was successfully applied to transient simulation of different electronic packages. The method consists of performing MDMR followed by an Eigen Decomposition of the reduced system. The solution is then obtained by using the poles and residues of the reduced model. The reduced solution is easily mapped back to the original model by a transformation matrix. By exploiting the multi-dimensional nature of the model reduction, BC's parameters as well as initial conditions, were incorporated explicitly into the reduced model, resulting in a boundary and initial conditions independent model (BICI). This allowed to perform multiple simulations with different sets of BC's applied to the reduced model. Moreover, the reduced model is capable of gen-

erating a full 3D temperature distribution in the original model. The method has been shown to be accurate and much faster than the TLM method it was compared to. The speed up was found to be a function of the model's size and complexity. For the models simulated in this work, speed-up of up to 2-3 orders of magnitude was achieved. The main limitation of this method is presented by the LU decomposition process that is performed during the MDMR. For models containing a large number of blocks, this process fails.

2. As a solution to the LU decomposition limitation, the MDMR-ED technique was extended to the case where multiple thermal ports were present. The systems simulated consisted of a detailed base-model and one or more component-models attached to it. The mixed method was introduced as an efficient way to link component-models to a base-model. The mixed method allows BC's of both the base-model and the component-models to be explicitly incorporated, so it can handle BC's variations of both the base-model and the component-models. Hierarchical model reduction of the mixed model is feasible, resulting in a very small mixed model that can easily and quickly be solved. The solution is then mapped back to the original mixed-model by using transformation matrix $\mathbf{Q}_m$. Then, by using the transformation matrix $\mathbf{Q}_c$, all the internal temperatures of the original component-model can be achieved.

5.2 Future Work

The work presented in this thesis may be further extended and could be applied in various fields. Given below is a brief list of the future work that can be done:

1. Applications of the MDMR-ED technique to Microelectromechanical systems (MEMS).
2. The MDMR technique can be used for optimization of large systems with respect to design parameters.
3. Use of the MDMR-ED technique for non-linear thermal systems.
4. Use of MDMR combined with parameterized eigen-decomposition process.
5. Thermal applications of MDMR-ED using multi-point model reduction.
6. Applications of MDMR-ED to complicated systems, containing several levels of hierarcal model reduction.

Appendix A

Simulation Summary

The tables below present the specifications for each simulation performed on the various packages discussed in this work.

A.1 MDMR-ED Simulations

Table A.1 presents the simulations performed using the MDMR-ED method. The size of the original model, the size of the reduced model, as well as the simulation time are presented. The boundary conditions applied to each model are also specified. Mixed boundary conditions are specified using a heat coefficient h , whereas T_f represents a fixed temperature surface. Adiabatic boundary conditions were applied to all other surfaces of the model.

Model	Model size	Reduced size	BC Applied	Simulation time (m)
Rectangular	200	12	Right - $T_f = 300K$ Bottom - $h = 0 - 10^{-3}$	0.29
Rectangular	1,444	12	Right - $T_f = 300K$ Bottom - $h = 0 - 10^{-3}$	0.352
Rectangular	12,856	12	Right - $T_f = 300K$ Bottom - $h = 0 - 10^{-3}$	2.19
Rectangular	32,480	12	Right - $T_f = 300K$ Bottom - $h = 0 - 10^{-3}$	6.71
Rectangular	62,096	12	Right - $T_f = 300K$ Bottom - $h = 0 - 10^{-3}$	72.42
Rectangular	200	16	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	0.32
Rectangular	1,444	16	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	0.4
Rectangular	12,856	16	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	2.53
Rectangular	32,480	16	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.9
Rectangular	62,096	16	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	78.8
Rectangular	62,096	9	Bottom - $h = 0 - 10^{-3}$ Bottom - $T_m = 320K$ Right - $T_f = 300K$ Rear - $T_f = 320K$	70.1
Rectangular	62,096	19	Bottom - $h = 0 - 10^{-3}$ Bottom - $T_m = 320K$ Right - $T_f = 300K$ Rear - $T_f = 320K$	79.3
Rectangular	62,096	27	Bottom - $h = 0 - 10^{-3}$ Bottom - $T_m = 320K$ Right - $T_f = 300K$ Rear - $T_f = 320K$	81

Model	Model size	Reduced size	BC Applied	Simulation time (m)
Rectangular	62,096	46	Bottom - $h = 0 - 10^{-3}$ Bottom - $T_m = 320K$ Right - $T_f = 300K$ Rear - $T_f = 320K$	89.6
CBGA	53,976	14	Bottom - $T_f = 300K$ Top - $h = 0 - 10^{-5}$	27.4
CBGA	53,976	20	Top - $h_1 = 0 - 10^{-5}$ Rear - $h_2 = 0 - 10^{-5}$ Right - $h_3 = 0 - 10^{-5}$	29.6

Table A.1: MDMR-ED Simulations

A.2 Macro-model Method Simulations

Table A.2 presents the simulations performed using the macro-model method. The parameters presented are the number of component-models (N_c), the base-model size (n_b), each component-model size (n_c), the number of ports for each component-model (n_p), the mixed-model size (n_m), the boundary conditions applied and the simulation time.

Model	N_c	n_b	n_c	n_p	n_m	BC Applied	Sim time (m)
Square	1	290	340	4	338	Top - $h = 0 - 10^{-3}$ Bottom - $T_f = 300K$	0.6
Rectangular	1	385	5986	1	397	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	0.8
Rectangular	1	385	5986	4	433	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	0.85
Rectangular	1	385	5986	16	577	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	0.9

Model	N_c	n_b	n_c	n_p	n_m	BC Applied	Sim time (m)
Rectangular	1	385	5986	32	769	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	0.95
Rectangular	1	385	5986	64	1152	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.03
Large PCB	1	13,876	20,504	4	13,947	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.63
Large PCB	2	13,876	20,504	4	14,018	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.64
Large PCB	3	13,876	20,504	4	14,089	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.65
Large PCB	4	13,876	20,504	4	14,160	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.66
Large PCB	5	13,876	20,504	4	14,231	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.67
Large PCB	6	13,876	20,504	4	14,302	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.68
Large PCB	7	13,876	20,504	4	14,373	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.69
Large PCB	8	13,876	20,504	4	14,444	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.7
Large PCB	9	13,876	20,504	4	14,515	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.72
Large PCB	10	13,876	20,504	4	14,586	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.73

Model	N_c	n_b	n_c	n_p	n_m	BC Applied	Sim time (m)
Large PCB	11	13,876	20,504	4	14,657	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.74
Large PCB	12	13,876	20,504	4	14,728	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.75
Large PCB	13	13,876	20,504	4	14,799	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.76
Large PCB	14	13,876	20,504	4	14,870	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.77
Large PCB	15	13,876	20,504	4	14,941	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.78
Large PCB	16	13,876	20,504	4	15,012	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.8
Large PCB	1	13,876	20,504	32	14,255	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.13
Large PCB	2	13,876	20,504	32	14,634	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.22
Large PCB	3	13,876	20,504	32	15,013	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.31
Large PCB	4	13,876	20,504	32	15,392	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.46
Large PCB	5	13,876	20,504	32	15,771	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.6
Large PCB	6	13,876	20,504	32	16,150	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.73

Model	N_c	n_b	n_c	n_p	n_m	BC Applied	Sim time (m)
Large PCB	7	13,876	20,504	32	16,529	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	6.86
Large PCB	8	13,876	20,504	32	16,908	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.03
Large PCB	9	13,876	20,504	32	17,287	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.21
Large PCB	10	13,876	20,504	32	17,666	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.4
Large PCB	11	13,876	20,504	32	18,045	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.55
Large PCB	12	13,876	20,504	32	18,424	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.7
Large PCB	13	13,876	20,504	32	18,803	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	7.9
Large PCB	14	13,876	20,504	32	19,182	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	8.08
Large PCB	15	13,876	20,504	32	19,561	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	8.31
Large PCB	16	13,876	20,504	32	19,940	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	8.55
Large PCB	16	13,876	20,504	1	13,914	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	1.2
Large PCB	16	13,876	20,504	16	14,079	Top - $h_1 = 0 - 10^{-3}$ Bottom - $h_2 = 0 - 10^{-3}$ Right - $T_f = 300K$	4.7

Table A.2: Macro-model Method Simulations

Bibliography

- [1] Mohamed-Nabil Sabry, "Dynamic compact thermal models used for electronic design: a review of recent progress," *Proc. of International Electronic Packaging Technical Conference and Exhibition*, July 6-11, 2003, Maui, Hawaii, U.S.A.
- [2] Niccolo Rinaldi, "On the modeling of the transient thermal behaviour of semiconductor devices," *IEEE Trans. Electron Devices*, vol. 48, no. 12, December 2001.
- [3] Yu Zhu, John K. Twynam, Motoji Yagura, Masatomo Hasegawa, Takao Hasegawa, Yoshihito Eguchi, Atsushi Yamada, Eiji Suematsu, Keiichi Sakuno, Hiroya Sato and Nobuo Hashizume, "Analytical model for electrical and thermal transients of self-heating semiconductor devices," *IEEE Trans. Microwave Theory and Techniques*, vol. 46, no. 12, December 1998.
- [4] J.E. Sergeant and A. Krum, "Thermal management handbook for electronic assemblies," *McGraw-Hill*, New-York 1998.
- [5] W. Luiten, "Sense and nonsense thermal requirements," 19th *IEEE SEMI-THERM Symposium*, pp. 332-335, March 2003.
- [6] Y. Joshi, M. Baelmans, D. Copeland, C.J.M. Lasance, J. Parry and J. Rantal, "Challenges in thermal modeling of electronics at the system level: summary of

- panel held at the Thermnic 2000 ,” *IEEE Trans. Components and Packaging Technologies*, vol. 24, no. 4, pp. 611–613, 2001.
- [7] Vladimir Szekely , Marta Rencz, Sandor Torok and Sandor Ress, ”Calculating effective board thermal parameters from transient measurments,” *IEEE Trans. Components and Packaging Technologies*, vol. 24, no. 4, December 2001.
 - [8] Filip Christiaens and Eric Beyne, ”Transient thermal modeling and characterization of a hybrid component,” *IEEE Electronic Components and Technology Conference*, 1996.
 - [9] B. Chambers, T.Y. Tom Lee and W. Blood, ”Steady state and transient thermal analysis of chip scale packages,” *6th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems ITherm 98*, pp. 68-75, 1998.
 - [10] M. Trivedi and K. Shenai, ”Failure mechanisms of IGBT’s under short-circuit and clamped inductive switching stress,” *IEEE Trans. Power Electronics*, vol. 14, no. 1, 1999.
 - [11] E. McShane and K. Shenai, ”Package effects on avalanche rating of power MOS-FET’s,” *IEEE International Workshop on Integrated Power Packaging, IWIPP*, pp. 93-96 2000.
 - [12] N.Y.D. Shammass, M.P. Rodriguez, A.T. Plumpton and D. Newcombe, ”Finite element modeling of thermal fatigue effects in IGBT modules,” *IEEE Proc. Circuits, Devices and Systems*, vol. 148, no. 2, 2001.
 - [13] Y.S. JU and K.E. Goodson, ”Thermal mapping of interconnects subjected to brief electrical stresses,” *IEEE Electron Device Letters*, vol. 18, no. 11, 1997.

- [14] G. Storti-Gajani, A. Brambilla and A. Premoli, "Electrothermal dynamics of circuits: analysis and simulations," *IEEE Trans. Circuits and Systems*, vol. 48, no. 8, 2001.
- [15] M. Rencz, V. Szekely, A. Morelli and C. Villa, "Determining partial thermal resistances with transient measurements, and using the method to detect die attach discontinuities," *18th annual IEEE Semiconductor Thermal Measurement and Management Symposium, SEMI-THERM XVIII*, pp. 15-20, 2002.
- [16] D.C. Katsis and J.D. van Wyk, "Void induced thermal impedance in power semiconductor modules: some transient temperature effects," *Industry Application Conference, IAS'01*, vol. 3, pp. 1905-1911, 2001.
- [17] C.J.M. Lasance and C. Lacaze, "A transient method for the accurate measurement of interface thermal resistance," *12th annual IEEE Semiconductor Thermal Measurement and Management Symposium, SEMI-THERM XII*, pp. 43-50, 1996.
- [18] D.S. Popescu, C. Dunare, P. Lerch, P. Renaud, M. Modreanu and D. Dascalu, "Integrated flow sensor for fluids with dynamic velocity, temperature and conductivity profiles," *International Conference on Semiconductor, CAS*, vol. 2, pp. 561-564, 1998.
- [19] O. Vancauwenberghe, J. Short, E. Giehler, P. Bildstein, P. Ancey and M. Gschwind, "Design of the interface electronics for an integrated microsensor for the preventive detection of water condensation," *8th International Conference on Solid State Sensors and Actuators*, vol. 1, pp. 835-838, 1995.
- [20] A.K. Henning, "Microfluidic MEMS," *IEEE Aerospace Conference*, vol. 1, pp. 471-486, 1998.

- [21] N.R. Swart and H. Nathan, "Coupled electrothermal modeling of microheaters using SPICE," *IEEE Trans. Electron Devices*, vol. 41, no. 6, 1994.
- [22] C.C. Lee, J.W. Lee, D.H. Chien and R. Shih, "Transient thermal study of laser diodes," *13th annual IEEE Semiconductor Thermal Measurement and Management Symposium, SEMI-THERM XIII*, pp. 218-223, 1997.
- [23] J.D. Chung, Y.M. Cho, J.S. Lee, C.H. Jung, Y.J. Choi and K. Jung, "Numerical analysis of 3-D, unsteady heat and flow in rapid thermal processing," *IEEE Proc. International Symposium on Industrial Electronics, ISIE 2001*, vol. 2, pp. 1208-1212, 2002.
- [24] V.A. Chiriack and T.Y.T. Lee, "Transient thermal analysis of an ACF package assembly process," *IEEE Trans. Components and Packaging Technologies*, vol. 24, no. 4, 2001.
- [25] Cao Lipeng, J.P. Krusius, M.A. Korhonen and T.S. Fisher, "Transient thermal management of portable electronics using heat storage and dynamic power dissipation control," *IEEE Trans. Components, Packaging and manufacturing Technologies*, vol. 21, no. 1, 1998.
- [26] D. Sylvester and H. Kaul, "Power-driven challenges in nanometer design," *IEEE Design and Test of Computers*, vol. 18, no. 6, 2001.
- [27] H.I. Rosten, C.J.M. Lasance and J.D. Parry, "The world of thermal characterization according to DELPHI part I: Background to DELPHI," *IEEE Trans. Components, Packaging and Technology*, vol. 20, no. 4, December 1997.
- [28] H. Vinke and C.J.M. Lasance, "Recent achievements in the thermal characterization of electronic devices by means of boundary condition independent compact models," *13th IEEE SEMI-THERM Symposium*, pp. 32-39, January 1997.

- [29] T. Smy, D. Walkey, and S.K. Dew, "Transient 3D heat analysis for integrated circuit devices using the transmission line matrix method on a quad tree mesh," *Solid-State Electronics*, vol. 45, pp. 1137–1148, Jan. 2001.
- [30] A. Bar-Cohen, T. Elperin and R. Eliasi, " θ_{jc} characterization of chip packages - justification, limitations and future," *IEEE Trans. CHMT*, 1989.
- [31] C.J.M. Lasance, "Thermal characterization of electronic parts with compact models: Interpretation, application and the need for a paradigm shift," 13th *IEEE SEMI-THERM Symposium*, pp. 40-48, March 1997.
- [32] H.I. Rosten, "DELPHI - A status report on the European-funded project for the Development of Libraries and Physical Models for an Integrated Design Environment," *IEEE Electronic Components and Technology Conference*, 1996.
- [33] J. DeVoe and A. Ortega, "An investigation of board level effects on compact thermal models of electronic chip packages," 18th *IEEE SEMI-THERM Symposium*, pp. 8-14, March 2002 .
- [34] J. DeVoe and A. Ortega, "An assessment of the behavior of compact thermal models of electronic packages in a printed circuit board level environment," 18th *IEEE SEMI-THERM Symposium*, pp. 69-76, May 2002.
- [35] J. DeVoe and A. Ortega, "On the performance of compact thermal models of electronic chip packages in conjugate board level simulation," 19th *IEEE SEMI-THERM Symposium*, pp. 313-318, March 2003.
- [36] V. Szekely and M. Rencz, "Thermal dynamics and the time constant domain," *IEEE Trans. Electron Devices*, vol. 23, no. 3, 2000.

- [37] C.V. Loh, K.C. Toh, D. Pinjala and M.K. Iyer, "Development of effective compact models for depopulated ball grid array packages," *IEEE Electronic Packaging Technology Conference 2000*, pp. 131-137, December 2000.
- [38] T. Nishio, Y. Yamada and K. Koyamada, "A compact modeling approach using a genetic algorithm for accurate thermal simulation," *IEMT/IMC Symposium*, pp. 167-172, April 1998.
- [39] P. Gunupudi, R. Khazaka, T. Smy, D. Celo and M. Nakhla, "Passive parameterized time-domain macromodels for high-speed transmission-line networks," *IEEE transactions on Microwave theory and Techniques* vol. 51, no. 12, December 2003.
- [40] Marta Rencz and Vladimir Szekely, "Dynamic thermal multiport modeling of IC packages," *IEEE Trans. Components and Packaging Technologies*, vol. 24, no. 4, December 2001.
- [41] A. Odabasioglu, M. Celik, and L.T. Pileggi, "PRIMA: passive reduced-order interconnect macromodelling algorithm," *TCAD*, vol. 17, no. 8, pp. 645-653, August 1998.
- [42] Lorenzo Codecasa, Dario D'Amore and Paolo Maffezzoni, "An Arnoldi based thermal network reduction method for electro-thermal analysis," *IEEE Trans. Components and Packaging Technology*, vol. 26, no. 1, March 2003.
- [43] R. Freund, "Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation," *Numerical Analysis Manuscript 98-3-02*, Bell Lab., Murray Hill, NJ, 1998.
- [44] L.M. Silvera, M. Kamen, I. Elfadel and J. White, "A coordinate transformed Arnoldi algorithm for generating guaranteed stable reduced-order models for RLC circuits," in *Tech. Dig. ICCAD*, pp. 2288-2294, November 1996.

- [45] P. Gunupudi, R. Khazaka, M. Nakhla, T. Smy and D. Celo, "Passive Parametrized Time-Domain Macromodels for Microwave Applications", *IEEE Trans. on Microwave Theory and Techniques*, vol. 51, pp.2347-2354, December 2003.
- [46] Pavan Gunupudi, Roni Khazaka, Michel Nakhla, "Analysis of transmission line circuits using multidimensional model reduction techniques," *IEEE Tran. Advanced Packaging*, vol. 25 no. 2, May 2002.
- [47] Xiaoming Guo, "Efficient thermal simulation using model reduction techniques," *Masters thesis Carleton University*, 2004.
- [48] T. Smy, D. Walkey, and S. Dew, "A 3d thermal simulation tool for integrated devices – Atar," *IEEE Trans. Computer-Aided design*, vol. 20, pp. 105–115, January 2001.
- [49] T. Smy, D. Walkey, M. Shams and R.V. Joshi, "Simulation of local heating in VLSI backand structures using Atar," *Advanced Metallization Conference 2001*, pp. 355, October 2001.
- [50] T. Smy, D. Walkey and S.K. Dew, "Transient 3D heat flow analysis for integrated circuit devices using transmission line matrix method on a quad tree mesh," *Solid State Electronicse*, vol. 45, no. 7, July 2001.
- [51] P. Gunupudi and M. Nakhla, "Multi-dimensional model reduction of vlsi interconnects," *Proc. IEEE CICC*, pp. 499-502, May 2000.
- [52] R. Achar and M. Nakhla, "Simulation of high-speed interconnects," *IEEE Proc.*, vol. 89, pp. 693-728, May 2001.

- [53] T. Tank and M. Nakhla, "Analysis of high-speed VLSI interconnects using Asymptotic Waveform Evaluation," *IEEE trans. Computer-Aided Design*, pp. 2107-2116, March 1992.
- [54] L.T. Pillage and R.A. Rohrer, "Asymptotic Waveform Evaluation for timing analysis," *IEEE trans. Computer-Aided Design*, vol. 9, pp. 352-366, April 1990.
- [55] J.E. Bracken, V. Raghaven and R.A. Rohrer, "Interconnect simulation with Asymptotic Waveform Evaluation," *IEEE trans. Circuits and Systems*, vol. 39, pp. 869-878, November 1990.
- [56] J.Y. Lee, X. Huang and R.A. Rohrer, "Pole and Zero sensitivity calculation in Asymptotic Waveform Evaluation," *IEEE trans. Computer-Aided Design*, vol. 11, pp. 586-597, May 1992.
- [57] M.S. Nakhla, E. Chiprout, R. Achar and R. Khazaka, "Recent progress in the simulation of high speed VLSI interconnects," *IEEE Proc. conference on Electrical Performance of Electronic Packaging*, pp 193-196, October 1995.
- [58] E. Chiprout and M. Nakhla, "Analysis of interconnect networks using Complex Frequency Hopping (CFH)," *IEEE trans. Computer-Aided Design*, vol. 14, pp. 186-200 February 1995.
- [59] P. Feldman and R.W. Freund, "Efficient linear cut analysis by Padé via Lanczos process," *IEEE trans. Computer-Aided Design*, vol. 14, pp. 639-649, May 1995.
- [60] Kevin. J. Kerns and Andrew T. Yang, "Preservation of passivity during RLC network reduction via Split Congruence Transformatoin," *IEEE Trans. Computer-Aided Design*, vol. 17, no. 7, July 1998.

- [61] Pavan Gunupudi, "Circuit-Reduction techniques and application to high-speed systems and interconnects," *Ph.D thesis*, 2002.
- [62] Roni Khazaka, "Projection based techniques for the simulation of RF circuits and high speed interconnects," *Ph. D thesis*, 2002.
- [63] W.C. Elmore, "The transient response of damped linear networks with particular regard to wide-band amplifiers," *Journal of Applied Physics*, pp. 55-63, January 1948.
- [64] J. Rubinstein, P. Penfield and M. Horowitz, "Signal delay in RC trees," *IEEE trans. Computer-Aided Design*, pp. 202-211, July 1983.
- [65] J.W. Demmel, *Applied Numerical Linear Algebra*, SIAM Publishers, Philadelphia, PA, 1997.
- [66] H. Carslaw and J.L. Jeager, *Conduction of Heat in Solids*, Second Edition, London:Oxford University Press, 1959.
- [67] L.C. Thomas, *Heat Transfer-Profesional Version*, Second Edition, Tulsa, Oklahoma:Capstone Publishing Corporation, 1999.
- [68] M. Necati Ozisik, *Heat Conduction*, Wiley, New York, 1980, (chapters 2 and 6).
- [69] Z. Johnson, K. Ramakrishna, B. Joiner and M. Eyman, "Thermal sub-modeling of the wirebonded plastic ball grid array package," 13th *IEEE SEMI-THERM Symposium*, 1997.
- [70] I.M. Elfadel and D.D. Ling, "A block rational Arnoldi algorithm for multiport passive model-order reduction of multiport RLC networks," *IEEE Proc. ICCAD-97*, pp. 66-71, November 1997.

- [71] Pavan K. Gunupudi, Michel Nakhla and Ramachandra Achar, "Multi-point multi-port reduction of high-speed distributed interconnects using krylov-space techniques," in *Proc. IEEE International Symposium on Circuits and Systems*, June 1999, vol. 6, pp. 242–245.