Fast Parasitic Extraction and Simulation of Three-dimensional Interconnect via Quasistatic Analysis

by

Mattan Kamon

B.S. Engineering Science, Pennsylvania State University (1991)
M.A. Mathematics, Pennsylvania State University (1991)
S.M. Electrical Engineering, Massachusetts Institute of Technology (1994)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1998

© Massachusetts Institute of Technology 1998
All rights reserved

Signature of Author_____
Department of Electrical Engineering and Computer Science
January 8, 1998

Certified by_____
Jacob K. White
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Professor of Electrical Engineering and Computer Science
Chairman, Department Committee on Graduate Students

Fast Parasitic Extraction and Simulation of Three-dimensional Interconnect via Quasistatic Analysis

by

Mattan Kamon

Submitted to the Department of Electrical Engineering and Computer Science
on January 8, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

As VLSI circuit speeds have increased, the need for accurate three-dimensional interconnect models has become essential to accurate chip and system design. Many of these interconnect structures are small compared to a wavelength, and much work has been directed at rapidly solving for the inductance and capacitance of these structures.

The first part of this thesis involves further development of efficient inductance extraction and simulation. First, a method of nonuniformly discretizing planar structures is developed which allows for hierarchical descriptions and modification while still enforcing current conservation. Also, a possible alternate preconditioner to accelerate the iterative method is described. Finally, the most significant contribution for this part is the numerically robust Arnoldi approach for efficiently generating compact, or *reduced order* models for the inductance problem.

Even though much work has been directed toward solving separately for the inductance and capacitance of these structures, these two quantities are not necessarily decoupled, and for higher frequencies a distributed model is necessary. The second part of this thesis develops fast algorithms for solving this full quasistatic Maxwell's equations to effectively capture this "coupled" or distributed capacitance and inductance. The technique employed for extraction is an integral equation approach for modeling the impedance of interconnect structures accounting for both the charge accumulation on the surface of conductors and the current traveling along conductors. While such computations by themselves and not new, a technique is proposed for which it is possible to generate guaranteed passive reduced order models for efficient inclusion in a circuit simulator such as SPICE.

In their basic form, these methods require direct LU factorization of very large linear systems. Such factorization is impractical due to the $O(n^3)$ computation time and $O(n^2)$ storage costs. Iterative techniques are then applied which can exploit fast potential solvers such as the Fast Multipole method to bring the cost of model generation to $O(n)$ operations and storage. For a modest problem size, the iterative methods show nearly 2 orders of magnitude speed up in computation time and an order of magnitude less memory than direct factorization.

Thesis Supervisor: Jacob K. White
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

The cover page may have only one author, but both this dissertation and the author himself have benefited from the contributions of many over the years.

Foremost, I thank Jacob White, my advisor, whose curiosity and desire to understand was irreplaceable and showed me what research can be. His patience and positive attitude provided an environment where the drive to learn and accomplish came only from within.

Next, I want to thank all the members of the 8th floor VLSI-CAD group, past and present. The technical interaction was key to my development as a graduate student. In particular I thank Miguel Silveira, Joel Phillips, and Nuno Marques who have worked closely with me and whose contributions can be attributed to a number of the sections of this thesis. I also thank the original research group whose warmth "convinced" me to come to MIT and my officemates over the years who made the day to day engaging and enjoyable.

I also thank my thesis committee, Professors Jeffrey Lang and Munther Dahleh, for taking the time to give me valuable comments on this document.

I could not have gotten this far without friends. This includes all of the 8th floor. From the Portuguese "mafia" and their associates to one who has chosen the dark side. You know who you are.

To the people of Edgerton House that gave me a great feeling of community. From the Thursday night lineup and Star Trek watchers to a certain self-made chef and oenophile. And then there were the sports. Of course much of that was made possible only thanks to my apartmentmates of 135 that made our place a real home from start to finish. Thanks.

Of course I can't leave out the couple who often reminded me that weekends (especially Friday happy hours) are for relaxing. Some even credit them for my Masters degree not taking longer than it did.

And my fondest gratitude goes to my source of greatest support, my wife, Susanna. It takes a great deal of patience and devotion to become newly married to someone in the final months of his thesis. For this love, support, and confidence I am most grateful.

◇ ◇ ◇

This work was supported by DARPA, NSF, SRC, Harris Semiconductor, and IBM.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Motivation

In the days of the first integrated circuits, the speed of the circuit was largely determined by the switching speed of individual transistor devices. The characteristics of the path a signal must travel from device to device were negligible. As speeds increased however, the delay incurred by this path, or interconnect, became critical to accurate chip and system design. More recently, not only have self characteristics such as delay become important, but also electromagnetic coupling to other nearby interconnect. For this reason, generating accurate models of these undesirable, or parasitic, properties has become essential to chip and system design.

Measurement has always been an approach to modeling interconnect, however such an approach can be difficult and time consuming, especially on-chip, and thus in recent years much effort has been devoted to the fast and accurate computation of interconnect models directly from Maxwell's equations. For many portions of a design, the significant interconnect may be long and uniform enough to be modeled using a two-dimensional approximation and transmission line theory. Unfortunately, discontinuities in this 2D interconnect, such as vias through planes, chip-to-board-connect and board-to-board connectors, require full three dimensional modeling. To account for these, the designer, or CAD tool, must glue together 2D and 3D models which is cumbersome. Additionally, as circuit density increases, these discontinuities become more prevalent, and full 3D modeling is the only course.

Many of these structures are small compared to a wavelength, and thus a low frequency, or quasistatic, approximation of Maxwell's equations is appropriate. This thesis describes the development of fast algorithms for the modeling and simulation of compli-

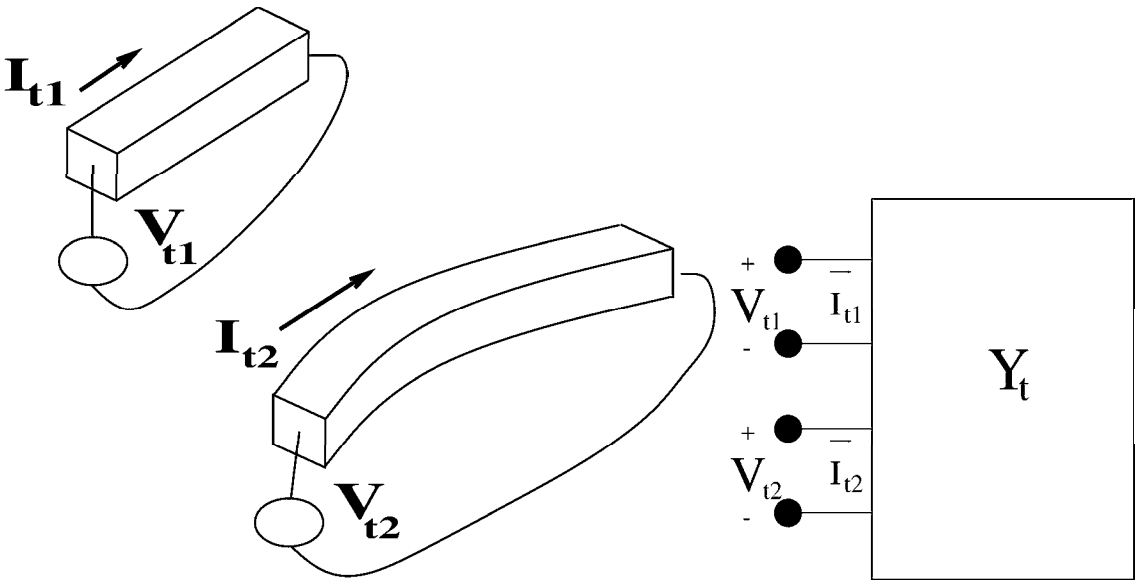


Figure 1-1: a) Two conductors, each with a pair of terminals b) The abstract port representation of the two conductors.

cated three-dimensional interconnect in this regime.

1.2 Problem Description

The goal of parasitic interconnect extraction for a set of conductors is to determine the relation between the currents and the voltages at the terminals (or ports) of the conductors. For a k terminal-pair problem in the sinusoidal steady-state at the frequency ω, this relation is described by the admittance matrix, $Y_t(\omega) \in \mathbb{C}^{k\times k}$ where

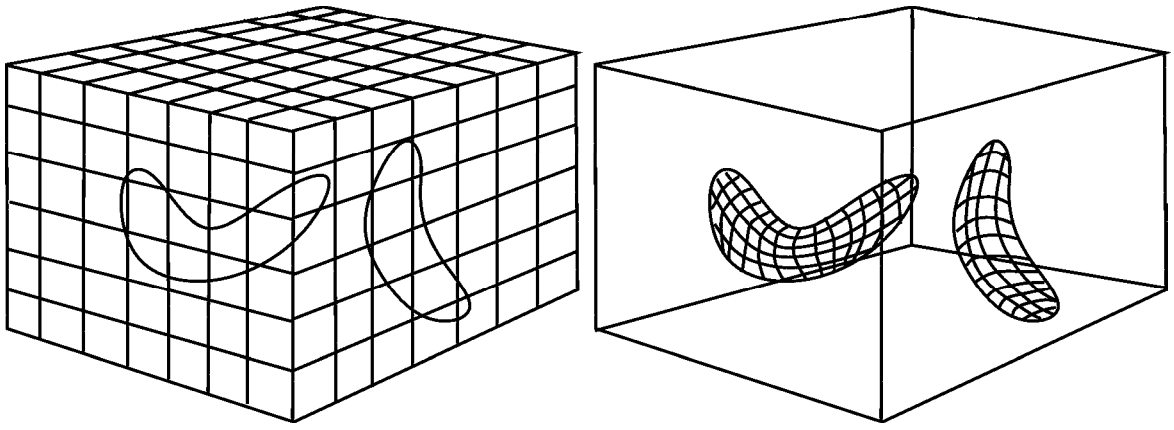
$$Y_t(\omega)V_t(\omega) = I_t(\omega), \tag{1.1}$$

where $I_t, V_t \in \mathbb{C}^k$ are vectors of the terminal current and voltage respectively [10].

To understand the connection to Maxwell's equations, consider a geometry consisting of two terminal pairs as shown in Fig. 1-1. For this problem,

$$Y_t(\omega) = \begin{bmatrix} Y_{11}(\omega) & Y_{12}(\omega) \\ Y_{21}(\omega) & Y_{22}(\omega) \end{bmatrix}. \tag{1.2}$$

From (1.1), we see that column one of Y_t can be computed by computing the currents I_{t1} and I_{t2} resulting from setting $V_{t1} = 1$ and $V_{t2} = 0$. Computing these currents given a set of potentials is accomplished through solutions for the electromagnetic fields governed by Maxwell's equations. A common approach is to apply a finite difference or



(a) (b)

Figure 1-2: Finite difference discretization of all of space versus a boundary element method discretizing only the surface. Reprinted with permission of Mike Chou

finite element method to the governing equations in differential form. Such an approach requires the discretization of all of 3-D space which can be cumbersome and generate very large linear systems. Instead, consider an integral equation approach, which requires discretization of only the conductors. This is illustrated in Figure 1-2 for the capacitance problem which can be solved via a boundary-element approach requiring the discretization of only the surface[12, 56]. Similarly, the inductance problem can be treated via a volume-element approach which requires the discretization of the interior of conductors only [54, 70]. Unfortunately, such methods generate dense matrix problems which, if solved via standard direct methods such as Gaussian Elimination, grow in computational cost on the order of n^3, and in memory as $O(n^2)$, where n is the number of elements into which the problem is discretized. For modern packaging structures for which n is 10^4 to 10^5, these growth rates are prohibitive. For instance, $n = 10^4$ corresponds to nearly 1 gigabyte of memory and hours for a single factorization at 100 MFlops (million floating point operations per second).

Over the past decade, solving the dense matrices associated with boundary or volume-element methods has been made substantially more efficient through the use of iterative solution techniques accelerated by "fast-multipole" or other matrix sparsification algorithms. This combined approach reduces the computational (and storage) cost of using boundary and volume-element methods to nearly $O(n)$ [21].

Fast multipole algorithms have been applied for capacitance and inductance extraction as described in [46, 48, 29, 35]. However, for general tools for fast parasitic extraction,

issues apart from these fast sparsification techniques must be addressed. In this thesis, we investigate these issues and develop algorithms which are shown to be two orders of magnitude faster and consume an order of magnitude less memory than direct approaches for $n \approx 10^4$.

The developments of this thesis are divided into two parts. Part I involves further development of efficient algorithms for inductance extraction and simulation and Part II develops fast algorithms for solving the full quasistatic Maxwell's equations to effectively capture the "coupled" capacitance and inductance.

To begin, Chapter 2 is a motivational chapter, describing the regimes for which the algorithms of the thesis are useful. It also demonstrates the need for tools capable of handling entire interconnect structures which generate these large linear systems.

To start Part I, Chapter 3 gives a detailed description of the previous work on multipole-accelerated inductance extraction in [35]. The detail is included not only to put the rest of this part in context, but also because many of the contributions are vital to Part II. However, the methods presented in this chapter are not the end of the story. In particular, methods of efficient discretization leave room for further improvement and in the first section of Chapter 4 we develop a method of nonuniformly discretizing planar structures. The approach allows hierarchical description and modification while still enforcing current conservation. Using such an approach is shown to reduce the size of the linear systems by at least an order of magnitude.

Issues related to the fast convergence of the the iterative matrix solution algorithm also leave room for improvement and in the second section of Chapter 4 an alternate preconditioner to accelerate the iterative method is described.

The algorithms of Chapter 3 and 4 capture the frequency dependent resistance and inductance at specific frequency points only, however, often the end use of interconnect models is for simulation with nonlinear devices. Such simulation must be performed in the time domain and knowledge of the resistance and inductance at individual frequencies is not adequate. Methods of efficiently generating compact yet accurate time domain descriptions of the interconnect are necessary.

Methods of generating compact, or *reduced order*, models from larger dynamic linear systems has been explored in many disciplines, however an approach which is computationally efficient for dense systems whose rank is on the order of 10^4 has only recently been explored. Perhaps the most important development of Part I is the development of the Arnoldi approach in Chapter 5 for efficiently generating numerically robust reduced order models.

The methods of Part I are concerned with inductance alone. However, even though

much work has been directed toward solving separately for the inductance and also capacitance of these structures [12, 5, 7, 6, 48, 35], these two quantities are not necessarily decoupled. While an integral equation approach has been developed for this coupled problem [55], the acceleration and efficient generation of compact models for complicated three dimensional geometries has not been addressed.

Part II of this thesis develops fast algorithms for solving the full quasistatic Maxwell's equations to effectively capture this "coupled" capacitance and inductance. This full quasistatic regime which we term "electromagnetoquasistatics" (EMQS), can be viewed as treating the inductance and capacitance in a distributed manner. The technique employed for extraction is an integral equation approach accounting for both the charge accumulation on the surface of conductors and the current traveling along conductors. In this work, an approach similar to the standard integral equation approach for interconnect analysis is used to formulate a system of equations amenable to techniques of model order reduction discussed in Chapter 5. But generating the low order model requires many solves of a dense linear system. The approaches developed for fast solution are the crux to fast and efficient model order reduction.

Many of the sections of this thesis also exist in published form. The motivational example of Section 2.2.2 appears in [31]. The inductance algorithm of Chapter 3 appears in [35] and the theory for the preconditioners in [34]. Some of the reduced order modeling work of Chapter 5 appears in [62]. All of Part I (Chapters 3, 4, and 5) is implemented in C code available from `ftp://rle-vlsi.mit.edu/pub/fasthenry`. Sections of Part II appear in [33] and [32].

2

Motivation

Through examples, this chapter places in context the contributions of this thesis and also demonstrates the necessity of accurate parasitic extraction tools. First in Section 2.1 we generate a simple circuit model for two pins of connector to demonstrate where such simple models break down. Then, in Section 2.2.2, the entire package of an analog IC is coupled into circuit simulation to demonstrate the need for accurate models to facilitate fast and accurate designs.

2.1 Placing this work in context

Through a small example, this section demonstrates the application regime for the developments in this thesis. To motivate the developments of subsequent chapters, the example will follow a common approach for interconnect modeling at low frequencies and demonstrates where such an approach breaks down.

For simplicity, consider only two pins of the eighteen pin connector built by Teradyne, Inc. shown in Figure 2-1. We wish to accurately model the admittance seen by a source when the pins connect the source to a resistive load as shown in Figure 2-2.

A common approach to generating a simple circuit model is to avoid solving the full quasistatic Maxwell's equations by instead computing first the resistance and inductance, that is, computing Y_t under the magnetoquasistatic assumption that the displacement current is negligible, and then second, computing the capacitance by solving an electrostatic problem. With these R, L, and C values, a common and simple circuit model for the system is the "Γ" model shown in Figure 2-4. To make a symmetric model for the interconnect, the π or T models shown in Figures 2-3 and 2-5 are common. Such circuit models are accurate for low frequencies for all values of R_{load}, but for higher frequencies, the models break down. For instance, in Figure 2-6, the resistance and inductance were

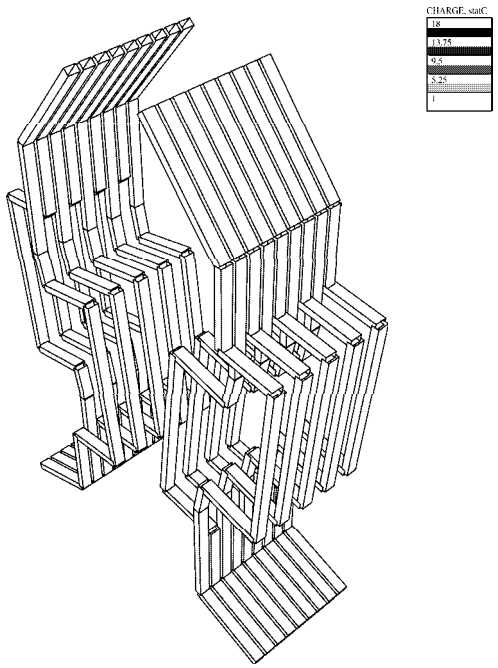


Figure 2-1: An eighteen pin backplane connector built by Teradyne, Inc.

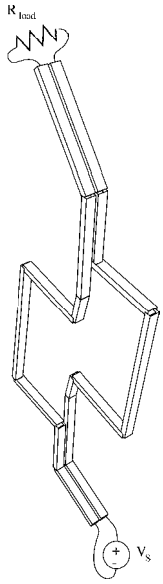


Figure 2-2: Two pins of the eighteen pin connector with terminating load.

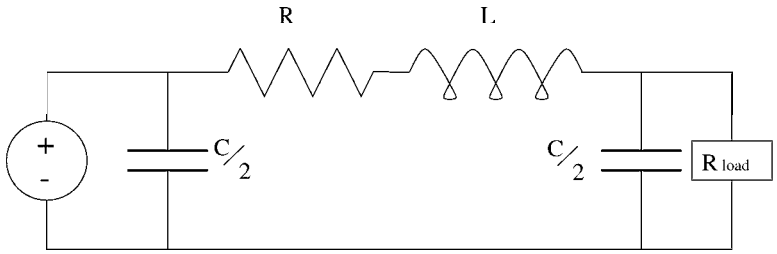


Figure 2-3: π circuit model

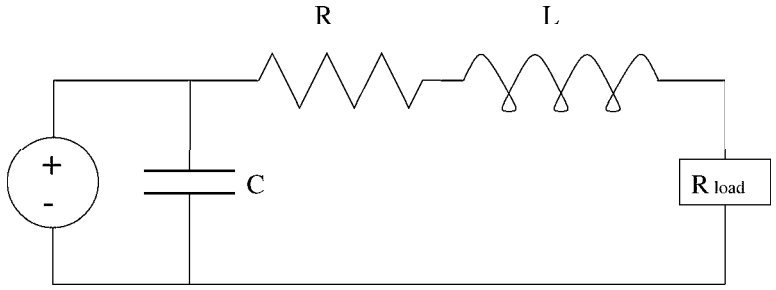


Figure 2-4: Γ circuit model

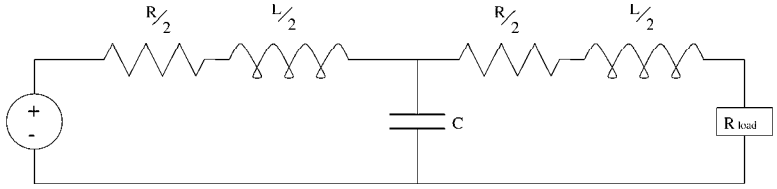


Figure 2-5: T circuit model

computed for zero frequency (DC) and R_{load} was set to zero. The choice of R_{load} was made to emphasize the inductance and the exact values were computed by methods of Chapter 6. Notice that all the models match the exact value until around 10 MHz. The models eventually break down due to skin effects. The current begins crowding toward the surface of the conductor, increasing the resistance and decreasing the inductance[70]. To improve the response at higher frequencies, consider using R and L computed at $f = 10$ GHz as shown in Figure 2-7. Now all the models match up to 300 MHz, but are inaccurate below 10 MHz.[1]

From these figures, it is apparent that a model which captures the frequency dependence of the resistance and inductance is necessary for accurate models from zero to 300 MHz. Chapter 5 describes numerically robust techniques for generating low order models to capture these effects over such a frequency range.

[1]Note that in this example the frequency dependent effect on the inductance is small. For an example of stronger frequency dependence, see Section 3.4.

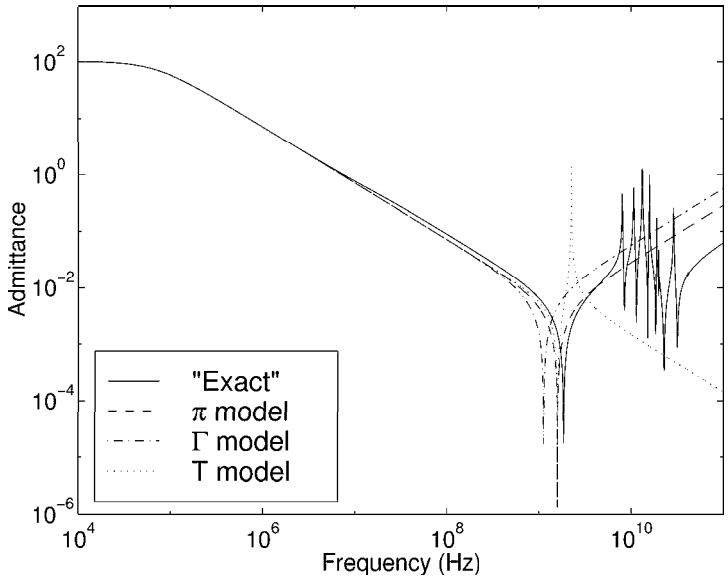


Figure 2-6: Response of 2 pins with $R_{load} = 0$ using various simple circuit models derived for $f = 0$.

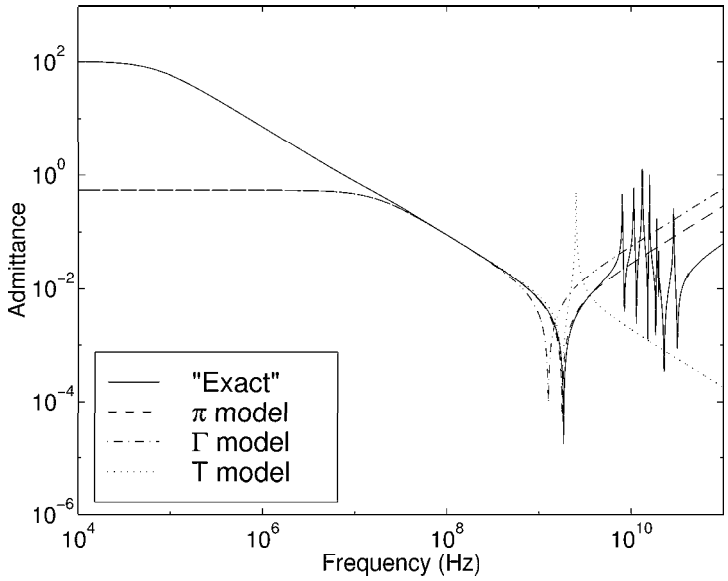


Figure 2-7: Response of 2 pins with $R_{load} = 0$ using various simple circuit models derived for $f = 10$ GHz.

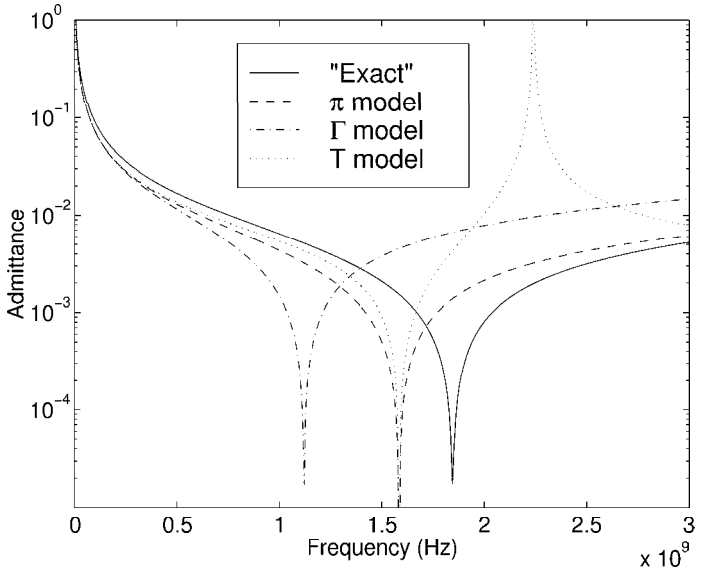


Figure 2-8: Response of 2 pins with $R_{load} = 0$ using various simple circuit models derived for $f = 0GHz$

As circuit speeds have increased, models for such interconnect beyond 300 MHz have become important. At higher frequencies, the interconnect becomes self-resonant and such sharp features in the response must be captured accurately. For the case of these two pins, the capacitance is high enough that the first resonance appears well within the quasistatic regime.

To investigate these resonances, consider the response of the structure in the gigahertz range as shown in Figure 2-8. Note that at the resonance points, the structure, which is 1.8 cm long, is still small compared to a wavelength. The lumped models shown were generated using R and L at $f = 0$ Hz. From the figure, the Γ model has missed the resonance by over 0.5 GHz, and the π and T models are better, but still inaccurate. Additionally, the T model has an extra peak at $2.25GHz$. Such deviation are unacceptable as we will see in the next section. Using the inductance and resistances at $f = 10$ GHz, all the models come slightly closer to the resonance but are still inaccurate as shown in Figure 2-9. The choice of $f = 10$ GHz for computing R and L for this example was not important. Choosing R and L at the resonant frequency produces similar results because for this example the frequency dependence on the admittance is small.

Of the three models above, the π model performs best. This is not surprising considering the structure "looks" like a π model. Starting at the source in Figure 2-2, the conductors are very closely spaced, indicating a high capacitance, then they separate, enclosing a large "loop" for which the flux, and thus the inductance, is high, and then

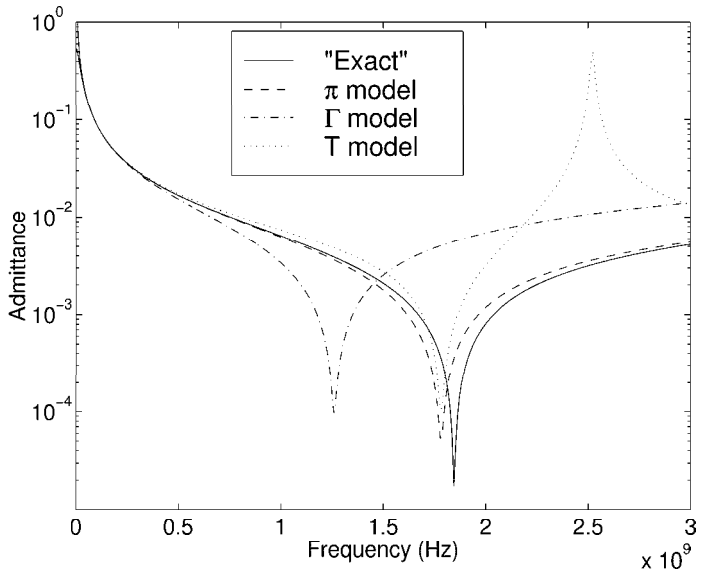


Figure 2-9: Response of 2 pins with $R_{load} = 0$ using various simple circuit models derived for $f = 10GHz$

the conductors come together again for a large capacitance.

Such heuristic knowledge of the geometry of the conductor may allow a designer to write down a specific model for each specific section of interconnect. Such an approach might result in close results as shown by the above example. However, this approach is cumbersome since there can be many sections of interconnect. Additionally, it may be difficult to derive a simple model such as the π model for a complex three-dimensional section of interconnect and even if possible, the model is still inaccurate as shown in Figure 2-9. Lastly and perhaps most importantly, is that the simple models can capture at most only one resonance and thus are inaccurate beyond the first resonance. For these reasons, it is necessary to develop fast and efficient methods which accurately capture the admittance directly, taking into account both capacitive and inductive effects simultaneously. This electromagnetoquastatic (EMQS) problem is the topic of Chapter 6. Also we find in Appendix A that for certain geometries common to interconnect analysis, the quasistatic approximation is useful beyond the short-compared-to-a-wavelength regime. For the connector example, this corresponds to the behavior including the higher resonances of Figure 2-6. This is a regime for which the simple single resonance of the π model can never capture.

To summarize, Figure 2-10 marks the different regimes discussed in this thesis. In region A, a single lumped R and L are adequate and Chapter 3 and 4 discuss fast methods of computing an appropriate R and L for models like in Figure 2-3. For models valid

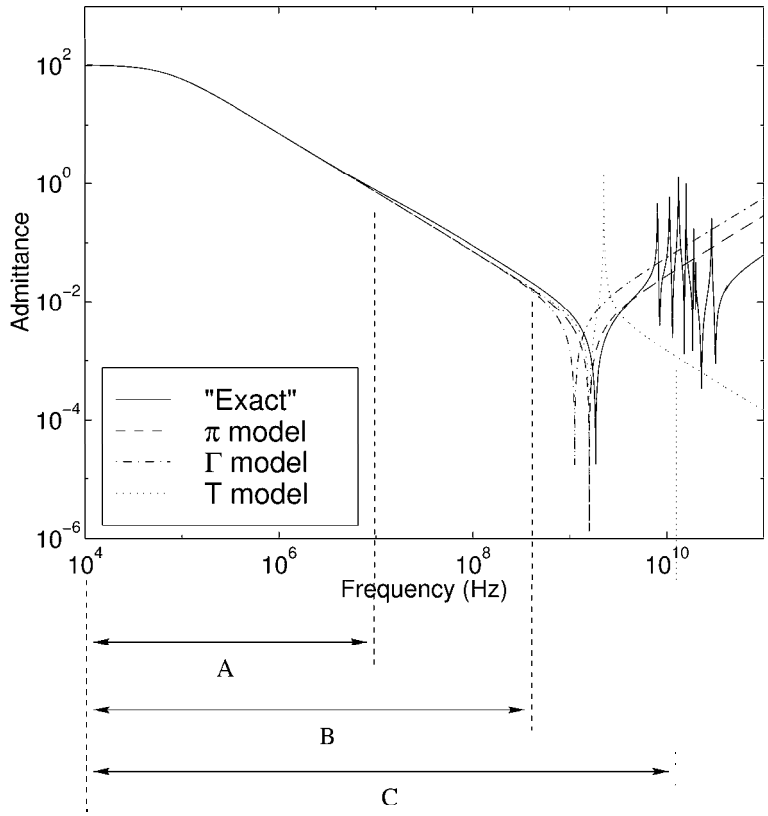
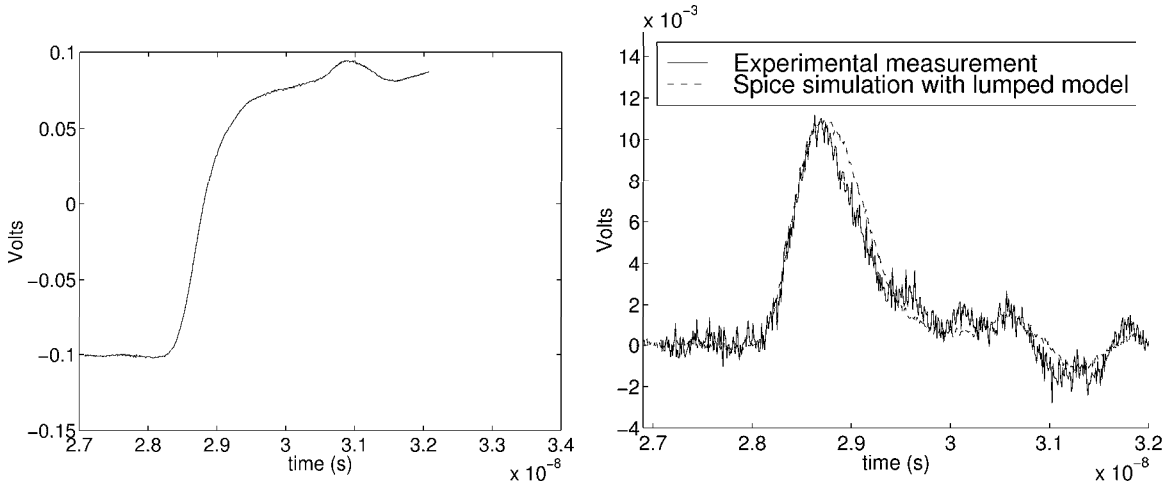


Figure 2-10: The different regimes discussed in this thesis. A: Single lumped inductance, Chapter 3 and 4, B: Frequency dependent inductance and model order reduction, Chapter 5, C: Coupled inductance and capacitance, Part II

from zero frequency to the higher frequencies of region B, the frequency dependence of the resistance and inductance must be captured and the single lumped R and L are not adequate. In Chapter 5 we develop numerically robust methods for generating low order models valid over this larger range. In region C, capacitance and inductance cannot be separated if the resonant behavior is to be captured. In Part II we develop fast methods to generate low order models of this interconnect which accurately captures these resonances.

Note that the frequency intervals of each regime are geometry dependent. For instance, if the pins of the connector were thinner, the skin effect would become important at a higher frequency and region A would have been wider. Similarly, if the conductors were spaced farther apart, region B would be wider because the capacitance would be smaller and the first resonance would occur at a higher frequency.



(a) (b)

Figure 2-11: a) A pulse with a 500 ps rise time. b) Crosstalk on an adjacent pin

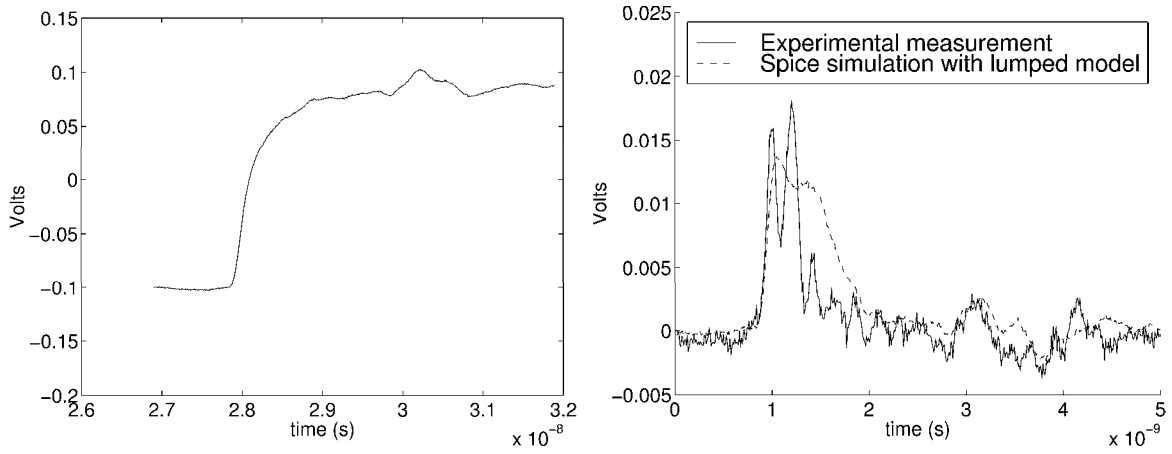
2.2 Real Examples from Experiment

Before beginning the developments of subsequent chapters, this section gives two examples to show the importance of capturing the frequency domain with such accuracy. As will be shown in Section 2.2.1, for digital circuits, the simulated response may be inaccurate if some harmonic of the operating frequency falls at a resonance . For analog chip design, a certain subsystem may be designed to have minimum impedance at a certain frequency. Inaccurate interconnect models can results in poor performance of the fabricated devices compared to simulation as will be demonstrated in Section 2.2.2.

2.2.1 A Digital Example

For the backplane connector of Figure 2-1, consider exciting one of the middle pins with step in voltage with a 500 ps rise time as shown in Figure 2-11a. Such a pulse can induce a voltage on adjacent pins and quantifying this switching noise is important for accurate system design. For this pulse, a fully coupled 18 pin version of the π circuit of Figure 2-3 captures the behavior accurately compared to experimental measurement as shown in Figure 2-11b. One might expect that such behavior is accurately captured since the pulse has a strong frequency component at $1/(500\text{ps}) = 2\text{GHz}$ which is a frequency where the π model is still accurate.

Next, consider moving to a much faster rise time of 100 ps. As seen in Figure 2-12,



(a) (b)

Figure 2-12: a) A pulse with a 100 ps rise time. b) Crosstalk on an adjacent pin

the pulse excites a resonance as can be seen by the jagged response of the measured data. The simple π model is now far from accurate. To understand this, note that $1/(100\text{ps}) = 10\text{GHz}$ which is beyond the simple single resonance that the π model is able to capture. Thus, it becomes necessary to model the fully distributed "electromagnetoquasistatic" case to capture these higher resonances accurately.

2.2.2 Modeling of an analog chip and its package

As analog circuits operating at RF frequencies are integrated on silicon, the *entire* package and interconnect structures become an integral part of the functional design used for matching and tuning networks. To demonstrate this point, this section observes the effect on simulation of not including models for the entire package of an analog IC package. The necessity of accurate package models thus motivates the development of fast parasitic extraction tools.

In particular, for a 2.4 GHz RF to IF Converter integrated circuit under development at Harris Semiconductor, we show that modeling the resistance, capacitance, and inductance of the package leads, bondwires, die attach plane, and on-chip interconnect is required for accurate circuit simulation. The HFA3624 shown in Figure 2-13 is a bipolar device that provides up/down conversion between a 2.4 - 2.5 GHz RF signal and 40 - 400 MHz IF operation for a wireless communication chipset.

The entire package is shown in Figure 2-14. The chip is the small rectangle in the center, with the bondwires rising from the chip and bonded to either the rectangular die-

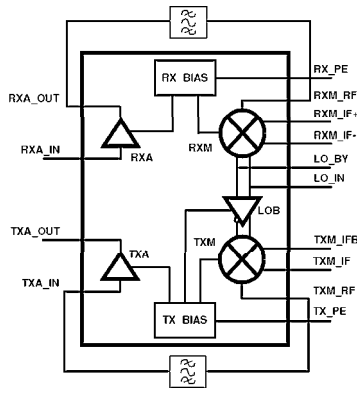


Figure 2-13: HFA3624 RF/IF Converter Block Diagram

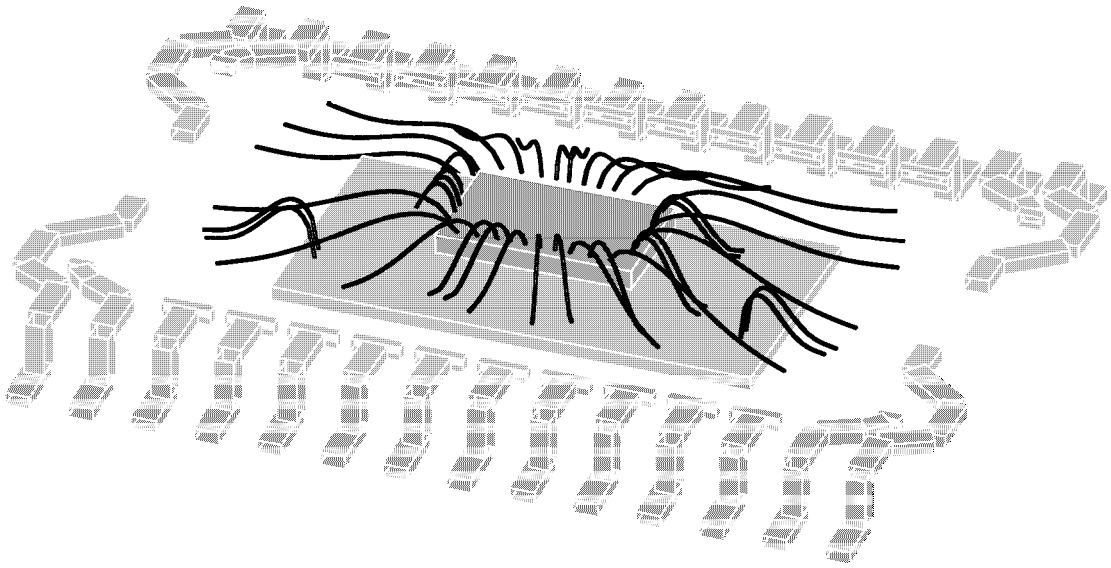
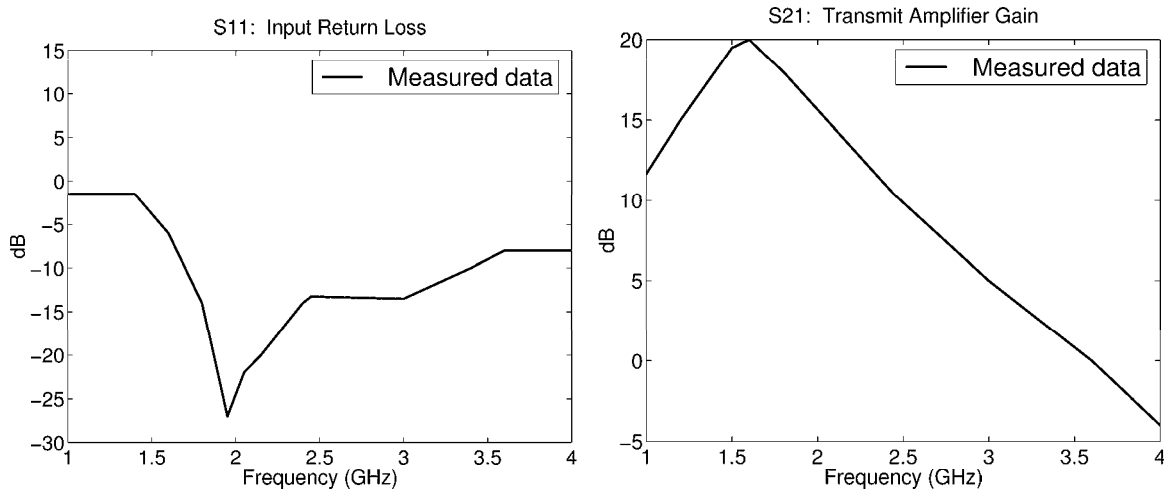


Figure 2-14: The package for an RF integrated circuit

attach plane below or extending out the package pins on the outside of the figure. The package pins are drawn a distance away from the package for illustration. The bondwires which do not terminate on the die attach would terminate on the package pins.

The analysis will focus on accurately predicting the gain, S_{21}, and return loss, S_{11}, as a function of frequency of one section of the converter, the transmit amplifier or TXA. While the simulation focuses on one section of the circuit, the interconnect external to the die will be modeled for the entire device to more fully capture coupling effects associated with the packaging.

The device was designed to have a maximum gain and minimum return loss at the operating frequency of 2.45GHz. The first design was performed without consideration of the package parasitics and the first pass out of fabrication showed very poor results as



(a) (b)

Figure 2-15: Experimental Measurement for the TXA. Measurement is off from intended design of minimum S_{11} and maximum S_{21} at 2.45 GHz

shown in Figure 2-15. Note that if package parasitics had been accounted for originally, this very expensive extra fabrication iteration might have been avoided.

In the following subsections we investigate the effects of modeling the package leads and bondwires, the on-chip interconnect, and the grounded die attach plane, respectively.

Modeling of the Package and Bondwires

The HFA3624 is packaged in a 28 lead SSOP as was shown in Figure 2-14. The die attach (DA) plane is used as a ground return and is fused to some of the package leads as ground connections. The chip internal ground connections are made by down bonding from bond pads to the DA.

The capacitance and inductance of just the package were assembled in a π-type model as illustrated in Figure 2-16 where the L1 terminal is the external connection to the PCB and the P1 terminal is the internal connection to the bonding post. The capacitance to ground of each lead was divided into the two capacitors, C1Ga and C1Gb. The self inductance was similarly divided between L1a and L1b. The small series resistance was lumped into a single element R1, and a large resistance, R1dc, was provided to allow the lead to be left unconnected with a DC path to ground. The capacitance between the internal end of the lead and the DA was captured in C1DA. The remaining capacitors represent the significant coupling capacitances between the package leads. The mutual inductance was also modeled but is not shown here.

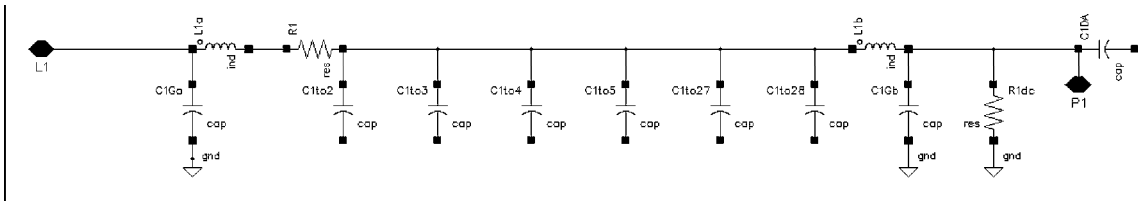
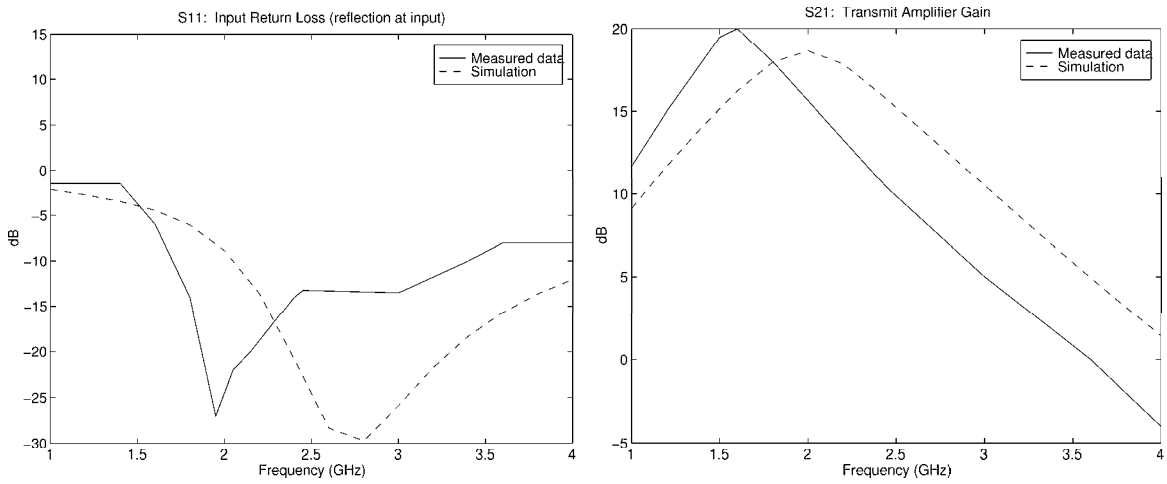


Figure 2-16: Lumped Element Model of a Single Package Lead



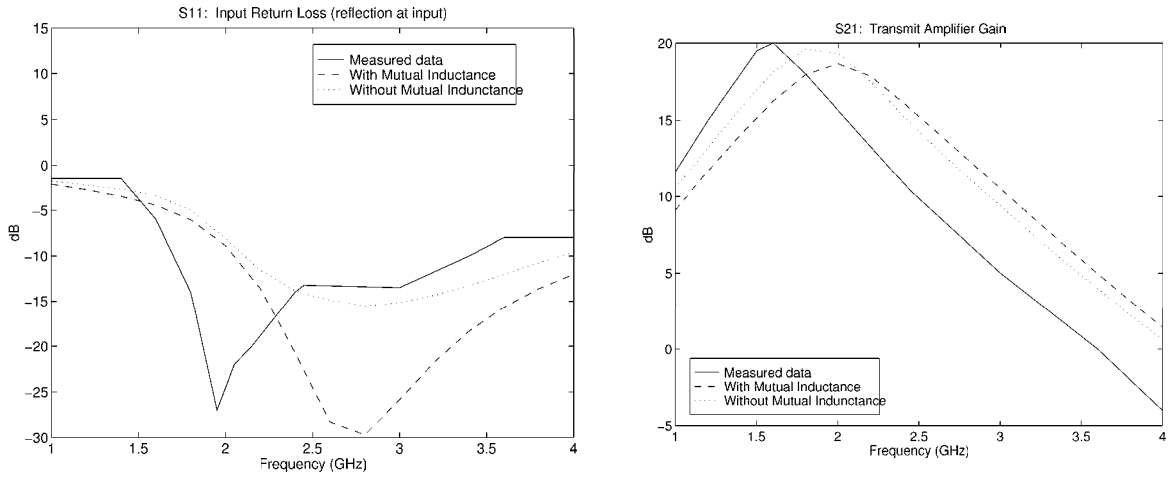
(a) (b)

Figure 2-17: Simulation with Package and Bondwire Models

The resistance and inductance of the bondwires were computed, but due to the small surface area of the bondwires compared to the separation, capacitance was neglected and a simple network of RL circuits was used for the lumped element model.

At this point all connections to the DA were modeled as a single node. The distributed currents in the DA will be modeled later in Section 2.2.2. The simulation results with only package and bondwire models matched poorly to measurement, as shown in Figure 2-17.

To appreciate the influence of parasitic inductance of the package leads and bondwires versus the intentional components for tuning, consider simulation with and without the mutual inductance terms in both models. This is shown in Figures 2-18. It can be seen that while neither result gives good agreement with measurement at this point, the parasitic inductive effects have a strong influence on the waveforms. Interestingly, even though the gain seems to have improved, this is merely coincidental as will be shown later.



(a) (b)

Figure 2-18: Simulation with Package and Bondwire Models, with and without Mutual Inductance

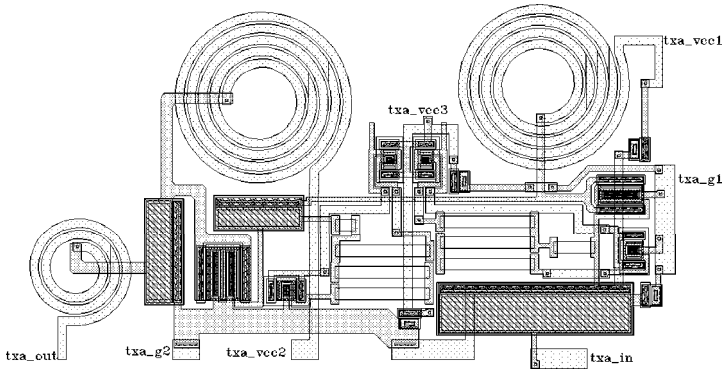
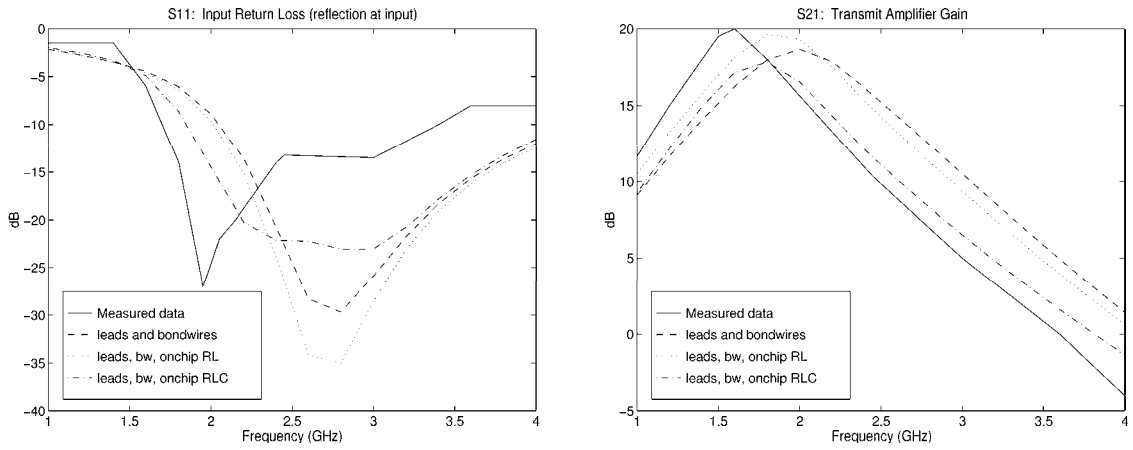


Figure 2-19: Layout of the TXA

Modeling of the On-Chip Interconnect

As observed from the results of the previous section, it is clear that with only package and bondwire models the simulation is unacceptably far from the measured data. While the final goal is to match the measured data over the whole frequency range, for this product we are especially interested in the results at 2.4 GHz. At this frequency, the predicted gain is 6 dB above the actual gain and the input return loss curve is significantly shifted in frequency. Next we show improvements in the simulation by including models for the on-chip interconnect. The layout of the TXA with the interconnect lines of interest is shown in Figure 2-19.

From the influence of the inductance seen in Figure 2-18 we can expect that the



(a) (b)

Figure 2-20: Simulation with Package, Bondwire, and On-chip Interconnect Models

on-chip interconnect inductance will also play a significant role. For instance, while the length of a given segment of on-chip interconnect is considerably shorter than a bondwire, the full path of current can be on the order of the bondwire length and therefore cannot be neglected. To compute the inductance and resistance, the geometries of selected lines were extracted from the layout in a partially automated process and read into the inductance solver. In some cases, the interconnect inductance values were found to be as much as ten percent or more of the fabricated spiral inductor values. In addition, coupling ratios for simulating mutual inductance were as high as 50 percent between adjacent lines.

For capacitance calculation, the fourteen largest sections of interconnect were modeled. For the process used in this product, it has been shown that in the gigahertz range, the epitaxial layer is too resistive to act as a ground reference for capacitance calculation [43]. For this reason, the doped bulk substrate was used as the ground reference. The maximum capacitance to ground was 0.17 pF and the maximum coupling capacitance was 0.03 pF.

The effect on simulation of adding just the on-chip RL and then also the on-chip capacitance is shown in Figure 2-20. The added capacitance greatly improved the prediction of the gain at 2.4 GHz. For the return loss the improvement is still not as drastic. The simulated minimum is approaching the measured data, but the overall S_{11} curve is still very inaccurate.

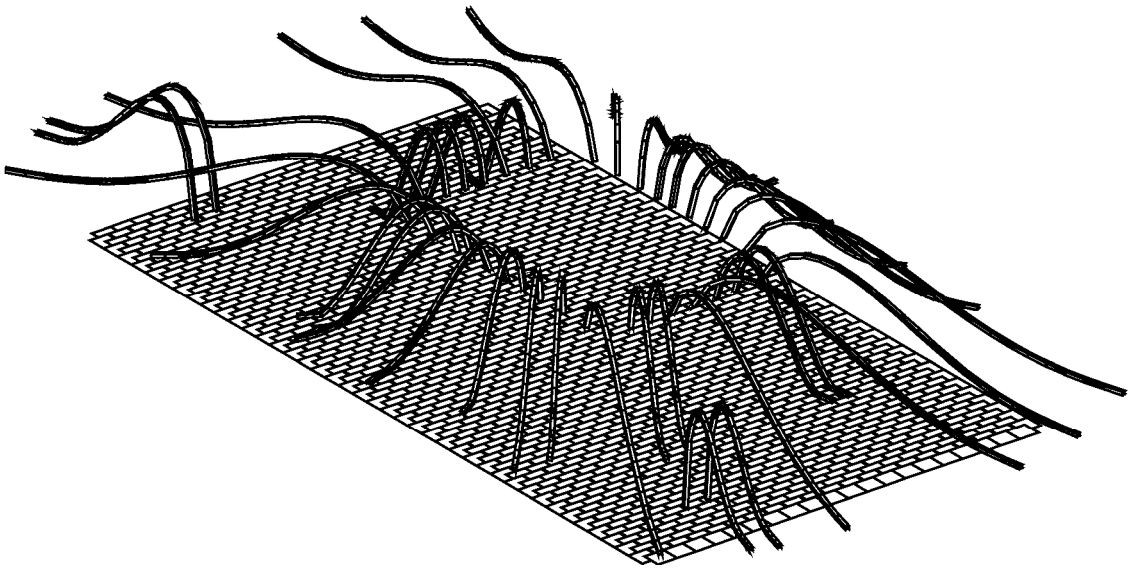


Figure 2-21: Bondwires with Ground Connections to the DA Plane

Modeling of the Die Attach Plane

As mentioned in Section 2.2.2, the DA plane was assumed ideal and all ground connections to the plane were shorted together in the Spice model to this point. However, in the partial inductance approach, it is necessary to model the full path of current flow. Note that even if the plane were perfectly conducting, it would still have inductive effects.

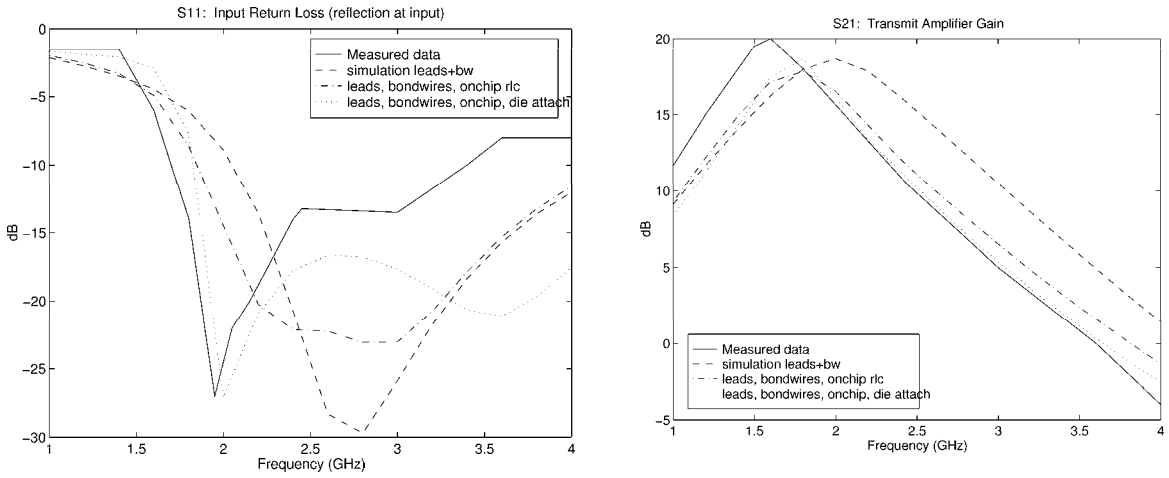
For this reason, a more accurate model for the bondwire current paths and ground connections was implemented by treating the DA as a distributed RL plane. The discretization is shown in Figure 2-21. This model replaced the previous bondwire model used in Section 2.2.2.

Simulation with the new bondwire model including the DA showed excellent results as seen in Figure 2-22. The gain matches very closely at 2.4 GHz and the minimum in the return loss occurs at nearly the same frequency as in measurement.

Better models

The above investigation shows the need for three-dimensional modeling in an efficient manner, however there is clearly room for improvement. The return loss, S_{11}, does not match well above 2 GHz and the gain, S_{21}, is off below 2 GHz. Some of the discrepancies might result from the fact that the package leads, bondwires plus die attach, and on-chip interconnect models were all built separately. Thus, the coupling between a package lead and a bondwire was not captured.

Additionally, the DA plane was modeled as a distributed RL plane only, with the



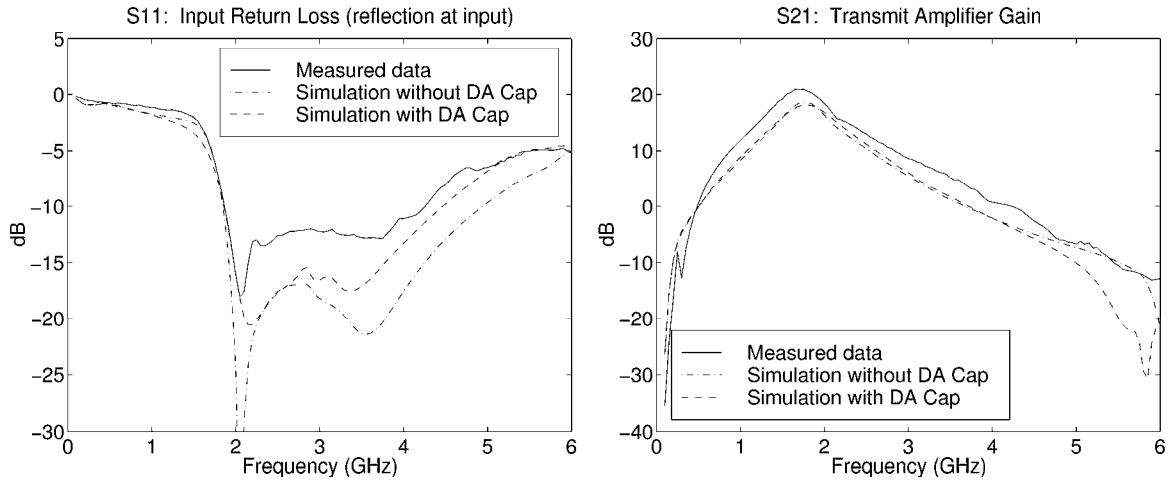
(a) (b)

Figure 2-22: Simulation with Package, Bondwire, On-chip Interconnect, and Die Attach Models

net capacitance to the printed circuit board's plane below modeled as a single capacitor. Such a lumped model is acceptable near 2.45GHz, but may be responsible for the deviation at the higher frequencies. To show this, for the same chip under different conditions, measured data was taken and the simulations were done with and without the DA capacitor. The results are shown in Figure 2-23 where it can be seen that the capacitance has a significant on the return loss, and actually improves the gain above 4 GHz.

Such strong results related to this lumped capacitor imply that an accurate distributed model of resistance, inductance, and capacitance simultaneously is necessary for further accuracy, especially at higher frequencies.

With the general need for quasistatic extraction tools motivated, the next chapter begins by describing fast methods of inductance extraction.



(a) (b)

Figure 2-23: Effect of neglecting the DA capacitance to the printed circuit board underneath

Part I

Effective Inductance Calculation and Model Order Reduction

As was seen in Section 2.1, accurate computation of the inductance and resistance alone is important for problems in which the frequencies of interest are well below the first resonance of the structure. Various workers have developed methods for inductance calculation [7, 9, 55]. The integral equation approach of Ruehli [55] has perhaps been the most popular for its wide range of applicability and its straightforward circuit interpretation. For complex three-dimensional geometries however, the approach of Ruehli leads to systems too large to solve by direct matrix solution techniques. Previous work of this author [35] extended the applicability of Ruehli's method to more complex geometries by applying an accelerated iterative solver to reduce both the time and memory required to solve the dense matrix problem at individual frequency points.

While the methods of [35] allow the analysis of geometries with tens of thousands of elements, there is still a need for more. As geometries become more complex, or higher accuracy is needed, the designer needs faster solvers which can analyze larger problems. This part of the thesis (Chapters 3, 4, and 5), addresses a number of issues for improving the process of inductance extraction developed in [35]. Many of this issues are also applicable to the EMQS domain of Chapter 6.

In Chapter 3 we review the methods of [55, 35] for inductance extraction at individual frequency points. A discretization of an integral form of Maxwell's equations is used to generate a dense linear system whose solution gives the desired inductance. For complicated interconnect structures, the linear system which must be solved is too large for direct methods such as Gaussian Elimination. For this reason we turn to Krylov-subspace iterative methods. The advantage of an iterative solve for dense matrix problems hinges on a small number of iterations to convergence. Fast convergence can be accomplished through *preconditioning*. Combining a fast iterative solver with fast potential solvers is then shown to give an $O(n)$ algorithm for inductance calculation which is orders of magnitude faster than direct methods.

In Chapter 4, we improve upon the methods in Chapter 3. From the implementation in [35], geometries with large planar structures require a uniform discretization which can generate systems with an excessively large number of elements. A method of nonuniformly discretizing planar structures becomes necessary and Section 4.1 describes an approach which allows for hierarchical descriptions and modification while still enforcing current conservation.

Also, different preconditioning techniques can have different performance for different classes of problems and improvement is always of interest. In Section 4.2 we explore an alternate preconditioning technique based on the work of [38].

The inductance approach described above is tuned to the computation of inductances

and resistances at one frequency only. But as was demonstrated in Chapter 2, the computation of a single lumped R and L as in Figures 2-4, 2-3, 2-5 or similar such circuits, is not adequate for capturing the frequency response over the entire frequency range of interest. It thus becomes important to generate a model to represent the inductance which is still of low order, but captures the frequency dependent effects. In Chapter 5, we look at applying a moment-matching variant of an eigenvalue computation technique known as Arnoldi's method to generate low order models for the inductance problem. These methods will also prove very important for model generation of the full electromagnetoquasistatic (EMQS) problem in Chapter 6.

3

Background for the Inductance Problem

In this chapter we review the methods of [55, 35] for inductance extraction at individual frequency points. The development can be divided into two parts: the first involves deriving an appropriate linear system representing the geometry and the second involves rapidly solving that linear system to compute the appropriate inductance. The layout of this chapter is shown graphically in Figure 3-1. To derive a linear system, a discretization of an integral form of Maxwell's equations is used to give a circuit interpretation. This circuit must then be "solved" at a specific frequency to compute the inductance. To efficiently solve the circuit we use the circuit technique known as mesh analysis.

For complicated interconnect structures, the linear system which must be solved is too large for direct methods such as Gaussian Elimination. For this reason we turn to Krylov-subspace iterative methods. The advantage of an iterative solve for dense matrix problems hinges on a small number of iterations to convergence. Fast convergence can be accomplished through *preconditioning* as will be discussed in Section 3.3. Combining a fast iterative solver with fast potential solvers is then shown to give an $O(n)$ algorithm for inductance calculation which is orders of magnitude faster than direct methods.

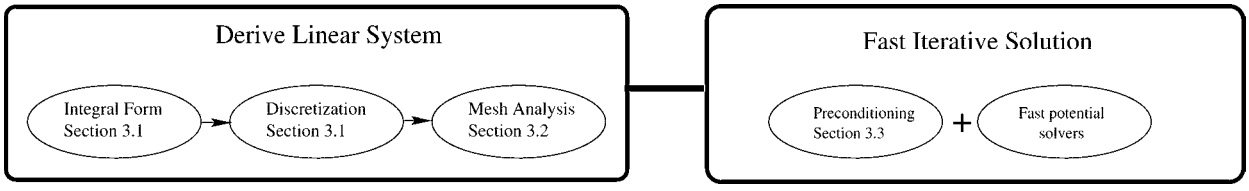


Figure 3-1: Outline of Chapter

3.1 Integral Equation

Inductance extraction is the process of computing the complex frequency-dependent admittance matrix of (1.1) under the magnetoquasistatic approximation [25].

Several integral equation-based approaches have been used to derive the Y_t associated with a given package or interconnect structure [54, 7, 66, 40] under the magnetoquasistatic approximation. These integral formulations are derived by assuming sinusoidal steady-state, and then applying the magnetoquasistatic assumption that the displacement current, $j\epsilon\omega\boldsymbol{E}$, is negligible everywhere. Given this, begin with Maxwell's equations in sinusoidal steady state,

$$\nabla \times \boldsymbol{E} = -jw\mu\boldsymbol{H} \tag{3.1}$$

$$\nabla \times \boldsymbol{H} = jw\epsilon\boldsymbol{E} + \boldsymbol{J} \tag{3.2}$$

$$\nabla \cdot (\epsilon\boldsymbol{E}) = \rho \tag{3.3}$$

$$\nabla \cdot (\mu\boldsymbol{H}) = 0 \tag{3.4}$$

where ω is the angular frequency. In addition, within the conductors, by Ohm's law,

$$\boldsymbol{J} = \sigma\boldsymbol{E} \tag{3.5}$$

where σ is the conductivity. As stated above, the frequencies of interest will be considered small enough such that the displacement current, $j\omega\epsilon\boldsymbol{E}$, can be neglected in (3.2). This assumption is clearly justified within the conductors where the conductivity is large. Under this quasistatic assumption, the divergence of (3.2) gives current conservation,

$$\nabla \cdot \boldsymbol{J} = 0. \tag{3.6}$$

Note that for this work, we wish to allow for ideal current sources, and thus (3.6) becomes

$$\nabla \cdot \boldsymbol{J} = I_s(\boldsymbol{r}). \tag{3.7}$$

From this point, we wish to eliminate the field quantities, $\boldsymbol{E}$ and $\boldsymbol{H}$, in favor of the current density, $\boldsymbol{J}$, and applied voltages. From Gauss' Law of magnetic flux, (3.4), the magnetic flux can be written as

$$\mu\boldsymbol{H} = \nabla \times \boldsymbol{A} \tag{3.8}$$

where $\boldsymbol{A}$ is the vector potential. Applying this to (3.1),

$$\nabla \times (\boldsymbol{E} + jw\boldsymbol{A}) = 0. \tag{3.9}$$

This implies that there exists a scalar function, Φ, such that

$$-\nabla\Phi = \boldsymbol{E} + jw\boldsymbol{A} \tag{3.10}$$

where Φ will be called the scalar potential. We require one final relation to relate the vector potential, $\boldsymbol{A}$ to the current density, $\boldsymbol{J}$. By use of (3.8) and by choosing the Coulomb gauge,

$$\nabla \cdot \boldsymbol{A} = 0. \tag{3.11}$$

Equation (3.2) then becomes

$$-\nabla^2\boldsymbol{A} = \mu\boldsymbol{J} \tag{3.12}$$

and thus

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu}{4\pi}\int_{V'}\frac{\boldsymbol{J}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}dv' \tag{3.13}$$

where V' is the volume of all conductors.

Substituting (3.13) and (3.5), into (3.10) yields the following integral equation:

$$\frac{\boldsymbol{J}(\boldsymbol{r})}{\sigma} + \frac{jw\mu}{4\pi}\int_{V'}\frac{\boldsymbol{J}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}dv' = -\nabla\Phi(\boldsymbol{r}). \tag{3.14}$$

Using equation (3.14) plus current conservation, (3.7), the conductor current density, $\boldsymbol{J}$, and scalar potential, Φ, can be computed [7].

3.1.1 Discretization

Given the magnetoquasistatic assumption, the current within a long thin conductor can be assumed to flow parallel to its surface, as there is no charge accumulation on the surface. Thus, for long thin structures such as pins of a package or connector, the conductor can be divided into *filaments* of rectangular cross-section inside which the current is assumed to flow along the length of the filament. In order to properly capture skin and proximity effects in these long, thin conductors, the cross section of the conductor can be divided into a bundle of parallel filaments as shown in Fig. 3-2. It is also possible to use the filament approach for planar structures, such as ground planes, where the current distribution is two-dimensional. In such cases, a grid of filaments can be used, as in Fig. 3-3. Once the conductors are discretized into filaments, the interconnection of the current filaments can be represented with a planar graph, where the n nodes in the graph are associated with connection points between filaments, and the b branches in the graph represent the filaments into which each conductor segment is discretized. An example graph, or circuit, for a single conductor example is shown in Fig. 3-4.

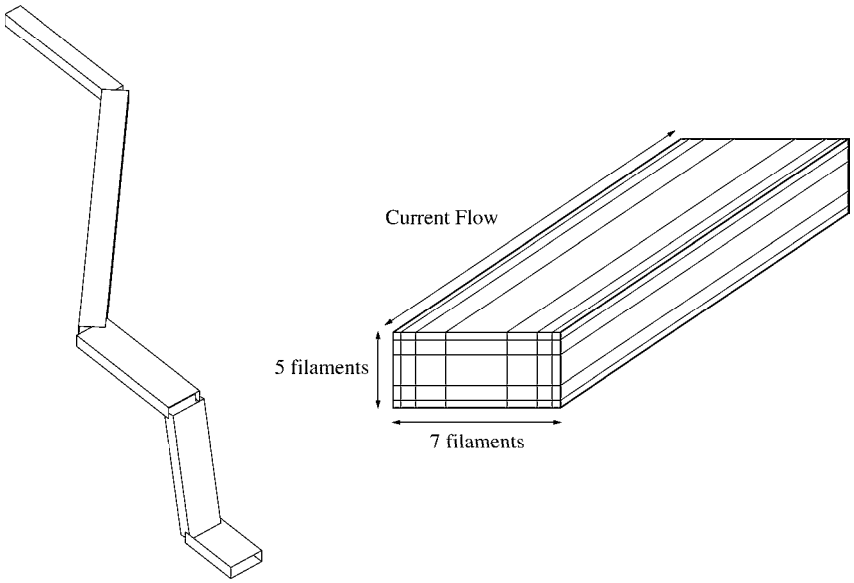


Figure 3-2: Single pin of a pin-connect divided into 5 sections, each of which is a bundle of 35 filaments.

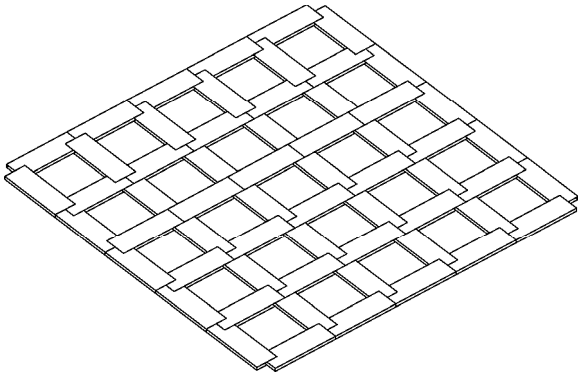


Figure 3-3: Discretization of a Ground Plane. Segments are drawn one-third actual width.

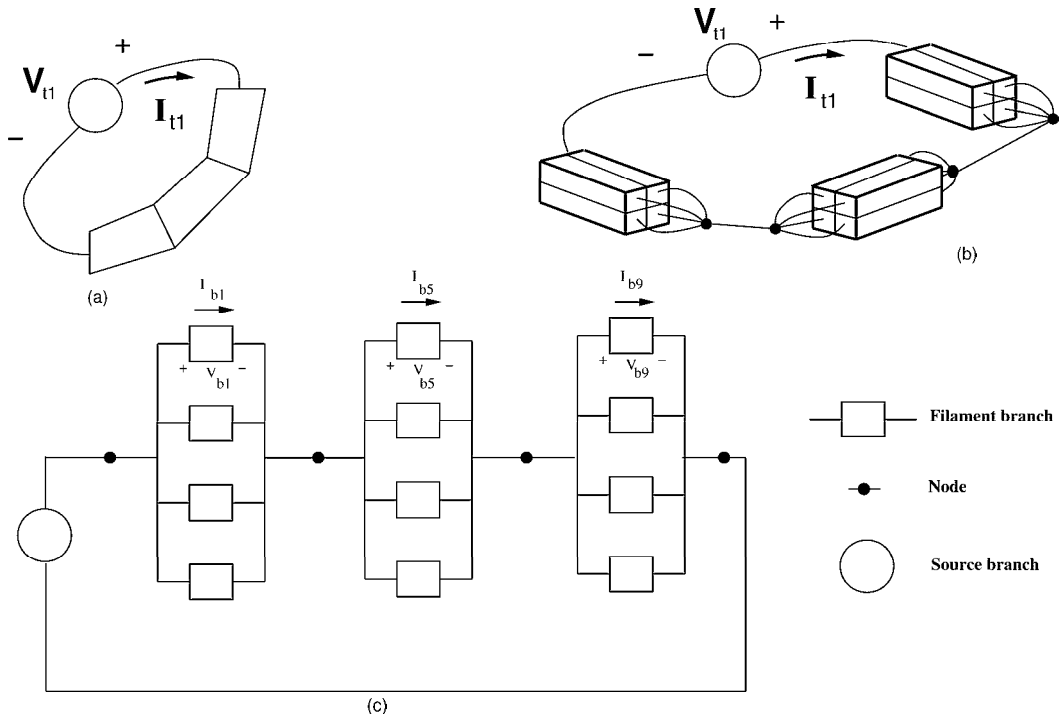


Figure 3-4: One conductor, (a) as piecewise-straight sections, (b) discretized into filaments, (c) modeled as a circuit with each box element representing a filament as a resistor in series with an inductor. Note that each filament has mutual inductance to every other inductor.

If the current density inside each filament is assumed to be constant, then the approximation to the unknown current distribution can then be written as

$$\boldsymbol{J}(\boldsymbol{r}) \approx \sum_{i=1}^{b} I_i w_i(\boldsymbol{r}) \mathbf{l_i} \tag{3.15}$$

where I_i is the current inside filament i, $\mathbf{l_i}$ is a unit vector along the length of the filament and $w_i(\boldsymbol{r})$ is the weighting function which has a value of zero outside filament i, and $1/a_i$ inside, where a_i is the cross sectional area. By defining the inner product of two vector functions, $\mathbf{a}$ and $\mathbf{b}$, by

$$(\mathbf{a}, \mathbf{b}) = \int_V \mathbf{a} \cdot \mathbf{b}\, dv \tag{3.16}$$

and following the method of moments [24], a system of b equations can be generated by taking the inner product of each of the weighting functions with the vector integral equation, (3.14). This gives

$$\left(\frac{l_i}{\sigma a_i}\right) I_i + j\omega \sum_{j=1}^{b} \left(\frac{\mu}{4\pi a_i a_j} \int_{V_i} \int_{V_j'} \frac{\mathbf{l_i} \cdot \mathbf{l_j}}{|\boldsymbol{r} - \boldsymbol{r}'|} dV'\, dV \right) I_j = \frac{1}{a_i} \int_{a_i} (\Phi_A - \Phi_B)\, dA \tag{3.17}$$

where l_i is the length of filament i, a_i is the cross section, Φ_A and Φ_B are the potentials on the filament end faces, and V_i and V_j' are the volumes of filaments i and j, respectively. Note that the right hand side of (3.17) results from integrating $\nabla\Phi$ along the length of the filament, and is effectively the average potential on face A minus the average on face B.

In matrix form, (3.17) becomes

$$(R + j\omega L) I_b = \tilde{\Phi}_A - \tilde{\Phi}_B \tag{3.18}$$

where $I_b \in \mathbb{C}$ is the vector of b filament currents,

$$R_{ii} = \frac{l_i}{\sigma a_i} \tag{3.19}$$

is the $b \times b$ diagonal matrix of filament DC resistances,

$$L_{ij} = \frac{\mu}{4\pi a_i a_j} \int_{V_i} \int_{V_j'} \frac{\mathbf{l_i} \cdot \mathbf{l_j}}{|\boldsymbol{r} - \boldsymbol{r}'|} dV'\, dV \tag{3.20}$$

is the $b \times b$ dense, symmetric positive definite matrix of partial inductances, and $\tilde{\Phi}_A$ and $\tilde{\Phi}_B$ are the averages of the potentials over the cross sections of the filament faces. Equation (3.18) can also be written as

$$ZI_b = V_b \tag{3.21}$$

where $Z = R + j\omega L \in \mathbb{C}^{\times}$ is called the branch impedance matrix and $V_b = \tilde{\Phi}_A - \tilde{\Phi}_B$ is the vector of branch voltages.

Note that one can view, for instance, filament i as a resistor with resistance R_{ii} in series with an inductor with self-inductance L_{ii} and mutual inductance L_{ij} with filament j. The circuit obtained from the graph described above is known as the Partial Element Equivalent Circuit (PEEC) [70, 54].

It is worth noting that in the example discretization shown in Fig. 3-4b and c, the ends of adjacent bundles of filaments are effectively shorted together at each node. This approximation is acceptable when the conductor is long and thin and thus the transverse current is negligible. When the conductor is not long and thin, it may be more appropriate to join the bundles together with a small grid of short filaments or even by a full 3-D discretization of filaments in the same manner that the ground plane in Fig. 3-3 is a 2-D discretization.

Also, the dense L matrix is referred to as the partial inductance matrix since its definition allow each filament to be modeled as a resistor in series with an inductor. However, since a filament is straight, it does not agree with the definition of inductance as a closed loop quantity. The filament inductances represent only part of a loop and are thus termed the "partial" inductance.

A full description of the connection between the partial inductance concept and the definition of loop inductance is given in [42]. To describe the idea with a simple example, consider four filaments connected in a loop as shown in Figure 3-5. If the two ends of the current source are close enough that the magnetic induction in the gap is negligible then from (3.1), the voltage across the source is given by [25]

$$v = \frac{d\lambda}{dt}$$

where λ is the flux linkage, or the magnetic flux linked by the filament loop,

$$\lambda = \int_S \mu \boldsymbol{H} \cdot d\boldsymbol{a} \tag{3.22}$$

where S is an open surface with edge defined by the closed contour C defined by the loop of filaments. The inductance is defined as $\ell = \frac{\lambda}{i}$ where i is the current through the loop.

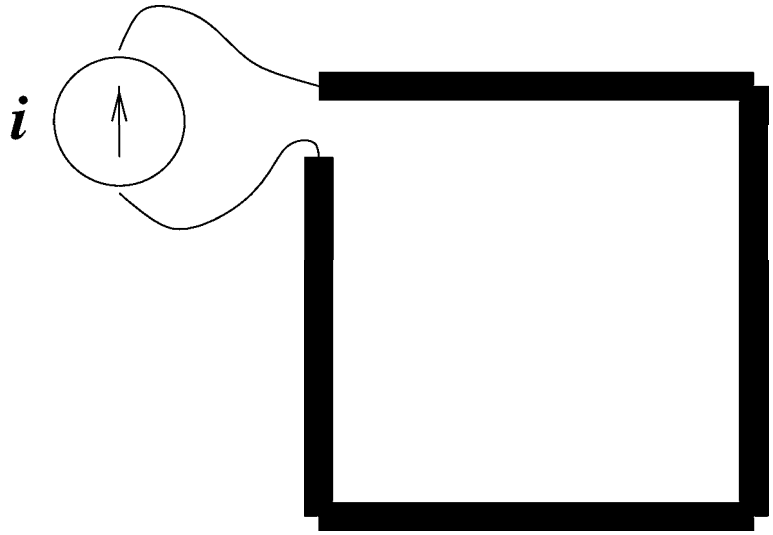


Figure 3-5: A four filament loop

This gives the familiar expression from circuit theory,

$$v = \ell \frac{di}{dt}. \tag{3.23}$$

To compute this inductance on a filament by filament basis, we can write (3.22) using (3.8) and Stokes' theorem as,

$$\lambda = \oint_C \boldsymbol{A}(\boldsymbol{r}) \cdot d\boldsymbol{r}. \tag{3.24}$$

Since the integral is now over the loop of filaments, define $\lambda_j, j = 1, \ldots, 4$ to be the integral over each filament individually,

$$\lambda = \sum_{j=1}^{4} \int_{filj} \boldsymbol{A}(\boldsymbol{r}) \cdot d\boldsymbol{r} \equiv \sum_{j=1}^{4} \lambda_j \tag{3.25}$$

where $\int_{filj}$ is the line integral along a filament. For illustration, assume the filaments are infinitely thin, then from (3.13),

$$\boldsymbol{A}(\boldsymbol{r}) = i \left(\frac{\mu}{4\pi} \sum_{j=1}^{4} \int_{fil_j} \frac{\boldsymbol{l}_j}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}' \right) \tag{3.26}$$

which gives

$$\frac{\lambda_j}{i} = \frac{\mu}{4\pi} \int_{filj} \left(\int_{fil1} \frac{\boldsymbol{l}_1}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}' + \int_{fil2} \frac{\boldsymbol{l}_2}{|\boldsymbol{r} - \boldsymbol{r}'|} d\boldsymbol{r}' + \cdots \right) \cdot d\boldsymbol{r} \tag{3.27}$$

$$= \sum_{k=1}^{4} L_{jk} \tag{3.28}$$

where L_{jk} are now defined as in (3.20) and the total loop inductance is thus

$$\ell = \sum_{j=1}^{4} \sum_{k=1}^{4} L_{jk}. \tag{3.29}$$

We think of L_{11} as the partial self-inductance of filament 1 and L_{1j} as the partial mutual inductance to the other filaments. For this simple one conductor case where skin effect is not modeled, computing Y_t as in (1.1) is a matter of summing the partial inductances to compute the total inductance in (3.29) and then $Y_t = 1/(r + j\omega\ell)$ where r is the sum of the resistances of the four filaments. For a general topology, however, we do not have such a simple expression and instead we must solve the circuit to compute Y_t.

3.1.2 Nodal Analysis Formulation

To solve the Partial Element Equivalent Circuit, current conservation, (3.7), must be enforced at each of n nodes where filaments connect. This can be written as

$$AI_b = I_s \tag{3.30}$$

where $A \in \mathbb{R}^{n \times b}$ is the branch incidence matrix and I_s is the mostly zero vector of source currents at the node locations. Each row in A corresponds to a filament connection node, and each column to a filament current. Column i in A has two nonzero entries: -1 in the row corresponding to the node from which filament i's current leaves, and $+1$ in the row corresponding to the node into which filament i's current enters.

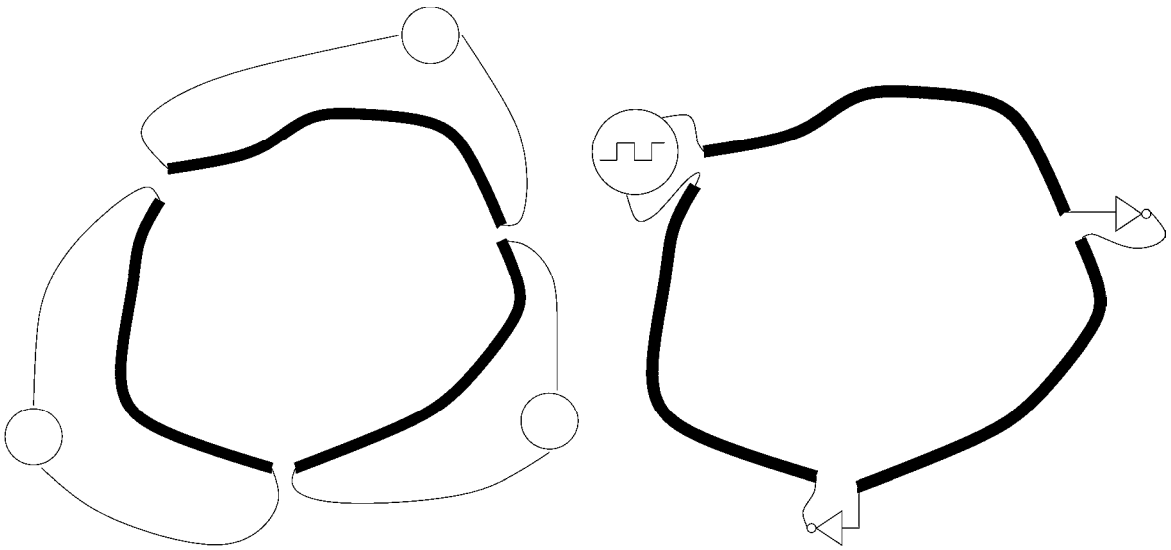
Since $\nabla^2\Phi = 0$, the branch voltages, V_b, can be derived from a set of node voltages, denoted $\tilde{\Phi}_n$, as in

$$A^t\tilde{\Phi}_n = V_b. \tag{3.31}$$

Combining (3.21), (3.30), and (3.31) yields

$$AZ^{-1}A^t\tilde{\Phi}_n = I_s. \tag{3.32}$$

Notice that column i of $Z_t = Y_t^{-1}$ can now be computed by appropriately setting the source currents, I_s, that correspond to I_{t_i} of Figure 3-4 equal to one (unit current through conductor i), and then solving (3.32) to compute the node voltages, $\tilde{\Phi}_n$. The difference



(a) (b)

Figure 3-6: (a) A single loop modeled as a 3-port so that devices in (b) can be connected in circuit simulation

of appropriate node voltages gives the entries of V_t, the vector of voltages across each of the conductors. Note that just as the magnetic interaction of individual filaments could be expressed as a partial inductance, the conductors such as in Figure 1-1 which correspond to the ports for Y_t can be individual sections of conductor and not necessarily entire loops. However, for valid simulation results, all the conductors which make up a current path must be included as ports of Y_t. In this manner, when Y_t is included in a circuit simulator, the summing as in (3.29) is performed in effect by the simulator. The advantage of such an approach is that circuit devices can be placed between the sections which make up a loop as illustrated in Figure 3-6.

In most programs, in order to compute Y_t, the dense matrix problem in (3.32) is solved with some form of Gaussian elimination or direct factorization. These programs avoid forming Z^{-1} explicitly by reformulating (3.32) into the sparse tableau form [23],

$$\begin{bmatrix} Z & -A^t \\ A & 0 \end{bmatrix} \begin{bmatrix} I_b \\ \tilde{\Phi}_n \end{bmatrix} = \begin{bmatrix} 0 \\ I_s \end{bmatrix}. \tag{3.33}$$

Using direct factorization to solve (3.33) implies that the calculation grows at least as b^3, where again b is the number of current filaments into which the system of conductors is discretized. For complicated packaging structures, b can exceed ten thousand, and solving (3.33) with direct factorization will take days, even using a high performance

scientific workstation.

3.2 The Mesh-Based Approach

The obvious approach to trying to reduce the cost of solving (3.33) is to apply iterative methods. However, such methods converge slowly because (3.33) contains equations of two different types. Another approach is to reformulate the equations using mesh analysis, and then apply an iterative method.

This section first describes the reformulation using mesh analysis and then discusses the use of a Krylov-subspace iterative method known as GMRES (Generalized Minimal Residual). Finally, the eigenspectra for the systems generated in (3.33) are compared to those generated from mesh analysis as insight into the effectiveness of GMRES for solving both systems.

3.2.1 Mesh Analysis

In mesh analysis [10], a mesh is any loop of branches in the graph which does not enclose any other branches. Also, the currents flowing around any mesh in the network are the unknowns, rather than node voltages. Mesh analysis is easiest to describe if it is assumed that sources generate explicit branches in the graph representing the discretized problem. Kirchoff's voltage law, which implies that the sum of branch voltages around each mesh in the network must be zero, is represented by

$$MV_b = V_s \tag{3.34}$$

where V_b is the vector of voltages across each branch except for the source branches, V_s is the mostly zero vector of source branch voltages, and $M \in \mathbb{R}^{m \times b}$ is the mesh matrix, where $m = b' - n + 1$ is the number of meshes and b' is the number of filaments branches plus the number of source branches.

The relationship between branch currents and branch voltages given in (3.21) still holds, and the mesh currents, that is, the currents around each mesh loop, satisfy

$$M^t I_m = I_b, \tag{3.35}$$

where $I_m \in \mathbb{C}^m$ is the vector of mesh currents. Note that each of the entries in the terminal current vector, I_t from (1.1), will be identically equal to some entry in I_m. And similarly, each of the entries in the terminal voltage vector, V_t, will correspond to

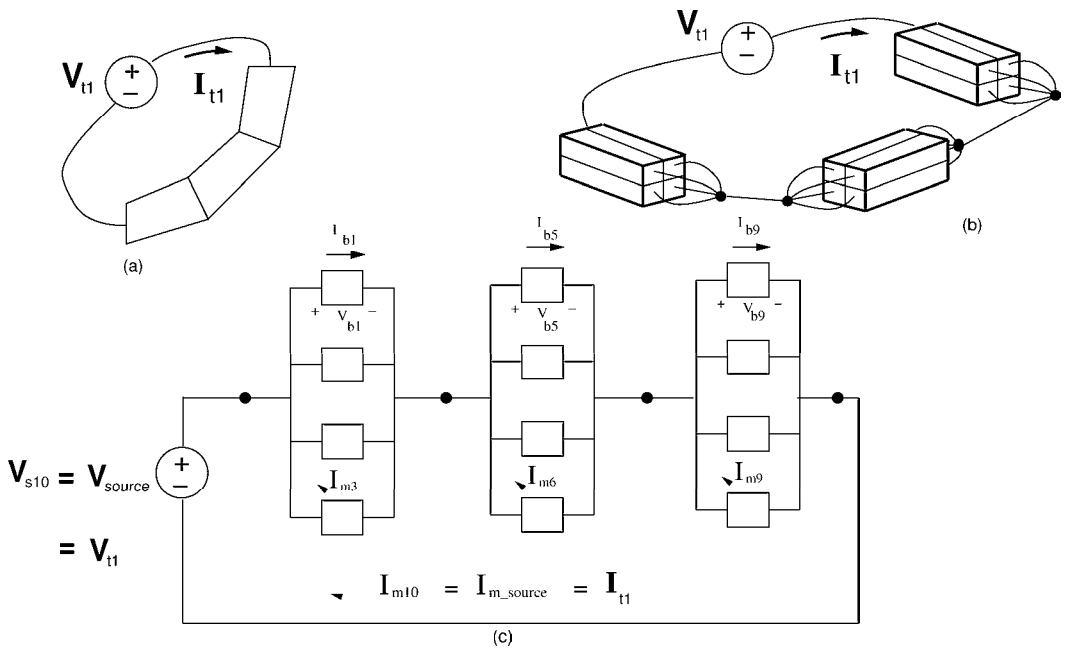


Figure 3-7: One conductor, (a) as piecewise-straight sections, (b) discretized into filaments, (c) modeled as a circuit.

some entry in V_s. Fig. 3-7 illustrates the definitions of the above quantities for a single conductor example.

Combining (3.35) with (3.34) and (3.21) yields

$$MZM^t I_m = V_s. \tag{3.36}$$

The matrix MZM^t is *easily* constructed directly [10]. To compute the i^{th} column of the reduced admittance matrix, Y_t, solve (3.36) with a V_s whose only nonzero entry corresponds to V_i, and then extract the entries of I_m associated with the source branches.

3.2.2 Using an iterative solver

The standard approach to solving the complex linear system in (3.36) is Gaussian elimination, but the cost is m^3 operations. For this reason, inductance extraction of packages requiring more than a few thousand filaments is considered computationally intractable. To improve the situation, consider using a conjugate-residual style iterative method like GMRES [58]. Such methods have the general form given in Algorithm 3.2.1.

When applying the GMRES algorithm to solving (3.36), the cost of each iteration of the GMRES algorithm is at least order m^2 operations. This follows from the fact that evaluating $\boldsymbol{r}^k$ implies computing a matrix-vector product, where in this case the matrix is MZM^t and is dense. Note also that forming MZM^t explicitly requires order m^2

Algorithm 3.2.1 (GMRES Algorithm for $Ax = b$).

```
guess x^0
for k = 0,1,... until converged {
   Compute the error, r^k = b - Ax^k
   Find x^{k+1} to minimize r^{k+1}
      based on x^i and r^i, i = 0,...,k
}
```

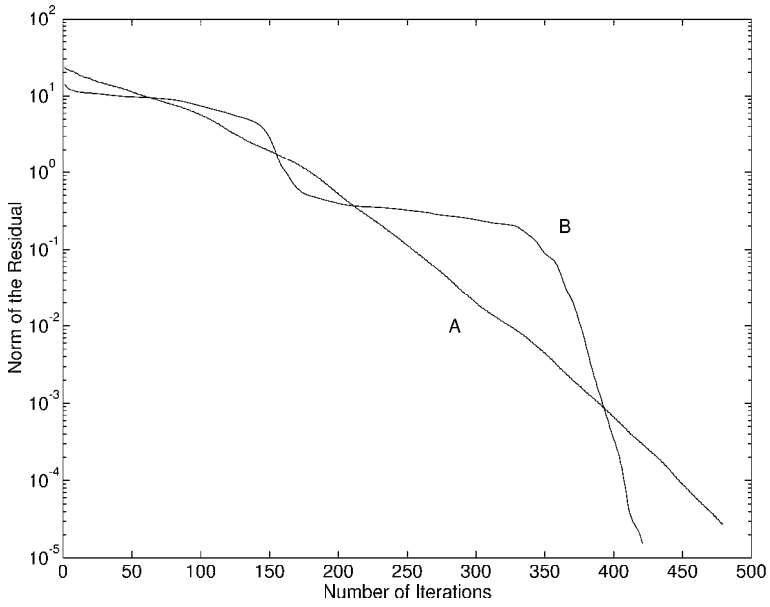


Figure 3-8: Convergence history of (A), the Sparse Tableau Formulation, and (B), the Mesh Formulation for a coarse discretization of the printed circuit board example of Figure 3-16.

storage. It is possible to approximately compute $MZM^tI_m^k$ in order b operations using a hierarchical multipole algorithm for electrostatic analysis [20]. Such algorithms also avoid explicitly forming MZM^t, and so reduce the memory required to order b. Using the multipole approach for inductance is described in [36].

Regardless of the approach used for matrix-vector products, the iterative approach will not be advantageous unless the number of iterations to convergence is small. The convergence behavior of applying GMRES to (3.36) and (3.33) for an example of dimension 751 is shown in Figure 3-8. The results indicate both that convergence is slow and that the mesh formulation does not have significant advantages over the nodal form.

However, note that the GMRES algorithm can be directly applied to solving (3.36)

because the matrix MZM^t is easily constructed explicitly. This is not the case for the nodal formulation, (3.32), as either the Z matrix must first be inverted or the sparse tableau form in (3.33) must be used. We shall see that the sparse tableau form is disadvantageous because it is a much larger system of equations and it is difficult to accelerate iterative solution as described in Section 3.3.

3.2.3 Understanding Convergence Rates

The rate of convergence of GMRES for solving $Ax = b$ can be related to the eigenspectrum of the matrix A. From this fact, the convergence properties of nodal analysis versus mesh analysis can be compared.

To begin, we note the following connections between matrices with real eigenvalues and convergence discussed in general in [65] and for inductance in [36]:

- If the eigenvalues are spread over a large interval, convergence will be slow, while clustered eigenvalues lead to faster convergence.
- Matrices with eigenspectra which have both negative and positive clusters of eigenvalues will converge slower than those with eigenvalues on only one side of the origin.
- Eigenvalues close to the origin slow convergence.

Consider now the spectra resulting from the nodal formulation, (3.33), and the mesh formulation, (3.36), for the printed circuit board example described later in Fig. 3-16. The printed circuit board is two thin metal sheets sandwiching 255 small copper lines.

For inductance extraction problems, the high frequency ($\omega \rightarrow \infty$) limit gives the worst case convergence so consider a coarse discretization of this structure yielding a 751×751 MLM^t matrix with a condition number $\kappa \approx 10^6$. The condition number gives a measure of the spread of the eigenvalues of a matrix and is defined as

$$\kappa(A) \equiv \|A\|_2 \, \|A^{-1}\|_2 = \frac{\lambda_{max}(A)}{\lambda_{min}(A)} \tag{3.37}$$

and $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are the maximum and minimum eigenvalues of A, respectively.

Clearly, MLM^t is always positive definite since L is positive definite and is normal since L is symmetric. Also, from the spectrum of MLM^t shown in Fig. 3-9, one can see that, while most of the eigenvalues are in the interval $[10^{-10}, 10^{-7}]$, the remaining isolated eigenvalues are located toward the origin in the interval $[10^{-13}, 10^{-10}]$. Such

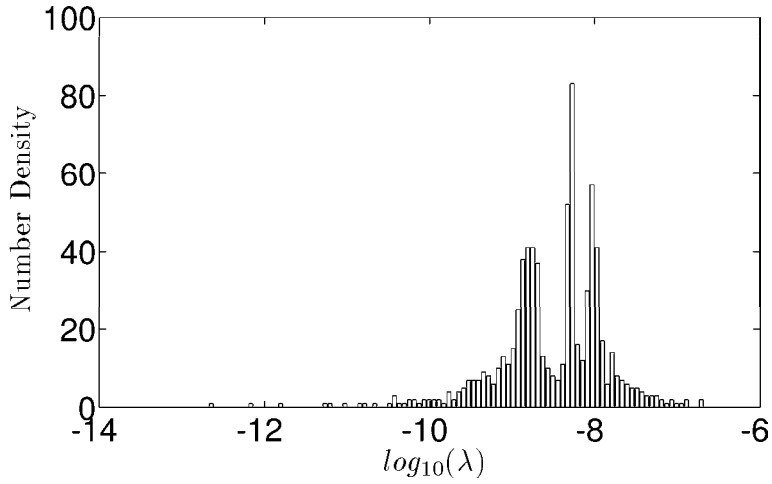


Figure 3-9: Eigenvalue spectrum of MLM^t for a coarse discretization of the printed circuit board example

small eigenvalues are not easily cancelled with the polynomial produced by the Krylov-subspace methods, so they will significantly slow convergence.

The high frequency limit for (3.33) can be written as

$$S = \begin{bmatrix} L & A^t \\ A & 0 \end{bmatrix} \tag{3.38}$$

where S is now a real symmetric matrix. The eigenspectrum, $\lambda(S)$, is shown in Fig. 3-10. While there are fewer eigenvalues close to zero, a large portion of eigenvalues are negative. Note that for S, since each entry in $L \approx 10^{-9}$, the values of L were scaled by 5×10^8 for the spectrum in Fig. 3-10 and the GMRES comparison of Fig. 3-8 described below.

Both MLM^t and S have an eigenstructure which does not yield fast convergence with GMRES as was shown in Fig. 3-8. In this case, the mesh formulation, MLM^t, has no advantage over the nodal formulation, S. Significant improvement is possible, however, through *preconditioning* which involves solving an equivalent system with a more favorable eigenstructure. Preconditioning MLM^t and difficulties in preconditioning S are discussed in Section 3.3.

3.3 Accelerating Iteration Convergence

In Section 3.2.3 it was observed that the GMRES iterative method converged slowly when applied to (3.36). In general, however, the GMRES iterative method applied to

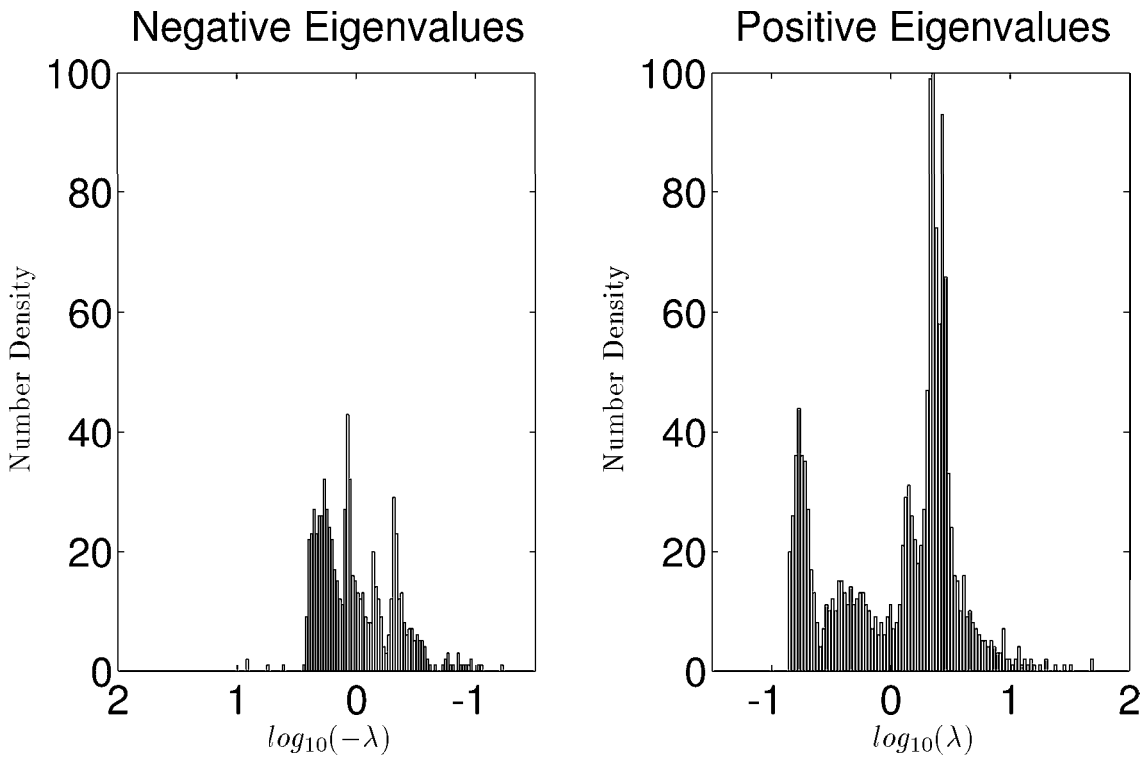


Figure 3-10: Eigenvalue spectrum of S, the sparse tableau formulation, for a coarse discretization of the printed circuit board example

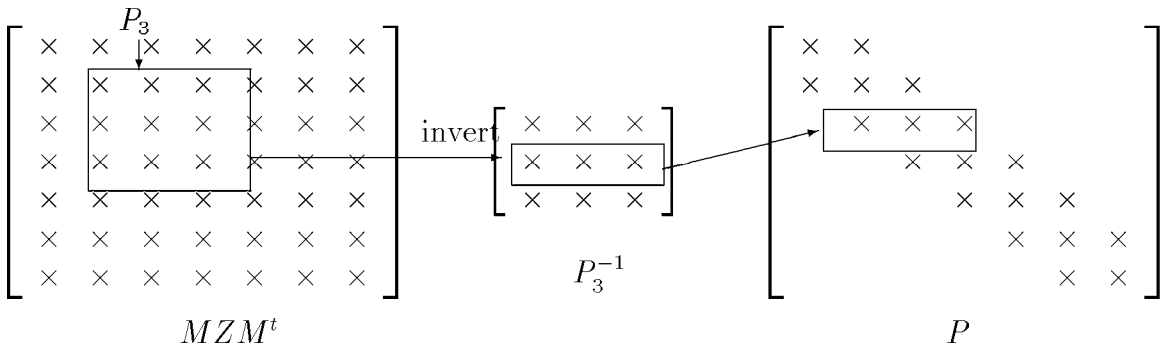


Figure 3-11: The steps leading to the third row of the preconditioner P ("$\times$" denotes a non-zero element). Note that for illustration, P_3 is drawn as a block along the diagonal.

solving (3.36) can be significantly accelerated by *preconditioning* if there is an easily computed good approximation to the inverse of MZM^t. We denote the approximation to $(MZM^t)^{-1}$ by P, in which case preconditioning the GMRES algorithm is equivalent to using GMRES to solve

$$(MZM^t)Px = V_s \tag{3.39}$$

for the unknown vector x. The mesh currents are then computed with $I_m = Px$. Clearly, if P is precisely $(MZM^t)^{-1}$, then (3.39) is trivial to solve, but then P will be very expensive to compute. This section highlights work to develop preconditioners for MZM^t.

3.3.1 Local Inversion

An easily computed good approximation to $(MZM^t)^{-1}$ can be constructed by noting that the most tightly coupled meshes are ones which are physically close. To exploit this observation, for each mesh i, the submatrix of MZM^t corresponding to all meshes near mesh i is inverted directly. Then, the row of the inverted submatrix associated with mesh i becomes the i^{th} row of P. This is illustrated in Fig. 3-11, where the submatrix is drawn as a block along the diagonal for illustration. This idea for preconditioning was shown to work well for capacitance extraction in [47] and in another context in [68]. We refer to this preconditioner as a "local-inversion" preconditioner, because it is formed by inverting physically localized problems.

For inductance problems, this preconditioner has been shown to work well for pin-connect and other similar structures for which most of the meshes are small and thus what is 'local' is obvious. The fact that most of the meshes are small can be observed from Fig. 3-7 by noticing that most of the meshes, such as those associated with I_{m3}, I_{m6}, and I_{m9}, are small and consist of only two physically close filaments. Comparatively, there are relatively few large meshes, such as I_{m10}, each which result from the presence of

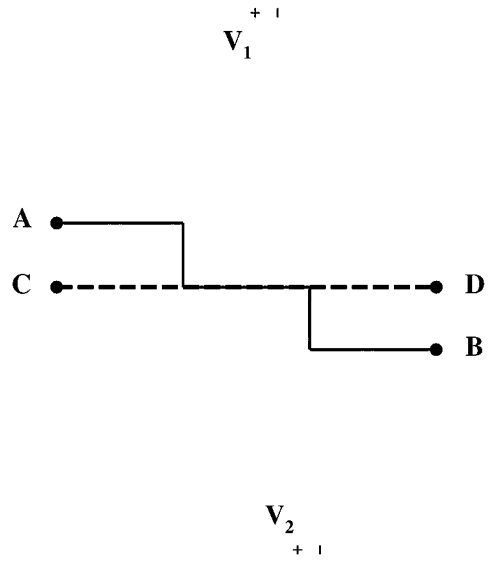


Figure 3-12: Two ground plane meshes due to external sources. One source is connected between points A and B and the other between C and D.

an external source and include many filaments. The many filaments which are included in each external source mesh span much of the physical problem domain and therefore much of the problem can be physically close to these large meshes. For this reason, the large meshes associated with sources cannot be included in the preconditioner, otherwise excessively large subproblems would be inverted directly.

The discretization of Figure 3-7 is representative of one pin of a pin-connect type structure as is shown in Figure 3-15. Since there are relatively few large meshes in a pin-connect structure and they are physically separate (only one per pin), not including the large meshes when forming the preconditioner does not significantly slow convergence.

For large ground-plane problems, with possibly hundreds of external sources, the performance of local-inversion is severely degraded. As for pin-connects, many of the meshes are small; in this case, most meshes include four filaments (See Fig. 3-3). Each external source, however, requires the formation of a large mesh traversing the ground plane between its two contact points as shown in Fig. 3-12. If there are hundreds of these meshes, in which case many of them will be physically close or even possibly partially overlapping, local-inversion becomes ineffective since it cannot include large meshes.

3.3.2 Sparsified Preconditioners

Other approaches to preconditioning which might help account for large mesh interactions involve somehow approximately factoring MZM^t or directly factoring an approx-

imation to MZM^t. The idea is then to first compute a sparse factorization

$$P^{-1} = LU \tag{3.40}$$

and then at each iteration, Px will be computed by solving $LUy = x$ with forward and backward substitution.

Approximately factoring MZM^t using an approach like incomplete LU factorization [59] is ineffective however, because the diagonals of MZM^t are not necessarily greater than the sum of the off-diagonals and therefore ignored terms can become more significant. Another approach is to "sparsify" MZM^t, possibly based on the sparsity pattern of MRM^t, and then directly factor the "sparsified" matrix to construct a preconditioner.

Fig. 3-13 shows the results of applying the local-inversion preconditioner, the sparsity-based preconditioner, plus an example of the sparsified-L class of preconditioners to be discussed in the next section. The MZM^t matrix is 751×751 and corresponds to the printed-circuit board example of Fig. 3-16 in the high frequency limit, that is, as $\omega \to \infty$. The high frequency case is chosen because it has been found to demonstrate the worst case convergence for the sparsified preconditioners, as discussed in the next section. A point worth noting here is that for these sparsified preconditioners in the low frequency limit, $\omega \to 0$, we have that P^{-1}, $MZM^t \to MRM^t$ and therefore $(MZM^t)P \to I$. The local-inversion preconditioner shows approximately the same convergence behavior at low and high frequency.

From Fig. 3-13 it is apparent that local-inversion and sparsity-based preconditioning only slightly accelerated convergence over no preconditioning. Note that this example includes approximately 300 large meshes.

3.3.3 Positive Definite Sparsifications of L

To develop a better preconditioner, instead of sparsifying MZM^t as described above, consider sparsifying the partial inductance matrix, L, and then generating the preconditioner by directly factoring the sparse result, $P^{-1} = M(R + j\omega L_s)M^t$, where L_s is the sparsified partial inductance matrix. We call this class of preconditioners "sparsified-L." Note that for the low frequency limit, the whole problem would be factored exactly since R is diagonal. This section shows that this type of preconditioner is effective for a large class of problems and that L_s must be chosen symmetric positive definite for this type of preconditioner to be effective.

To motivate the discussion of this section, consider the following two choices for L_s: the sparsest approach to choosing L_s would be to take the diagonal of L, or consider

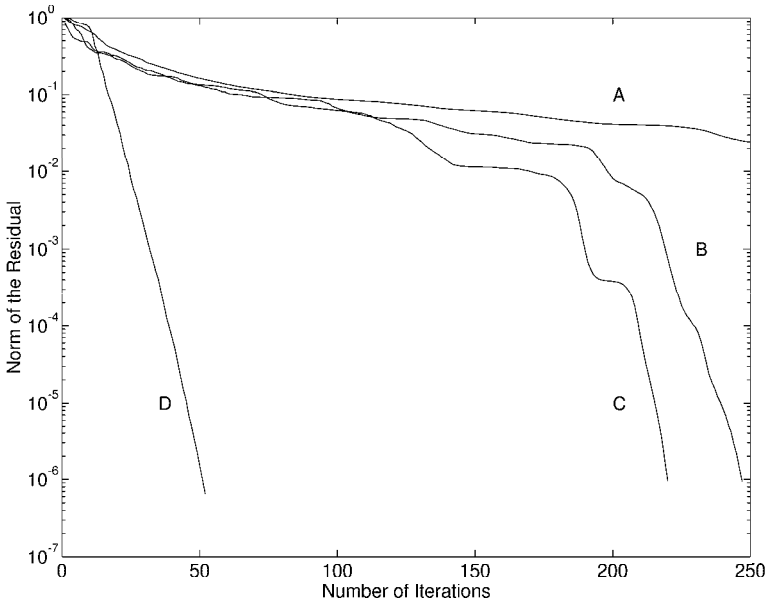


Figure 3-13: Convergence of GMRES applied to PCB example with no preconditioning (A), sparsity-based preconditioning (B), local-inversion preconditioning (C), and sparsified-L preconditioning using the diagonal of L (D)

sparsifying L based on the magnitude of the elements by zeroing all terms except those that satisfy $L_{ij}^2 > \epsilon |L_{ii} L_{jj}|$, for some ϵ. This threshold sparsification approach produces an L_s which is not necessarily positive definite. Fig. 3-14 compares the threshold sparsification to preconditioning choosing L_s as the diagonal of L. Clearly, choosing L_s to be the diagonal of L produced the better preconditioner, yet many more terms were included in L_s for the threshold sparsification.

The high frequency limit

For inductance extraction problems, the high frequency ($\omega \to \infty$) limit gives the worst case convergence. As $\omega \to \infty$, the preconditioned matrix reduces to $(MLM^t)(ML_sM^t)^{-1}$. In what follows, we present results to show that L_s should be chosen to be symmetric positive definite for this sparsified-L class of preconditioners to be effective. For proofs, see [36].

Theorem 3.3.1. If L_s is symmetric positive definite, then the preconditioned system, $(MLM^t)(ML_sM^t)^{-1}$ has positive eigenvalues.

Theorem 3.3.1 leads to the result that the condition number of preconditioned system in the high frequency limit is bounded *independent* of the mesh matrix, M.

Theorem 3.3.2. If L_s is positive definite, then

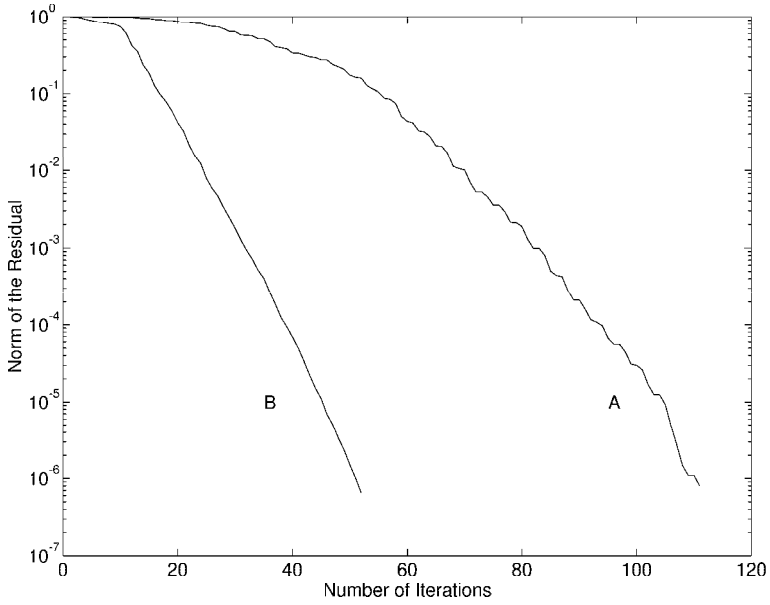


Figure 3-14: Convergence of GMRES applied to PCB example with threshold preconditioning for $\epsilon = 10^{-1}$ (A), and diagonal-of-L preconditioning (B)

$$\kappa[(MLM^t)(ML_sM^t)^{-1}] \leq \kappa(LL_s^{-1})$$

where, for a matrix with positive eigenvalues A, the condition number $\kappa(A)$ is defined as $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ and $\lambda_{\max}(A)$, $\lambda_{\min}(A)$ are the maximum and minimum eigenvalues, respectively.

The above two theorems lead to the conclusion that one should focus on choosing positive definite L_s matrices. As described above, the sparsest approach would be to take the diagonal of L. Another approach is to divide physical space into disjoint regions and then to include in L_s only the principal submatrices of L corresponding to the groups of filaments contained inside each region. Thus, a filament will be included in exactly one region and by appropriately numbering the branches, L_s can be written as a block diagonal matrix. An immediate approach is for each region, or block, to consist of the set of parallel filaments in a single section of conductor. Each block will then be no larger than the number of filaments in a section. For the simple one conductor example shown in Fig. 3-7, L_s would consist of three 4×4 blocks. Another choice is to uniformly divide space into cubes and have each block consists of filaments within a cube. This cube-block method is easy to implement since the cube information is needed for implementation of the multipole algorithm. Note also that the cube-block preconditioner is denser than the

section-block or diagonal-of-L preconditioners.

Theorem 3.3.3. L_s for the cube-block and section-block preconditioners is positive definite.

Proof. The set of eigenvalues of a block diagonal matrix is the union of the sets of eigenvalues from each block. Since L is symmetric positive definite, so are all of its principal submatrices (See, for instance [26], p. 397). Given the block diagonals of L_s are principal submatrices of L, the theorem is proved. □

Performance of Sparsified-L Preconditioners

To compare the relative merits of the diagonal-of-L, cube-block, and section-block preconditioners, consider the two industrial examples in Figs. 3-15 and 3-16. Fig. 3-15 is thirty-five pins of a 68-pin cerquad package and Fig. 3-16 is a portion of a printed circuit board (PCB) that would be placed underneath a Pin-Grid-Array package. The PCB example consists of two resistive reference planes sandwiching 255 copper lines. Each plane in the PCB has 53 external contacts not shown in the figure. For this experiment, the cerquad package was discretized into 3488 filaments which corresponds to 3305 meshes and each reference plane in the PCB was discretized into a 60×60 grid of meshes giving a total 7501 meshes including the copper lines. The GMRES error in the solution at high frequency as a function of iteration is plotted in Fig. 3-17 for the cerquad example, and in Fig. 3-18 for the PCB example. Note that the section-block and diagonal-of-L preconditioners are identical for the PCB example since there is only one filament per conductor section. As the figures clearly show, the block diagonal preconditioners are an improvement over the diagonal-of-L and local-inversion preconditioners. It is worth noting that for the cerquad package example, the number of non-zero elements in the factored cube-block preconditioner is 43 times that for the diagonal-of-L preconditioner, possibly prohibiting its use for larger problems. Also, for the pin-connect example, unlike the PCB example, local-inversion preconditioning did better than diagonal-of-L. This behavior can be expected since there are only 35 large meshes which must be excluded from the local-inversion preconditioner.

3.3.4 Difficulties Preconditioning the Nodal Formulation

Preconditioning the S matrix of (3.38) is very difficult since it contains equations of two different types: those resulting from $V_b = ZI_b$, and those from current conservation, $AI_b = I_s$. While it is possible to approximate the inverse of Z, it is difficult to account

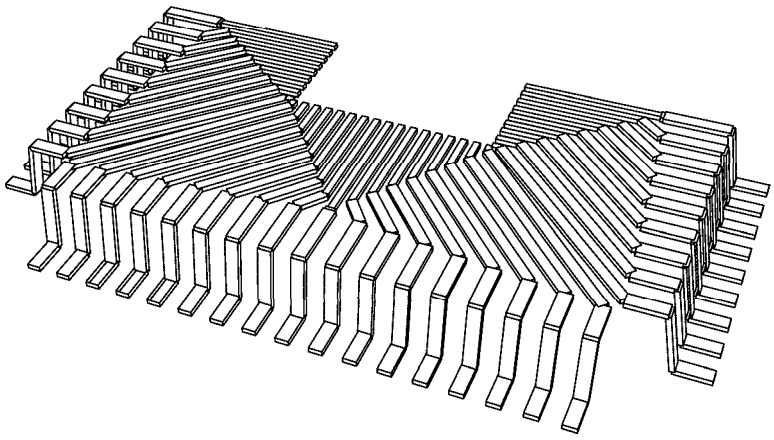


Figure 3-15: Half of a cerquad package. Thirty-five pins shown.

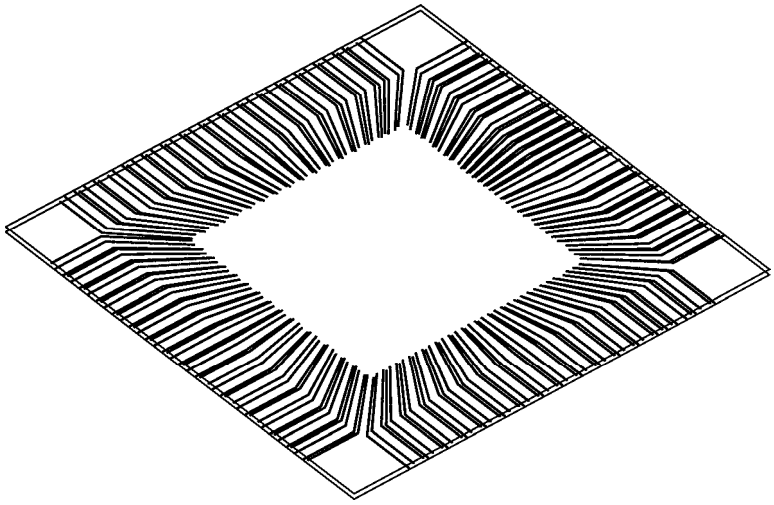


Figure 3-16: A portion of a printed circuit board directly underneath a PGA package. Two resistive reference planes sandwiching 255 copper lines. Only the outline of the planes is drawn.

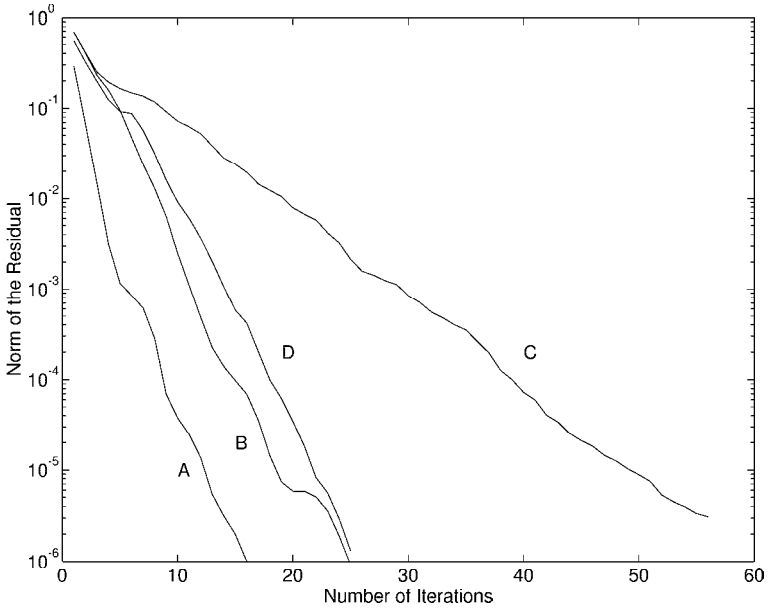


Figure 3-17: Convergence of GMRES applied to the cerquad example with cube-block preconditioning (A), section-block preconditioning (B), diagonal-of-L preconditioning (C), and local inversion preconditioning (D).

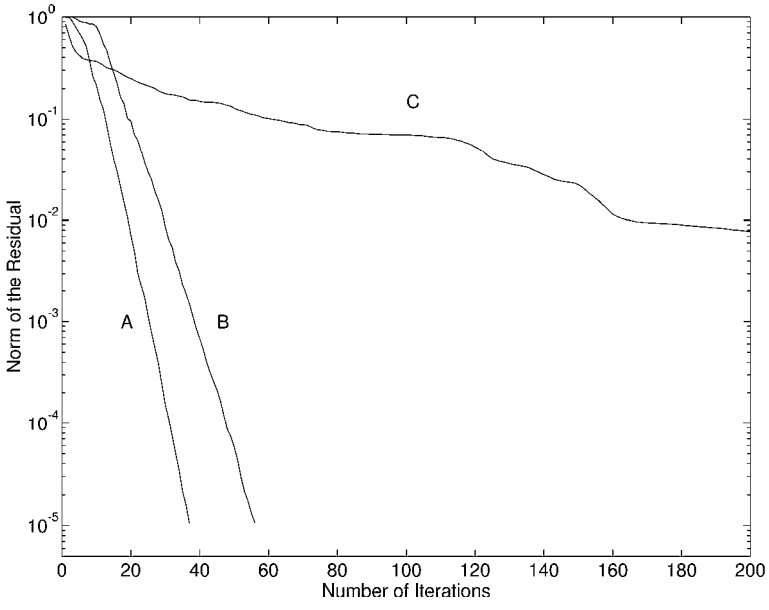


Figure 3-18: Convergence of GMRES applied to the PCB example with cube-block preconditioning (A), and diagonal-of-L preconditioning (B), and local-inversion preconditioning (C).

for the effect of the A matrix. In some sense, the A matrix and its position in (3.33) are in fact responsible for the many negative eigenvalues. As the following theorems show, for the ideal case of $L = I$, where I is the identity, S has exactly n negative eigenvalues, where n is the number of rows of A. Note that $L = I$ corresponds to the low frequency limit of a system of conductors discretized into identical filaments.

Theorem 3.3.4. Let $C = \begin{bmatrix} I & B \\ A & 0 \end{bmatrix}$ and assume AB is nonsingular and diagonalizable, where I is the bxb identity matrix, $B \in \mathbb{R}^{bxn}$, $A \in \mathbb{R}^{nxb}$ and $n < b$.

Then $b - n$ eigenvalues of C are one, and the other $2n$ eigenvalues of C satisfy $\lambda(\lambda - 1) = \lambda'$ where λ' is an eigenvalue of AB.

Proof. See [36] □

Theorem 3.3.5. The matrix S with $Z = I$ has n negative eigenvalues and b positive eigenvalues.

Proof. Since AA^t is positive definite, Thm 3.3.4 gives that the n negative eigenvalues are given by

$$\lambda = (1 - \sqrt{1 + 4\lambda'})/2. \tag{3.41}$$

where $\lambda' \in \lambda(AA^t)$. Similarly, another n are positive, and the remaining $b - n$ are equal to one. □

Even if Z^{-1} could be computed exactly, preconditioning using only Z^{-1} as in

$$P = \begin{bmatrix} L^{-1} & 0 \\ 0 & I \end{bmatrix} \tag{3.42}$$

for the high frequency limit, does not improve the performance of GMRES in part due to the fact that there are still n negative eigenvalues. This can be shown by a similar argument as in Theorem 3.3.5 with

$$SP = \begin{bmatrix} I & A^t \\ AL^{-1} & 0 \end{bmatrix}. \tag{3.43}$$

Note that $AL^{-1}A^t$ is positive definite since L is positive definite.

This preconditioned matrix, in fact, does not show much improvement in convergence rate over the unpreconditioned system as shown in Fig. 3-19.

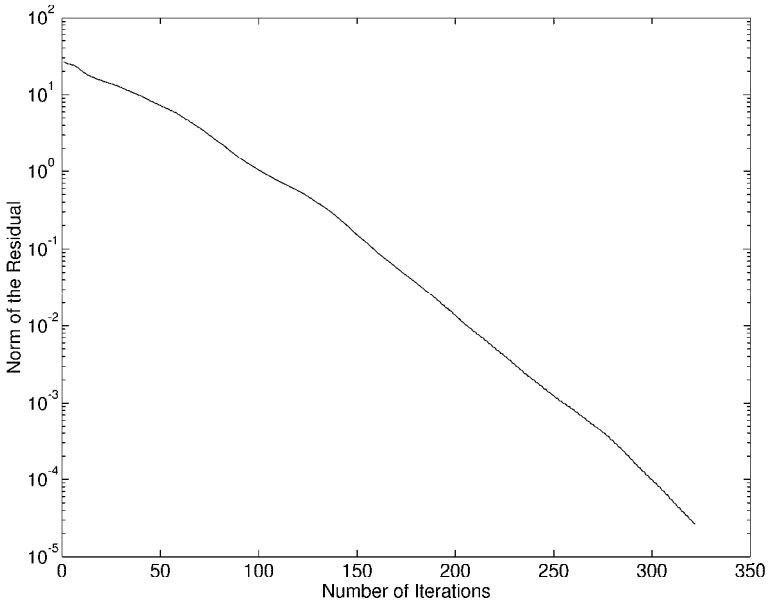


Figure 3-19: Convergence history for the Sparse Tableau Formulation using L^{-1} as a preconditioner

3.4 Algorithm Results

As an example of the utility of frequency dependent inductance extraction possible with FastHenry, consider the two cases of computing the mutual inductance between a pair of PC board traces over a resistive ground plane, as shown in Fig. 3-20, and the same pair of traces over a divided ground plane, as in Fig. 3-21. At the near end of each trace, a source is connected from each trace to the plane below. At the far end, the traces are shorted to the plane so that each trace has its return paths through the ground plane. For the divided plane case, the two portions of the plane are electrically connected with short resistive 'tethers' toward the outer edges as shown. The traces are 8 mils wide, 1 mil thick, 8 mils above the 1 mil thick ground plane, and their center to center distance is 32 mils.

If one of the sources is set to one volt and the other to zero current will flow down the excited trace and return through the plane. For the solid plane case, the current in the plane with a DC source produces a current distribution pattern which spreads to fill the width of the plane. Similarly in the divided plane case, the current spreads throughout most of the plane, but narrows as it crosses the tethers. The situation at a high frequency is quite different. For the solid plane, the ground plane return current is concentrated directly underneath the trace, but for the divided plane the current leaves

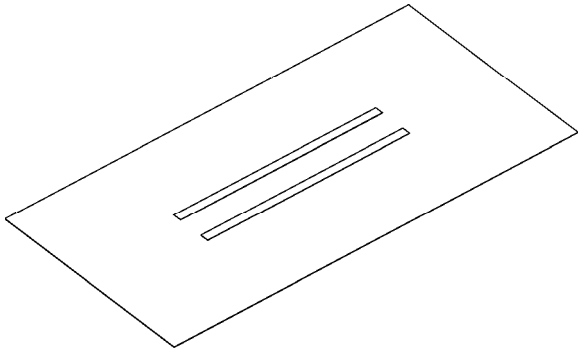


Figure 3-20: Two Traces over a Solid Ground Plane. The return path for the traces is through the plane. Traces are widened for illustration.

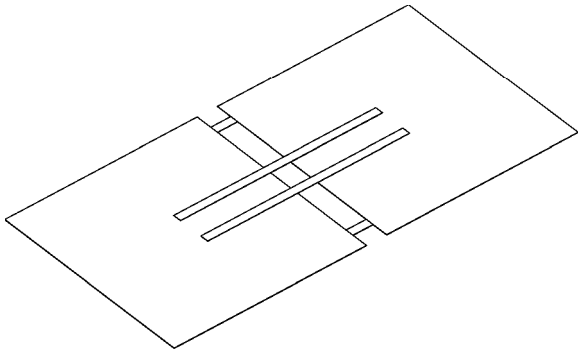


Figure 3-21: Two Traces over a Divided Ground Plane. The return path for the traces is through the plane. The divided portions are connected together toward the edges as shown. Traces are widened for illustration.

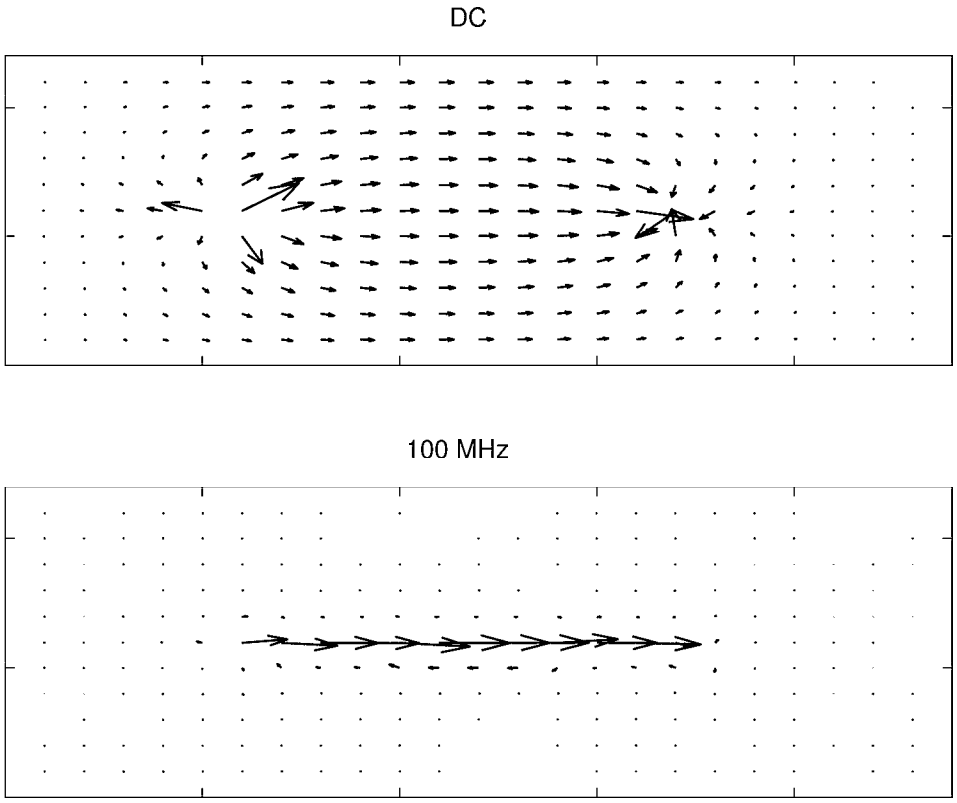


Figure 3-22: Current Distribution in Solid Ground Plane at DC and high frequency

the path underneath the trace to cross the tethers (See Figs. 3-22 and 3-23).

This difference has a marked effect on the mutual inductance between the traces as the frequency rises. For the solid plane, as the frequency rises, the current gathers underneath the trace and the mutual inductance drops by two orders of magnitude, however for the divided plane, little decrease is observed with frequency (See Fig. 3-24).

To demonstrate the computational efficiency of the multipole-accelerated code we call FastHenry, the coarse discretization of the ground plane of the example shown in Fig. 3-20 was successively refined. As the discretization of the plane is refined, the size of the problem will grow quickly, making the memory and CPU time advantage of the multipole-accelerated, preconditioned GMRES algorithm apparent (see Figs. 3-25 and 3-26). As the graphs clearly indicate, the cost of direct factorization grows like m^3, the cost of explicit GMRES grows as m^2, but the cost of multipole-accelerated GMRES grows only linearly with m. In addition, the memory requirement for multipole-accelerated GMRES algorithm grows linearly with m, but grows like m^2 for either explicit GMRES or direct factorization. In particular, for a $12,802$ mesh problem, the multipole accelerated algorithm is more than two orders of magnitude faster than direct factorization, and

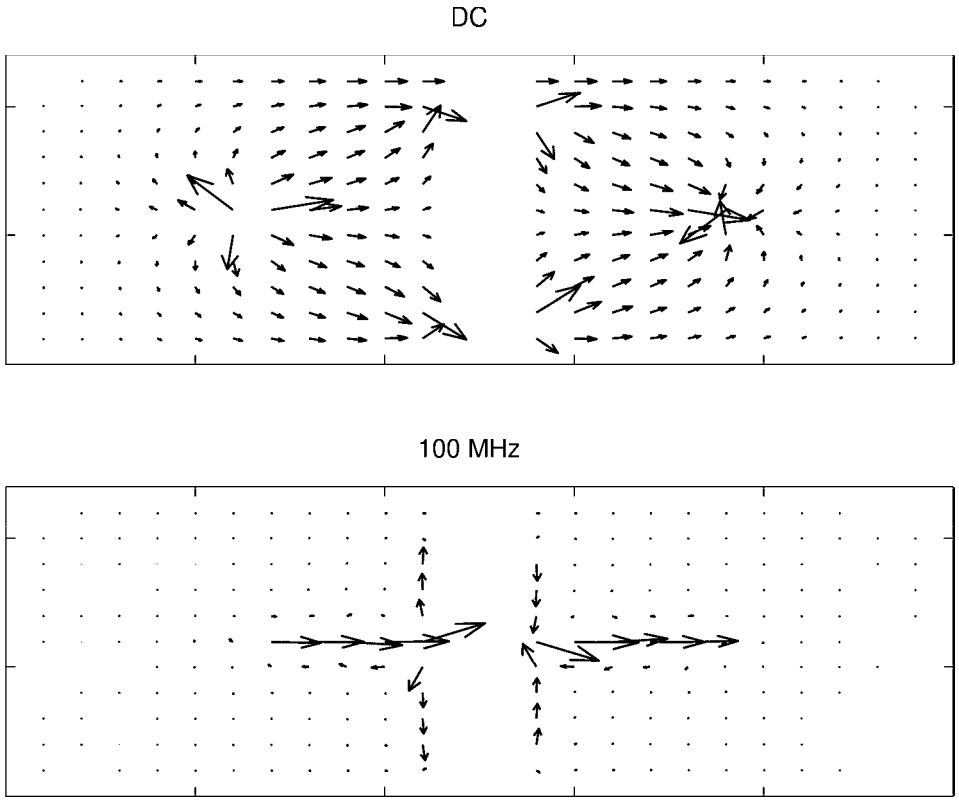


Figure 3-23: Current Distribution in Divided Ground Plane at DC and high frequency

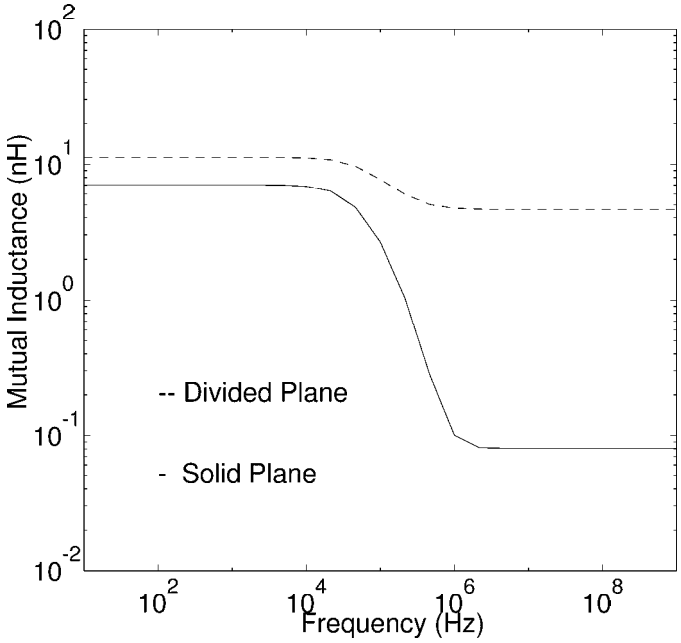


Figure 3-24: Mutual Inductance Between Traces

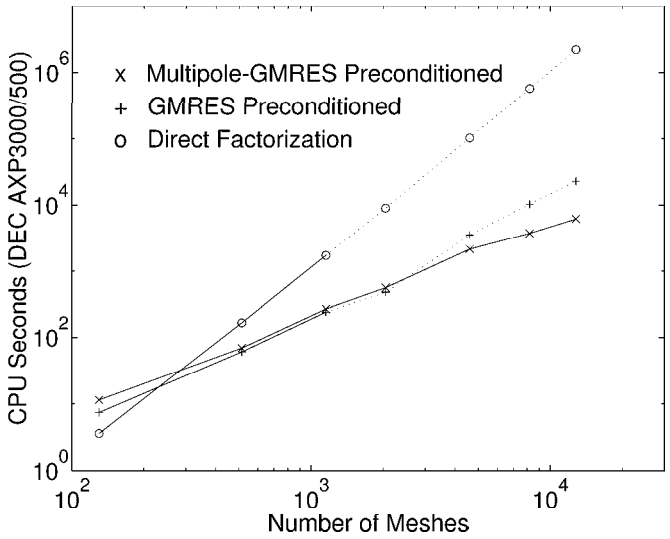


Figure 3-25: Comparison of the CPU time to compute the reduced inductance matrix for two traces over a solid plane using direct factorization, GMRES, and GMRES with with multipole acceleration.

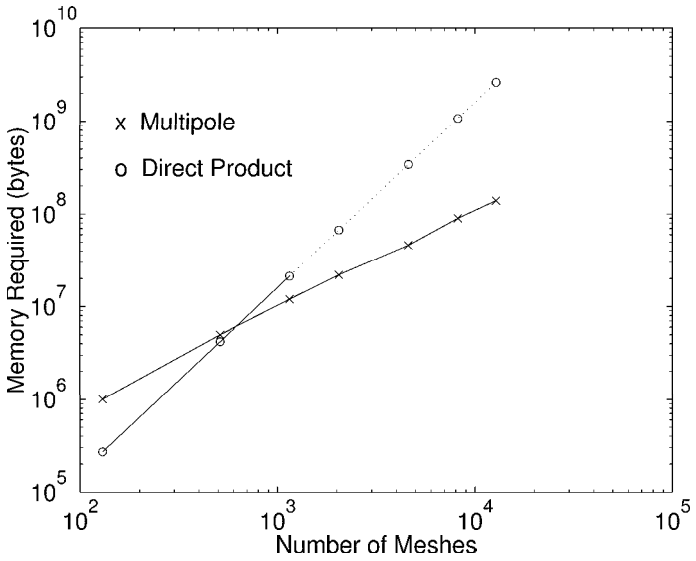


Figure 3-26: Comparison of the memory required using explicit matrix-vector products and using the multipole algorithm.

uses an order of magnitude less time and memory than explicit GMRES. Note that the dotted lines in Figs. 3-25 and 3-26 indicate extrapolated values due to excessive memory requirements.

A significantly more complex problem and one that uses the sparsified-L preconditioner is the high frequency analysis of a portion of a PCB described previously and shown

Precond. type	Size of MZM^t	Precond. factor time	Total execution time	Total number of iterations	Average # of iters. per solve
diagonal-L	751x751	0.26	450.46	729	41
cube-block	751x751	6.07	254.35	374	21
diagonal-L	1099x1099	0.81	1042.57	760	42
cube-block	1099x1099	11.86	755.12	518	29
diagonal-L	2101x2101	3.43	1901.58	760	42
cube-block	2101x2101	44.91	1381.15	502	28
diagonal-L	4351x4351	15.87	5522.79	842	47
cube-block	4351x4351	174.13	4609.96	641	36
diagonal-L	7501x7501	46.24	8894.92	883	49
cube-block	7501x7501	452.11	7309.18	635	35

Table 3-1: Execution times and iteration counts for diagonal-of-L and cube-block preconditioning of the printed circuit board example. Times are in CPU seconds for the DEC AXP3000/500.

in Fig. 3-16. To properly model the current flow in the two reference planes surrounding the copper lines, the planes must be finely discretized. Here again, as the discretization is refined, the cost of direct factorization grows like m^3, the cost of explicit GMRES grows as m^2, but the cost of multipole-accelerated GMRES grows only linearly with m as shown in Fig. 3-27. For this PCB example, the associated impedance matrix is 18x18, while the pair of traces over plane example has only a 2x2 impedance matrix. Thus, nine times as many GMRES solutions are required to compute the PCB example's impedance matrix. Even so, for a $10,000$ mesh problem, the multipole-accelerated GMRES algorithm is still over an order of magnitude faster in computation time.

From Table 3-1 it can be observed that the time to compute the preconditioners is negligible compared to the total execution time, although for larger problems, the time required to compute the cube-block preconditioner may become significant. Also, the required number of iterations for either of the preconditioners does not grow rapidly with problem size.

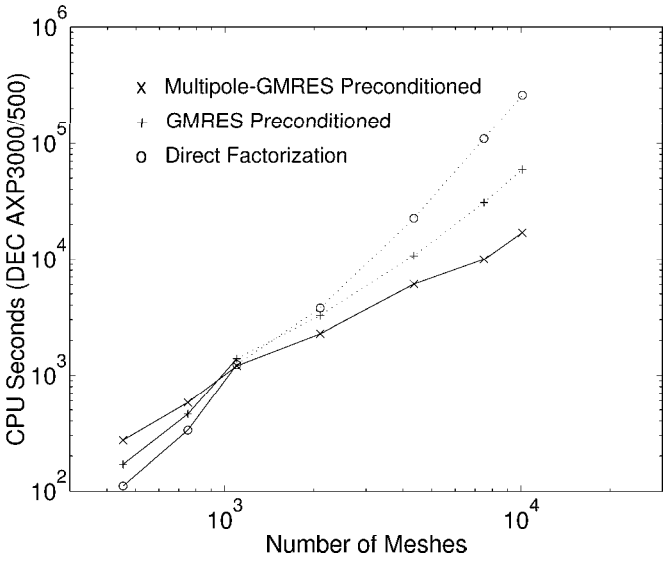


Figure 3-27: Comparison of the CPU time to compute the reduced inductance matrix for the PCB package using direct factorization, GMRES, and GMRES with with multipole acceleration.

4

Extensions for the Inductance Problem

While the methods reviewed in the previous chapter allow the analysis of geometries with tens of thousands of elements, there is still a need for more. As geometries become more complex, or higher accuracy is needed, the designer desires faster solvers which can analyze larger problems.

Geometries with large planar structures require a uniform discretization which can generate systems with an excessively large number of elements. A method of nonuniformly discretizing planar structures becomes necessary and Section 4.1 describes an approach which allows for hierarchical descriptions and modification while still enforcing current conservation.

Fast convergence of the iterative solver can be accomplished through preconditioning as was described in Chapter 3. The preconditioning technique chosen can have different performance for different classes of problems and improvement is always of interest. In Section 4.2 we explore an alternate preconditioning technique based on the work of [38].

4.1 Nonuniform planar discretization

The inductance methods of the previous chapter work particularly well for problems with a large number of elements. However, problems of many times the size or complexity may even exceed the ability of such methods. The next course is to consider ways to reduce the number of elements a priori without affecting the accuracy of the solution.

For even simple problems such as a single trace over a large finite conductivity plane, the number of elements using the discretization in Figure 3-3 can be in the thousands in order to resolve the current distribution under the trace at high frequencies as shown in

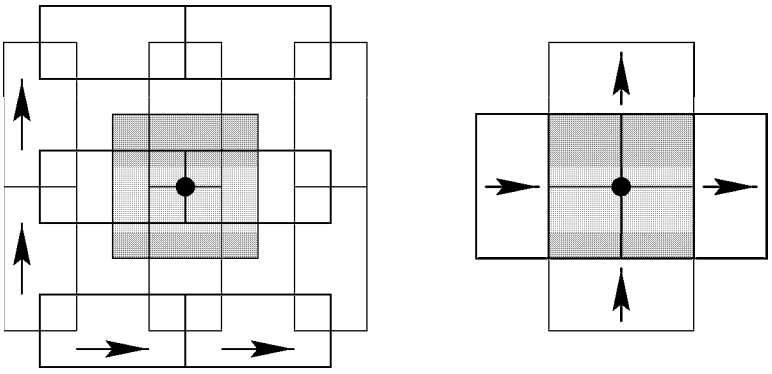


Figure 4-1: On the left is part of the discretization of a plane with the segments drawn with half their actual width. The arrows mark the direction of current for the filament. The shaded area defines a cell with the shaded area on the right figure showing the cell when filaments are drawn at their actual width

Figure 3-22. Because the current crowding underneath the trace is about the width of the trace, the discretization must be fine to capture the current distribution accurately. Since the discretizations supported in the previous section must be uniform, then the discretization must be fine everywhere. To avoid this excess, this section develops a method of nonuniformly discretizing a plane for the circuit-like discretization scheme of the previous section. First we consider a standard approach which is shown to be inadequate.

4.1.1 A standard approach

The interconnected element discretization does not lend itself immediately to an obvious approach for refinement. For instance, discretization schemes for standard finite element methods have a simple concept of a nonoverlapping element, and it is possible to divide the element into multiple elements to accomplish refinement. Here, that is not the case, and we must also enforce current conservation. A common approach when a conservation law as in 3.6 must be enforced is for the discretization to consist of nonoverlapping "cells" or control-volumes whose net flux through the cell surface is zero [1]. Refining the discretization involves breaking these cells into subcells. The PEEC discretization can be cast into such an interpretation by viewing each node in Figure 3-3 as the center of a cell and the midpoints of each of the node's filaments as center of a cell boundary edge as shown in Figure 4-1. The current which crosses any cell boundary is only that of the filament it bisects, and since current conservation is enforced at the node, it is enforced also as the flux through the cell.

Then, dividing each cell into four subcells would generate a discretization as shown

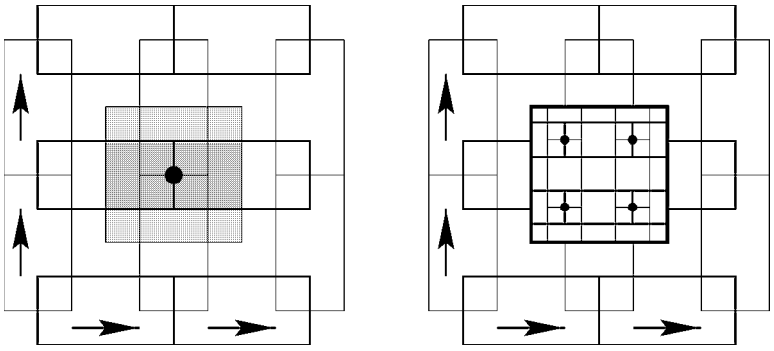


Figure 4-2: A set of subcells

in Figure 4-2. Such a discretization has the advantage that the cells are disjoint so local refinement is straightforward. It has the disadvantage, however, in that a new condition must be enforced at the cell boundary. When cells of different levels of discretization share a cell boundary, the sum of the currents on both sides must be equal. From Figure 4-3a, this amounts to requiring $I_a = I_b + I_c$. But by only this boundary condition, current traveling across the boundary is free to jump in the x-direction. If the filaments were only modeling resistance, this is equivalent to the discretization having a zero resistance path in the x-direction, which is unphysical. As an example, Figure 4-3b shows that in going from a fine to coarse discretization and back again, there is a path with zero resistance in the x-direction. This path has a slope of 10/3 which means this discretization has an inherent four percent error. Since the goal of a nonuniform discretization is to allow coarser discretizations without significantly impacting accuracy, this *inherent* error is unacceptable.

4.1.2 A different "cell" approach

Instead of the above, consider letting a cell simply be the squares bounded by four filaments of Figure 3-3. Drawing each filament with zero width gives Figure 4-4. With this line segment interpretation of cells, refinement is just dividing a cell into subcells. Any new line segment drawn to divide a cell becomes a new segment and any intersection of line segments is a new node at which current conservation is enforced. See Figure 4-5.

Since the line segments are no longer evenly spaced, the width of the corresponding filaments is determined by the distance to the next segment of the same direction. The cells and corresponding filaments for a sample node are shown in Figure 4-6. Note that as one travels from coarse to fine discretization, current cannot jump laterally as before.

With this scheme it is straightforward to generate the filaments once the distance to the closest adjacent line segment is known. Determining such information requires first

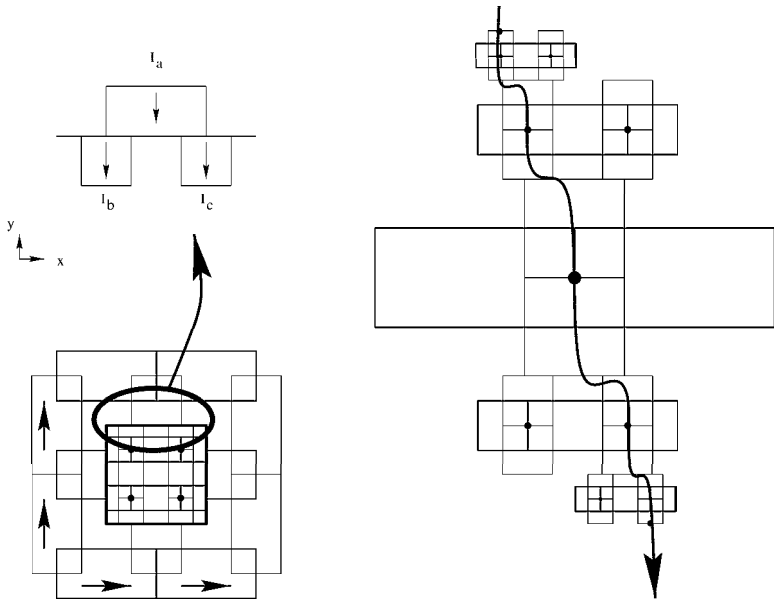


Figure 4-3: Boundary condition and a zero x-resistance path

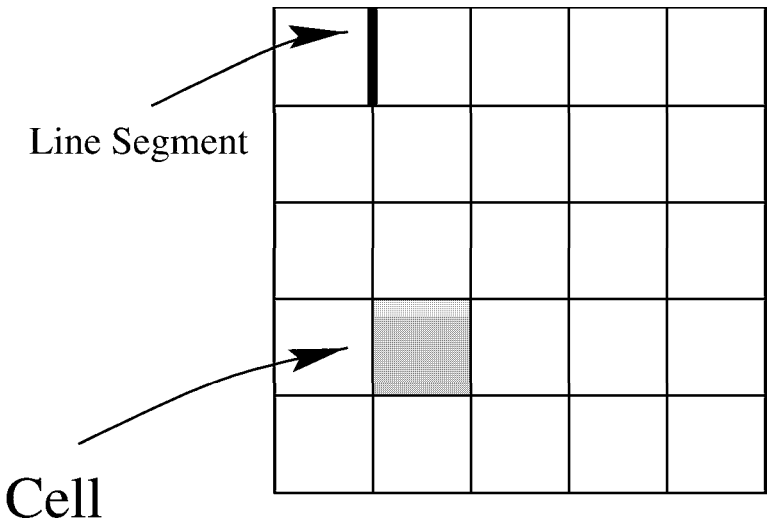


Figure 4-4: Line segment representation of a plane

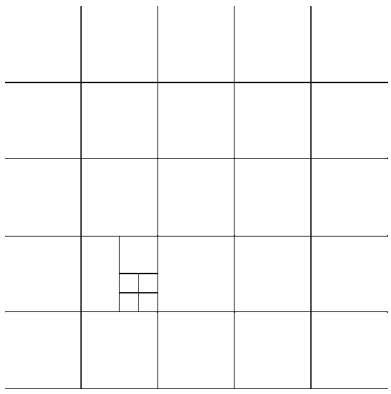


Figure 4-5: Dividing into subcells

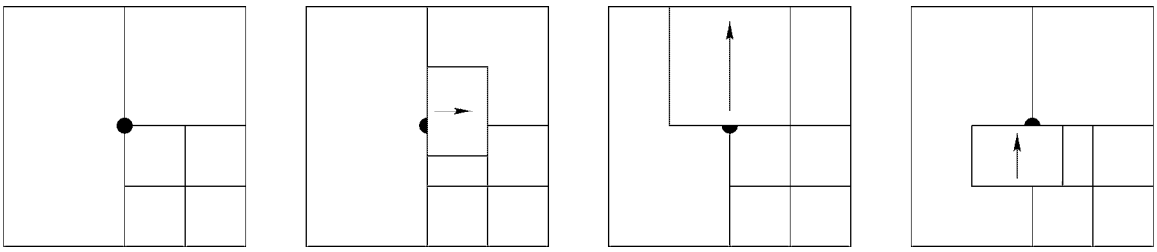


Figure 4-6: The three filaments which meet a node

a method of representing the discretization. To allow for arbitrarily fine discretizations, a hierarchical representation of the discretization is necessary. An example binary representation is shown in Figure 4-7. In the figure, the entire plane is the root cell, and dividing it in half generates two children. Dividing each of those gives grandchildren, etc.. With such a representation, it is possible to traverse the tree to find the necessary information to build the filaments.

4.1.3 Results

In this section we show results demonstrating the benefits of a nonuniform discretization for two different types of problems. Since resistance and inductance computation are intertwined, we first look at computing the resistance between two equipotential contacts to give an idea of the discretization that would be necessary to accurately capture the resistive part of the magnetoquasistatic calculation. Second, we look at an appropriate discretization to capture both the inductance and resistance of a trace over a finite conductivity ground plane.

Consider calculating the resistance between two equipotential circular contacts in a thin plane as shown in Figure 4-8. For a thin plane of *infinite* extent in x and y, the

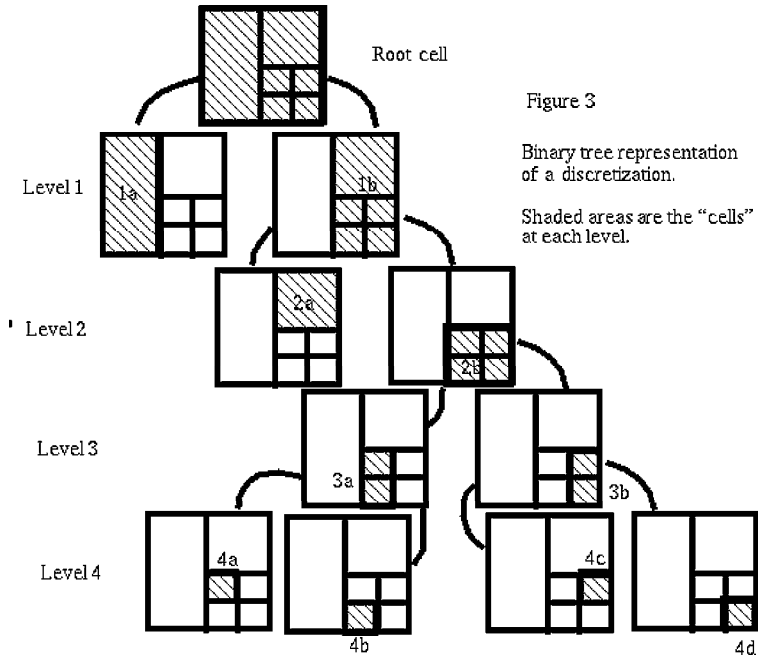


Figure 4-7: The hierarchy of cells defining a discretization

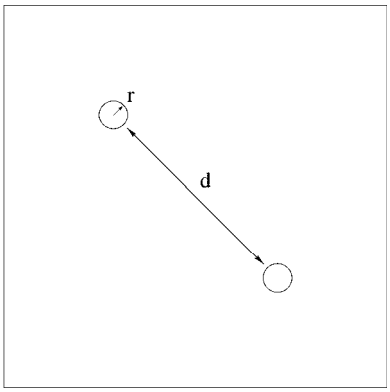


Figure 4-8: Two contacts on a thin plane

current distribution at DC does not vary in the z-direction and this problem is a matter of solving a two-dimensional Laplace equation. If d is large compared to r, the resistance between two well separated cylindrical contacts can be shown to be

$$R = \frac{1}{\pi\sigma t} ln\left(\frac{d-r}{r}\right)$$

where t is the thickness of the plane and σ is the conductivity. From the form of this equation it is clear that for small radii, the radius must be modeled accurately.

For $r = 0.01$ m, $d = \sqrt{2}m$ and $\sigma = 5.8 \times 10^7 \frac{S}{m}$, R is calculated as $R = 2.71 \times 10^{-6}\Omega$. Using a uniform discretization on a $3m \times 3m$ plane, 70,000 elements were required to give 8% error. To use a nonuniform discretization, a method of successive refinement which improves results is needed. Using a simple approach of choosing the cell size such that the net current in all the filaments is roughly constant, a discretization similar to that in 4-9 was used to generate an 8000 element problem which resulted in less than 2% error. A nonuniform discretization is thus feasible to reduce the memory and increase the accuracy of resistive part of the magnetoquasistatic calculations.

Next, consider computing the loop inductance of a trace over a plane as in Figure 3-20. A 26 micron wide, 2 mm trace, which is 8 microns above a thick substrate is shorted to the plane on one end and driven by a source at the other. The plane is 3mm by 3mm and is discretized into 3 filaments in the thickness to capture skin effects. Using the discretization in xy shown in Figure 4-9 generated a 4884 filament problem which consumed 50 MB of memory. This was solved in less than 6 minutes on a IBM RS-6000-3BT computer compared to 188 MB and 65 minutes using a commercial finite-element solver. The commercial solver did not account for skin effects and the results were within 1% of one another.

4.1.4 Future work

This section developed a method of nonuniformly discretizing planar structures for inductance computation to save time and memory. One can easily specify a discretization, however generating an appropriate discretization is a matter which requires further investigation. Underneath a trace and near contacts a finer discretization is needed but it is difficult to determine the error incurred at a particular refinement level. Methods of adaptive grid refinement are necessary. For capturing the induced currents of the trace correctly, the adaptive approach of [63] for the capacitance problem could be applied to the inductance problem. The current distribution near contacts is close to that of the resistance problem. In fact, without inductance the uniform element discretization of

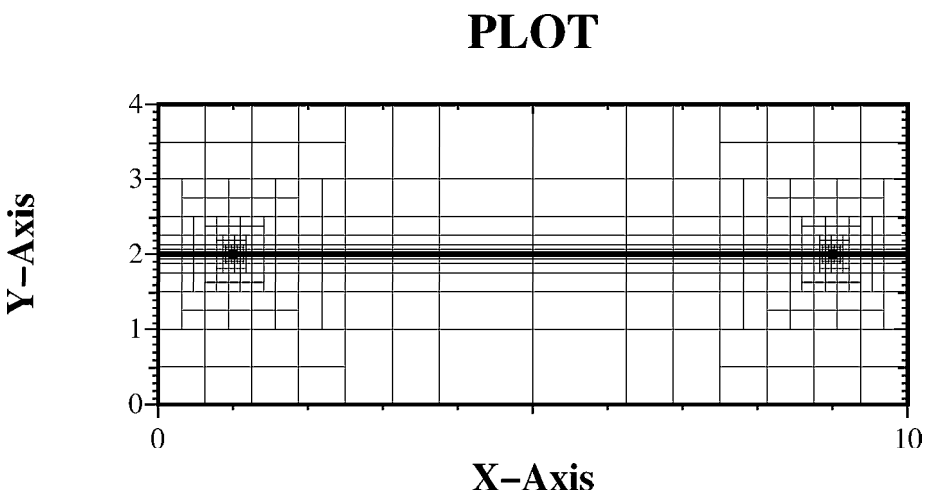


Figure 4-9: The discretization of a plane with a single trace overtop. Refinement produced where trace contacts plane and also in the y-direction underneath the trace.

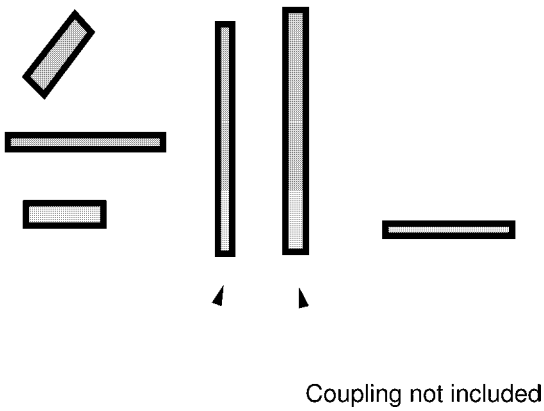


Figure 4-10: Couplings captured by block preconditioner. Two cubes shown as dashed squares. Coupling between filaments in different blocks not included in preconditioner

the plane gives exactly the finite difference grid for the pure resistance problem, ie. the Laplace equation. Methods of adaptive refinement for finite difference methods for the Laplace equation may be applicable here.

4.2 The Shell Preconditioner

As discussed in Chapter 3, iterative algorithms used to solve the dense systems of equations resulting from the integral equations of magnetoquasistatic analysis rely on preconditioning to insure fast convergence. The preconditioning matrix must be a good approximation to the inverse of the original system and be inexpensive to compute.

Various approaches to preconditioning for magnetoquasistatic analysis have been explored in Section 3.3 which show that since the original system is positive definite, so must be the preconditioner. The dominant technique presented involves deriving a positive definite sparsification of the partial inductance matrix. The simplest approach to such a sparsification is to choose a block diagonal sparsification based on a disjoint covering of space. In Section 3.3, the set of filaments in a cube formed by the multipole algorithm are chosen to form a block. Filaments contained in a block will have their coupling terms included in the preconditioner. Since these are probably the strongest couplings, the preconditioner will capture the dominant interactions. Such a choice, however, misses couplings which are near the boundary of the cubes as shown in Figure 4-10

A method for *stably* approximating the partial inductance matrix to any degree of sparsification but which captures all the nearby coupling was proposed by [38]. The

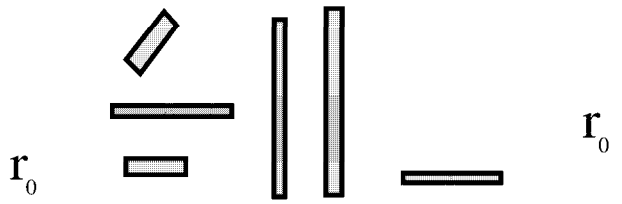


Figure 4-11: Coupling captured by shells

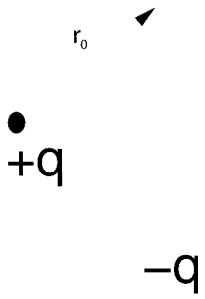


Figure 4-12: Charge +q in center and -q on shell

central idea of this approach is to assume that the partial element conductor currents return at some finite and constant radius from their origin rather than from infinity. The coupling inductances within the "shells" of return current are shifted, while those outside become zero. This approach will capture *all* the nearby coupling within the shell of radius r_0 as shown in Figure 4-11.

It is easiest to understand the shell idea from a charge point of view. Consider a charge $+q$ at the origin and infinity as the zero potential reference. With infinity as the zero potential, there is a charge of $-q$ at infinity. Consider moving this $-q$ charge to a spherical shell of radius r_0 from charge $+q$ as shown in Figure 4-12. From Gauss' Law, the potential outside the shell is zero, and inside it is shifted by the constant $\frac{q}{4\pi\epsilon_0 r_0}$.

The same idea can be applied to an incremental length of current filament carrying

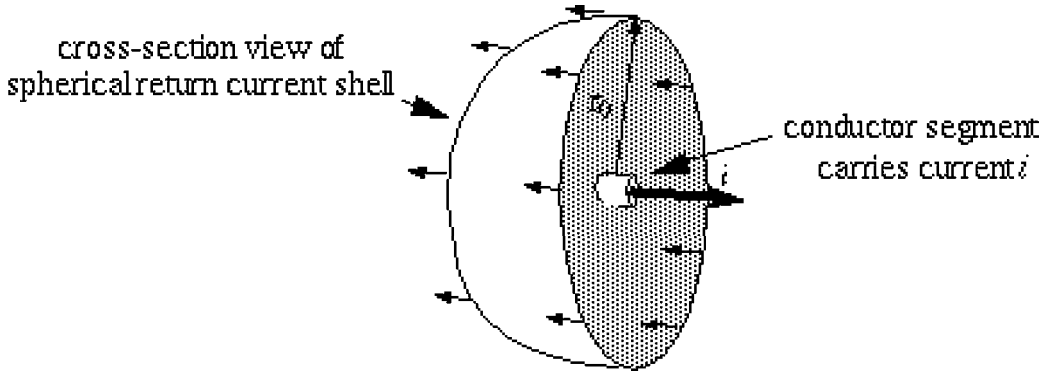


Figure 4-13: Current i in center and $-i$ on shell. Reprinted with permission of Byron Krauter.

current i as shown in Figure 4-13. Now, a shell of current in the opposite vector direction surrounds the incremental filament and the vector potential inside the shell is shifted by $\frac{\mu_0 i}{4\pi r_0}$. If this vector potential is used in forming L of (3.18), then $L_{ij} = 0$ for filaments i and j separated by more than r_0 and thus L is now a sparse approximation to the original.

It is shown in [38] that if r_0 is large enough to contain the entire geometry, then the shifted partial inductance matrix would give identical results to the original matrix. In [38], if it can be assumed that the coupling farther than some distance, r_0, is negligible, then using this distance as a radius for the shell will give a sparse, positive definite approximation to the partial inductance matrix which can be used in place of the original dense partial inductance matrix.

For the purposes of this work, we do not wish to use this approximation as a replacement for the original partial inductance matrix, but instead choose an r_0 to form a good, sparse preconditioner. By choosing the radius small enough, the resulting matrix will be sparse enough that this shell approach could be used as a preconditioner. Conductor segments which are only partially outside the shell can be treated approximately.

Figure 4-14 and Table 4-1 compare using block-diagonal preconditioners to the shell preconditioner for various radii. Clearly, the shell preconditioner converges in many fewer iterations, however the overall execution speed up is not as dramatic since the preconditioner is more dense than the block-diagonal matrix.

In conclusion, in the above implementation, there is no advantage to the new shell preconditioner. However, even though the shell preconditioner does not offer great advantages for the problem shown here, it does indicate a more robust method of capturing all the nearby coupling. As future work, it may then be possible to produce a superior preconditioner for the small radii preconditioners by more accurately treating the

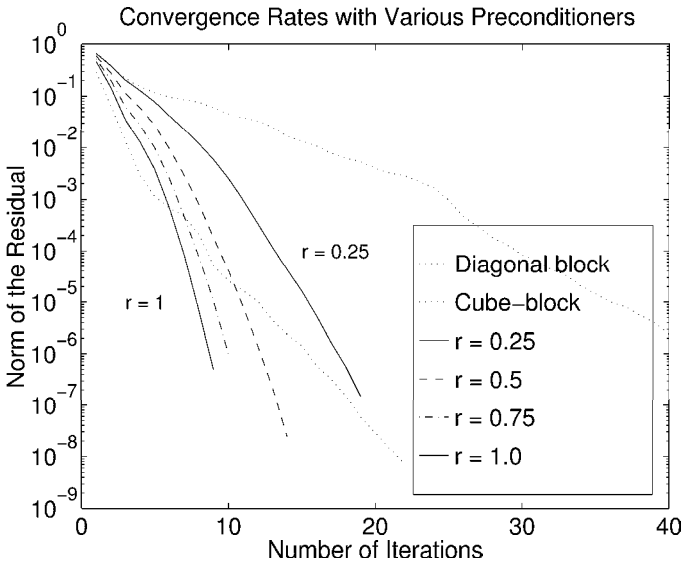


Figure 4-14: Results of using various preconditioners to solve for the admittance of a coarse discretization of a 35 pin package at a high frequency. Cube-block refers to dividing space into cubes whose side lengths are roughly 1/8th of the package width. The radii are relative to the cube side length.

Preconditioner type		avg. # iters per conductor	non-zeros in precond	total CPU time
Block Diagonal	diagonal-of-L	43.9	4613	1187
	cube-block	22.1	101671	729
Shell Current	$r = 0.25$	18.3	24829	590
	$r = 0.5$	13.5	184295	565
	$r = 0.75$	11.1	326579	595
	$r = 1.0$	10	432121	656

Table 4-1: Results using various preconditioners for computing the 35×35 admittance matrix

conductor segments that are only partially outside the current shells.

5

Model Order Reduction for the Inductance Problem

In Chapter 2 it was observed that a simple first order model, $R+sL$, where R and L are single lumped values, is only adequate to capture the behavior of the interconnect over a limited frequency range. To review, the example in Figure 2-10 showed a slight deviation from a single lumped model before the first resonance, but later in Figure 3-24 we saw that the frequency dependence can lead to an order of magnitude deviation in the inductance. For such large deviations in digital design, perhaps one might choose a lumped model which uses the smallest inductance over the frequency range of interest. This however would underestimate the inductive coupling in other regions. But a design which uses the worst case, or largest inductance, would be more conservative than necessary.

We therefore desire to derive models which are valid over a broader range but are still of low order for efficient incorporation in a circuit simulator. While the model reduction problem is well known in system theory, we require a method which is computationally tractable for very high order systems and also numerically robust. To this end we investigate using a moment-matching variant of an eigenvalue computation technique known as Arnoldi's method to generate low order models for the inductance problem. Much of what is developed here is also highly relevant to the developments for the full electromagnetoquasistatic case of Chapter 6.

5.1 Background for Coupled Interconnect-Circuit Simulation

The end use of interconnect models is for simulation with circuit devices. Since these devices are generally nonlinear, simulation must be performed in the time domain. The

inductance methods described in the previous sections, however, have computed the admittance at individual frequency points only. The most straightforward approach to including general frequency-domain data in a circuit simulator is to calculate the associated time-domain impulse response using an inverse fast Fourier transform [60]. Then, the response of the interconnect at any given time can be determined by convolving the impulse response with an excitation waveform. Such an approach is too computationally expensive for use in general circuit simulation, as it requires that at every simulator time step, the impulse response be convolved with the entire computed excitation waveform. An alternative approach is to approximate the frequency-domain representation with a rational function, in which case the associated convolution can be accelerated using a recursive algorithm [41].

Both these approaches suffer from requiring the frequency domain data be computed at many discrete frequency points. This can be too computationally expensive since each point requires solving the mesh impedance matrix, MZM^T in (3.36).

Instead, we would like to go directly from (3.36) to a low order time domain description. The single lumped R, $L \in \mathbb{R}$ description of Figure 2-4 is of course a first order model,

$$(R + sL)i = v$$

which can be written in the time-domain as

$$L\frac{di}{dt} = -Ri + v.$$

We wish to derive a system of equations of the form

$$\mathcal{L}\frac{d\boldsymbol{I}}{dt} = -\mathcal{R}\boldsymbol{I} + \boldsymbol{V} \tag{5.1}$$

where now $\mathcal{L}, \mathcal{R} \in \mathbb{R}^{k\times k}$ and this system captures the frequency dependence of the inductance and resistance. Since circuit simulators such as SPICE [49] solve nonlinear systems of ordinary differential equations, systems of the form (5.1) fit naturally into these solvers.

An obvious choice for $\mathcal{L}$ and $\mathcal{R}$ is to use (3.36) which gives

$$\begin{aligned} \tfrac{d}{dt}(\boldsymbol{M}\boldsymbol{L}\boldsymbol{M}^T)\boldsymbol{I}_m &= -(\boldsymbol{M}\boldsymbol{R}\boldsymbol{M}^T)\boldsymbol{I}_m + \boldsymbol{N}\boldsymbol{V}_t \\ \boldsymbol{I}_t &= \boldsymbol{N}^T\boldsymbol{I}_m. \end{aligned} \tag{5.2}$$

where $\boldsymbol{Z}$ has been expanded to $\boldsymbol{R} + j\omega\boldsymbol{L}$, $j\omega$ replaced by $\frac{d}{dt}$, and $\boldsymbol{V}_t$ and $\boldsymbol{I}_t$ re-introduced as the terminal quantities from (1.1) where $\boldsymbol{I}_t = \boldsymbol{N}^T\boldsymbol{I}_m$, $\boldsymbol{V}_s = \boldsymbol{N}\boldsymbol{V}_t$, with $\boldsymbol{N} \in \mathbb{R}^{m\times t}$ as an easily constructed terminal incidence matrix. Such an approach is equivalent to placing

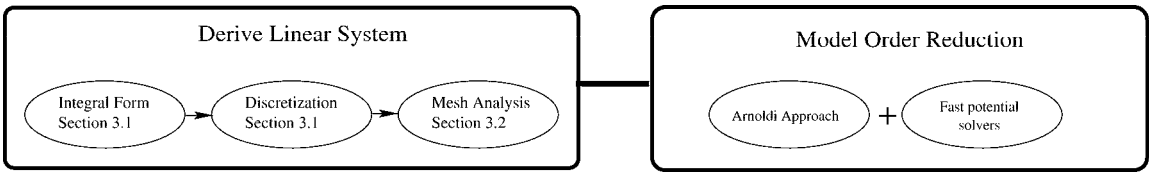


Figure 5-1: Connection of Model Reduction Methods to Inductance

the entire PEEC RL circuit into the circuit simulator as is done in [55]. Unfortunately, if the PEEC circuit is placed directly into a circuit simulator, at every simulation time step, one is effectively solving (3.36). For modern complex packaging geometries, such computation is intractable as shown in Chapter 3. We thus seek a reduced order version of (5.2) which still captures the dominant frequency dependent behavior, but is small enough for efficient circuit simulation.

Since in the following development we will be going directly from (5.2) to a reduced order model, no iterative matrix solution will be necessary but we will still be able to apply the fast multipole approach for $O(n)$ computation. To illustrate these connections, we alter the layout figure of Chapter 3 as shown in Figure 5-1.

5.2 Approaches to Model Order Reduction

The topic of model order reduction appears in many disciplines (see for instance [14]). In control systems, perhaps the best known approaches are that of Truncated Balanced Realizations [45] (TBR) or Hankel singular values [17]. Other popular approaches are based on the modal properties of the system, such as the method of Selective Modal Analysis (SMA) which is known in the power systems community [51]. Unfortunately, these methods require $O(n^3)$ computation since they require explicit knowledge of the entire eigenspectrum of the system as in SMA, or the Hankel singular values as in TBR. For complicated packaging structures for which n can easily exceed 10^4, such methods would require days to compute and gigabytes of memory.

In the area of circuit simulation, asymptotic waveform evaluation (AWE) [53] has popularized the use of model reduction via Padé approximation [4]. Computing Padé approximates are comparatively inexpensive compared to TBR and SMA. Also, as a desirable property for circuit simulation, they provide a systematic method for enforcing successively more accurate representations of the approach to steady-state. To describe

this more formally, write a single input and a single output of (5.2) generally as

$$\begin{aligned} \boldsymbol{A}\tfrac{d}{dt}\boldsymbol{I}_m &= \boldsymbol{I}_m + \boldsymbol{b}u \\ y &= \boldsymbol{c}^T\boldsymbol{I}_m. \end{aligned} \tag{5.3}$$

where $\boldsymbol{A} = -(\boldsymbol{MRM}^T)^{-1}(\boldsymbol{MLM}^T)$ and $\boldsymbol{b} = (\boldsymbol{MRM}^T)^{-1}\boldsymbol{N}_j$ and $\boldsymbol{c} = \boldsymbol{N}_i$, where $\boldsymbol{N}_i$ indicates the i^{th} column of $\boldsymbol{N}$. Notice that we have chosen to define $\boldsymbol{A}$ as multiplying $\frac{d}{dt}\boldsymbol{I}_m$ rather than the traditional definition on the right, multiplying $\boldsymbol{I}_m$. This choice avoids an $\boldsymbol{A}$ whose formation would involve the computation of $(\boldsymbol{MLM}^T)^{-1}$ which would be an $O(n^3)$ dense factorization. Instead we choose $\boldsymbol{A} = -(\boldsymbol{MRM}^T)^{-1}(\boldsymbol{MLM}^T)$ since $(\boldsymbol{MRM}^T)$ is a sparse matrix and its factorization can be computed rapidly.

Applying the Laplace transform to (5.3) and then eliminating $\boldsymbol{I}_m$, (5.3) can be written as a transfer function and expanded in a MacLaurin series (Taylor series about $s = 0$),

$$\boldsymbol{Y}_{ij}(s) = \boldsymbol{c}^T\left(\boldsymbol{I} - s\boldsymbol{A}\right)^{-1}\boldsymbol{b} = \sum_{k=0}^{\infty} m_k s^k, \tag{5.4}$$

where $m_k = \boldsymbol{c}^T\boldsymbol{A}^k\boldsymbol{b}$ is the k^{th} moment of the transfer function. A diagonal Padé approximation of q^{th} order is defined as the rational function

$$\boldsymbol{Y}_q^P(s) = \frac{b_{q-1}s^{q-1} + \cdots + b_1 s + b_0}{a_q s^q + a_{q-1}s^{q-1} + \cdots + a_1 s + 1} \tag{5.5}$$

whose coefficients are selected to match the first $2q-1$ moments of the transfer function (5.4). Matching more moments corresponds to more accurate representations of the approach to steady state since powers of s around $s = 0$ in the Laplace domain correspond in the time domain to the response and its derivatives as $t \to \infty$.

The Padé approximates can be computed using direct evaluation of the moments which involves the accurate computation of the vectors $\{\boldsymbol{b}, \boldsymbol{Ab}, \boldsymbol{A}^2\boldsymbol{b}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{b}\}$. Explicitly computing these vectors, however, is ill-conditioned because as k is increased, $\boldsymbol{A}^k\boldsymbol{b}$ and $\boldsymbol{A}^{k+1}\boldsymbol{b}$ quickly converge to the dominant eigenvector of $\boldsymbol{A}$ and thus become linearly dependent to numerical precision (See Figure 5-2). Information in the $(k+1)^{st}$ moment is thus lost. More numerically robust methods are needed as developed next.

5.3 Order Reduction using Arnoldi Iterations

A more robust approach of deriving a moment matching reduced order model is to use an Arnoldi process [2]. The Arnoldi process has its origins in eigenvalue computation but as we shall see, it can be used to generate moment matching reduced order models.

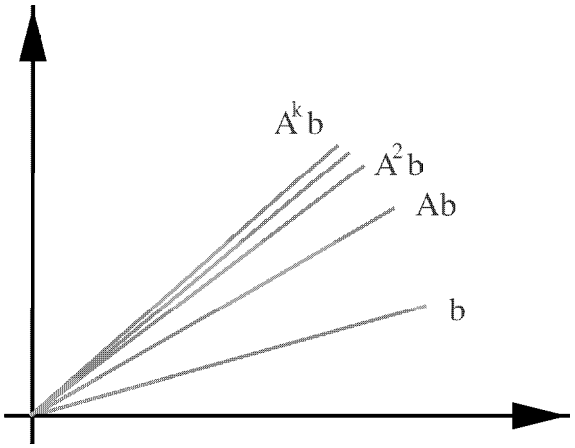


Figure 5-2: A two dimensional example of $\boldsymbol{A}^k\boldsymbol{b}$ converging to a single vector as k grows.

The idea behind this approach is similar to that of [13, 19, 15], and is that of computing an orthonormal basis for the space spanned by the vectors $\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{b}\}$ without explicitly computing each of $\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \cdots, \boldsymbol{A}^{k-1}\boldsymbol{b}\}$. The orthogonality between the basis vectors makes the Arnoldi algorithm a better conditioned process than direct evaluation of the moments. The space spanned by the vectors is commonly called the Krylov subspace, denoted $\boldsymbol{\mathcal{K}}_k(\boldsymbol{A}, \boldsymbol{b})$. The basic outline of the Arnoldi process is given in Algorithm 5.3.1.

After q steps, the Arnoldi algorithm returns a set of q orthonormal vectors, as the columns of the matrix $\boldsymbol{V}_q \in \mathbb{R}^{m\times q}$, and a $q \times q$ upper Hessenberg matrix $\boldsymbol{H}_q$ whose entries are the Gram-Schmidt orthogonalization coefficients $h_{i,j}$. These two matrices satisfy the following relationship:

$$\boldsymbol{A}\boldsymbol{V}_q = \boldsymbol{V}_q\boldsymbol{H}_q + h_{j+1,j}\boldsymbol{v}_{q+1}\boldsymbol{e}_q^T \tag{5.6}$$

where $\boldsymbol{e}_q$ is the q^{th} column of the identity matrix in $\mathbb{R}^{q\times q}$.

The matrices $\boldsymbol{V}_q$ and $\boldsymbol{H}_q$ can be related directly to the moments. Since $\boldsymbol{v}_1 = \boldsymbol{b}/\left\|\boldsymbol{b}\right\|_2$ and with repeated use of (5.6), it can be shown that after q steps of an Arnoldi process, for $k < q-1$,

$$\boldsymbol{A}^k\boldsymbol{b} = \left\|\boldsymbol{b}\right\|_2 \boldsymbol{A}^k\boldsymbol{V}_q\boldsymbol{e}_1 = \left\|\boldsymbol{b}\right\|_2 \boldsymbol{V}_q\boldsymbol{H}_q^k\boldsymbol{e}_1. \tag{5.7}$$

With this relation, the moments can be related to $\boldsymbol{H}_q$ by

$$m_k = \boldsymbol{c}^T\boldsymbol{A}^k\boldsymbol{b} = \left\|\boldsymbol{b}\right\|_2 \boldsymbol{c}^T\boldsymbol{V}_q\boldsymbol{H}_q^k\boldsymbol{e}_1$$

and so the q^{th} order Arnoldi-based approximation to $\boldsymbol{Y}_{ij}$ can be written as

$$\boldsymbol{Y}_q^A(s) = \left\|\boldsymbol{b}\right\|_2 \boldsymbol{c}^T\boldsymbol{V}_q\left(\boldsymbol{I} - s\boldsymbol{H}_q\right)^{-1}\boldsymbol{e}_1 \tag{5.8}$$

Algorithm 5.3.1 (Arnoldi process).

```
arnoldi(in:A, b, q; out:V_q, v_{q+1}, H_q, h_{j+1,j})
{
    v_1 = b/||b||_2
    for (j = 1; j <= q; j++) {
        /* Get next vector in space */
        w = A v_j
        /* Orthogonalize new vector to previous vectors */
        for (i = 1; i <= j - 1; i++) {
            h_{i,j} = w^T v_i
            w = w - h_{i,j} v_i
        }
        h_{j+1,j} = ||w||_2
        if (h_{j+1,j} != 0) {
            /* normalize */
            v_{j+1} = w/h_{j+1,j}
        }
    }
    V_q = [v_1 ... v_q]
    H_q = (h_{i,j}),  i, j = 1, ..., q
}
```

corresponding to the state-space realization

$$\begin{aligned} \boldsymbol{H}_q \dot{\boldsymbol{x}} &= \boldsymbol{x} + \boldsymbol{e}_1 u \\ y &= (\|\boldsymbol{b}\|_2 \boldsymbol{c}^T \boldsymbol{V}_q)\boldsymbol{x} \end{aligned} \tag{5.9}$$

which can be written as the triplet $[\boldsymbol{A}_r, \boldsymbol{b}_r, \boldsymbol{c}_r] = [\boldsymbol{H}_q, \boldsymbol{e}_1, \|\boldsymbol{b}\|_2 \boldsymbol{V}_q^T \boldsymbol{c}]$.

Note that the rational function $\boldsymbol{Y}_q^A(s)$ is *not* a Padé approximation. Equation (5.7) is only valid for $k < q-1$, thus $\boldsymbol{Y}_q^A(s)$ has q poles but only matches $q-2$ moments. Conversely, $\boldsymbol{Y}_q^P(s)$ matches $2q-1$ moments. The Arnoldi approach is non-optimal in a moment matching sense compared to the Padé-via-Lanczos (PVL) approach which robustly computes a q^{th} order Padé approximant as in [13, 19, 15]. However, it is not clear the effect on the accuracy of the resultant model as we will see in Section 5.5. Also, later we will see that one can use the Arnoldi approach to "trade in" these unmatched moments for properties such as stability and passivity.

5.4 Treating Multiple Inputs and Outputs

The presentation in the previous section described a method of generating reduced order models using a single input and single output of a multiple input, multiple output system. Thus, to generate reduced order models for the entire $Y_t \in \mathbb{R}^{t \times t}$ would naively require the generation of t^2 different models. If used inside a circuit simulator, these models would add qt^2 states to the circuit system, where q is the order of each model. Intuitively this seems inefficient since all the models share a common system matrix $\boldsymbol{A}$ and are likely to share many of the same poles.

For the Arnoldi approach, we see immediately that since the computation of $\boldsymbol{H}_q$ and $\boldsymbol{V}_q$ involved only $\boldsymbol{b}$, we can directly replace $\boldsymbol{c} = \boldsymbol{N}_i$ in (5.9) with all the outputs, $\boldsymbol{C} = \boldsymbol{N}$. The resultant single input, multiple output system will be a model for an entire column of Y_t and will still match $q-2$ moments for each of the entries of the column. Now one is required to only compute t different models, one for each of the inputs, $\boldsymbol{b}$.

We can take this one step further and use a block version of the Arnoldi algorithm which handles all t inputs at once [59]. The outline of a basic block Arnoldi process is given in Algorithm 5.4.1.

After q steps, the block Arnoldi algorithm returns a set of q orthonormal blocks as the block columns of the matrix $\boldsymbol{V}_q^b \in \mathbb{R}^{m \times tq}$, and a $tq \times tq$ upper band Hessenberg matrix $\boldsymbol{H}_q^b$ whose entries are $t \times t$ blocks $\boldsymbol{H}_{i,j}$. These two matrices satisfy a relationship similar to that in (5.6), namely

$$\boldsymbol{A}\boldsymbol{V}_q^b = \boldsymbol{V}_q^b \boldsymbol{H}_q^b + \boldsymbol{V}_{q+1} \boldsymbol{H}_{q+1,q} \boldsymbol{E}_q^T \tag{5.10}$$

Algorithm 5.4.1 (Block Arnoldi process).

```
arnoldi(input A, B, q; output V_q^b, V_{q+1}, H_q^b, H_{j+1,j}, R_1)
{
   B = V_1 R_1     (QR Factorization)
   for (j = 1; j <= q; j++) {
      W = A V_j
      for (i = 1; i <= j - 1; i++) {
         H_{i,j} = V_i^T W
         W = W - V_j H_{i,j}
      }
      W = V_{j+1} H_{j+1,j}     (QR Factorization)
   }
   V_q^b = [V_1 ··· V_q]
   H_q^b = (H_{i,j}),  i, j = 1, ···, q
}
```

where $\boldsymbol{E}_q$ is the matrix of the last t columns of the identity in $\mathbb{R}^{tq \times tq}$.

From (5.10), the q^{th} order block Arnoldi-based approximation to $\boldsymbol{Y}_{ij}$ can be written as

$$\boldsymbol{Y}_q^{bA}(s) = \boldsymbol{R}_1 \boldsymbol{C}^T \boldsymbol{V}_q^b \left(\boldsymbol{I} - s\boldsymbol{H}_q^b \right)^{-1} \boldsymbol{E}_1 \tag{5.11}$$

corresponding to the state-space realization using the triplet $[\boldsymbol{A}_r, \boldsymbol{B}_r, \boldsymbol{C}_r] = [\boldsymbol{H}_q^b, \boldsymbol{E}_1, \boldsymbol{R}_1 \boldsymbol{V}_q^{b^T} \boldsymbol{C}]$ where $\boldsymbol{R}_1$ results from the QR factorization of $\boldsymbol{B}$.

The total cost of computing the matrix transfer function using the block Arnoldi algorithm is $\mathcal{O}(tq)$ resulting in an approximation of order tq.

5.5 Complexity Comparisons

The efficiency of a model reduction technique has two facets. The first is the computational cost of generating the reduced order model and the second is the compactness of the reduced model for a given level of accuracy. To address the first, because $\boldsymbol{L}$ is dense, the dominant cost of an Arnoldi process is the matrix-vector product, $\boldsymbol{Ax} = -(\boldsymbol{MRM}^T)^{-1}(\boldsymbol{MLM}^T)\boldsymbol{x}$ computed at each step of the process. In practice, the matrix-vector cost dominates even when the dense part, $(\boldsymbol{MLM}^T)\boldsymbol{x}$, is rapidly computed with a fast potential solver such as the hierarchical multipole-algorithm discussed in Section 3.2.2.

Method	# mv prods	# nonzeros in		
		$\boldsymbol{A}_r$	$\boldsymbol{B}_r$	$\boldsymbol{C}_r$
Padé	$2tq$	$\approx 3t^2q$	t^2q	t^2q
Arnoldi	tq	$\approx \frac{1}{2}tq^2$	tq	t^2q
Block Arnoldi	q	$\approx \frac{1}{2}q^2$	tq	tq

Table 5-1: Costs for a q^{th} order approximation of a t-input, t-output system (number of matrix vector products and number of nonzeros in the system matrices).

To address the second facet, the compactness is judged by the number of nonzero entries in the triplet $[\boldsymbol{A}_r, \boldsymbol{B}_r, \boldsymbol{C}_r]$ of the reduced system. For instance, for block Arnoldi, $\boldsymbol{A}_r = \boldsymbol{H}_q$ is upper-block Hessenberg which has roughly $\frac{1}{2}q^2$ nonzero entries. The number of nonzeros in these matrices is directly related to the computational cost of including them in circuit simulation.

Table 5-1 compares the number of matrix-vector products and also the number of nonzeros in the reduced-order matrices for computing models for a t input, t output system via Arnoldi, block Arnoldi, and a Padé approximant (computed via a numerically robust Lanczos algorithm [13]). Note that non-block Arnoldi needs to build t q^{th} order models but Padé must build t^2 such models.

To compare accuracy of the reduced model for a fixed amount of work to generate the model, consider computing models for the small set of package pins shown in Figure 5-3. To compute the resistance and inductance matrices, the pins were discretized into three filaments along their height and four along their width producing a system of size $m = 887$. This discretization allows the modeling of changes in resistance and inductance due to skin and proximity effects. Since there are seven pins, the model to be produced has only seven inputs and seven outputs.

For the admittance between pins 1 and 2, Figure 5-4 shows the relative error for Padé and Arnoldi, and block Arnoldi admittance models with model order 4, 8, and 56, respectively. The model order was chosen such that the computation would require 56 matrix-vector products for each method. It is worth noting that Padé, Arnoldi, and block Arnoldi match 7,6, and 6 moments, respectively, for each entry of $\boldsymbol{Y}_t(s)$, yet Arnoldi gives a more accurate approximation.

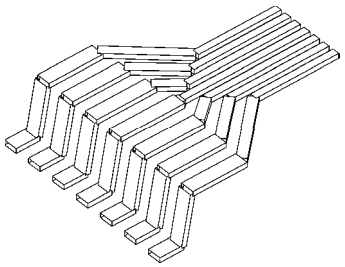


Figure 5-3: Seven pins of a cerquad pin package.

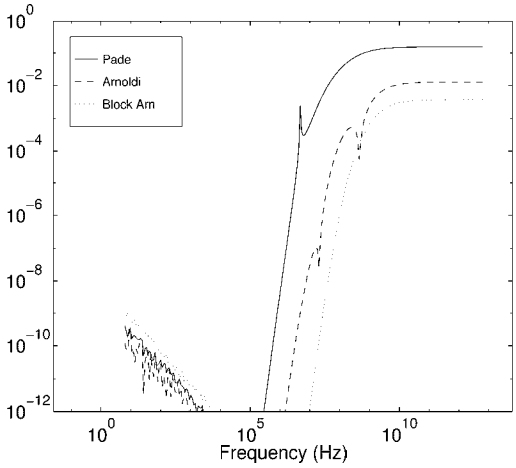


Figure 5-4: Relative error for $\boldsymbol{Y}_4^P(s)$ and $\boldsymbol{Y}_8^A(s)$. Each method required 56 matrix-vector products.

Method	Order	# mv prods	# nonzeros in		
			$\boldsymbol{A}_r$	$\boldsymbol{B}_r$	$\boldsymbol{C}_r$
Padé	8	112	1764	392	392
Arnoldi	8	56	252	56	392
Block Arn.	14	14	98	98	98

Table 5-2: Order of approximation, number of matrix vector products and number of nonzeros in the reduced-order matrices for approximation yielding an accuracy of 5%.

Finally, to demonstrate that block Arnoldi can be superior for both facets, we compare the models based on a fixed accuracy. Table 5-2 compares the computational cost and the complexity of the reduced order models for a desired accuracy of 5% pointwise error. In particular, for each entry of the reduced order admittance matrix, $\boldsymbol{Y}^r(s)$,

$$\epsilon_{ij} = \frac{|\boldsymbol{Y}_{ij}(j\omega) - \boldsymbol{Y}^r_{ij}(j\omega)|}{|\boldsymbol{Y}_{ij}(j\omega)|} < 0.05$$

for ω in the frequency range shown in Figure 5-4. The table shows that block Arnoldi requires the fewest matrix-vector products, and thus requires the least computation to generate the model. The table also shows that the reduced-order matrices for block Arnoldi have the fewest total nonzero entries and are thus the most efficient for subsequent circuit simulation.

It is worth noting that a block version of the Padè-via-Lanczos algorithm has also been developed in [13] and would have similar performance as above.

5.6 Coupled Simulation Results

To demonstrate using these models in circuit simulation, we wish to observe the crosstalk between the pins in Fig. 5-3. Assume the five middle lines carry output signals from the chip and the two outer pins carry power and ground. The signals are driven and received with CMOS inverters. The drivers are capable of driving a large current to compensate for the impedance of the package pins. The inductance of the pins is modeled as described in the previous sections, and the capacitance is assumed to be 8pF. The interconnect from the end of pin to the receiver is modeled with a capacitance of

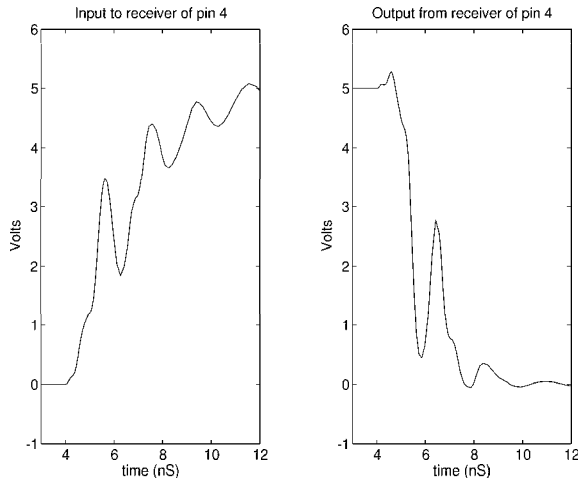


Figure 5-5: The middle pin's receiver when four adjacent pins switch 1ns after the middle pin.

5pF. A $0.1\mu F$ decoupling capacitor is connected between the driver's power and ground to minimize supply fluctuations.

Since we have a reduced model for the inductance instead of a single L value, the model for the admittance matrix is incorporated into SPICE as voltage-controlled current sources (VCCS). As a sample time domain simulation, imagine that at time $t_0 = 4$ns the signal on the middle pin of Fig.5-3 is to switch from high to low and the other four signal pins switch from low to high but due to delay on chip, the other four pins switch at $t_1 = 5$ns. In this case, significant current will suddenly pass through the late pins while the middle pin is in transition. Due to crosstalk, this large transient of current has significant effects on the input of the receiver of the middle pin, as shown in Fig. 5-5. Note that the input does not rise monotonically. Fig. 5-5 also shows that the bump in the waveform is carried through to the output of receiver, as a large glitch.

5.7 Guaranteeing Stability and Passivity

Padé approximants, whether via direct moment matching or via a Lanczos-type method as in [13], suffer from a significant drawback: For a passive original system, there is no guarantee that the approximant is passive or even stable.

Similarly, there is no guarantee for the Arnoldi approximants generated by the algorithms in the previous section. However since the development of the original Arnoldi algorithm in the previous sections, two important modifications regarding the stability and passivity of the Arnoldi algorithm have been developed [61, 50]. These results can

be viewed as exploiting the observation that a q^{th} order Arnoldi approximant matches $q-2$ moments, but a q^{th} order Padé approximant matches $2q-1$ moments. These extra unmatched moments can be "traded in" to guarantee stability or passivity. The stability and passivity results are essential for model order reduction and will be reviewed here.

5.7.1 A Guaranteed Stable Arnoldi Algorithm

It is possible to derive a version of the Arnoldi algorithm for which stability is guaranteed. While the passive algorithm described next in Section 5.7.2 is also stable, the following is useful for its insights into state-space transformations and the Arnoldi algorithm.

The main result in [61] was that the Arnoldi algorithm applied to (5.3) under a change in state-space coordinates,

$$\tilde{\boldsymbol{I}}_m = \boldsymbol{\mathcal{L}}^{\frac{1}{2}} \boldsymbol{I}_m \tag{5.12}$$

will generate a guaranteed stable reduced order model, $\boldsymbol{H}_q$, where stability implies all eigenvalues have nonpositive real parts.

To avoid computations with $\boldsymbol{\mathcal{L}}^{\frac{1}{2}}$, one can use a modified version of the Arnoldi algorithm which uses a "hiding the square-root" trick commonly used when preconditioning Conjugate-Gradient schemes [18]. The key idea is that most of the operations involve inner products of the form

$$\left(\boldsymbol{\mathcal{L}}^{\frac{1}{2}} \boldsymbol{u}\right)^T \boldsymbol{\mathcal{L}}^{\frac{1}{2}} \boldsymbol{y}. \tag{5.13}$$

If $\boldsymbol{\mathcal{L}}$ is symmetric, which is the case for RLC circuits, then (5.13) can be rewritten as $\boldsymbol{u}^T \boldsymbol{\mathcal{L}} \boldsymbol{y}$, which no longer requires the square root. The presence of the matrix $\boldsymbol{\mathcal{L}}$ can be construed as endowing $\mathbb{R}^n$ with an induced dot product, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{\mathcal{L}}} = \boldsymbol{y}^T \boldsymbol{\mathcal{L}} \boldsymbol{x}$, thus leading to what is termed a modified $\boldsymbol{\mathcal{L}}$-orthogonal version of the Arnoldi Algorithm.

Additionally, because $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{R}}$ are symmetric for the inductance problem, $\boldsymbol{H}_q$ will be tridiagonal (block tridiagonal for block Arnoldi) which is comparable to PVL in its number of nonzeros. This special structure also implies that the back orthogonalization can be truncated to only two steps.

5.7.2 Guaranteed Passivity

Very recently, a modification to the Arnoldi algorithm to guarantee *passivity* has been developed. In addition to the eigenvalues (and thus poles) of the system having nonpositive real parts, for passivity, the zeros must also. Passivity is important because

an interconnection of merely stable systems is not necessarily stable, but the interconnection of passive systems in passive. For the inductance problem, which is cast as a system of R and L, the original system is passive since it is composed of passive circuit components. We thus require that the reduced order models preserve this quality.

In [50], given the original system of m states,

$$s\mathcal{L}x = -\mathcal{R}x + BV_t \tag{5.14}$$

$$I_t = C^t x. \tag{5.15}$$

the idea is to use the Arnoldi vectors $\boldsymbol{V}_q$ where $0 < q \leq m$ in a congruence transform [37] to preserve passivity. This corresponds to the reduced system

$$s\tilde{\mathcal{L}}x = -\tilde{\mathcal{R}}x + \tilde{B}V_t \tag{5.16}$$

$$I_t = \tilde{C}^t x. \tag{5.17}$$

where $\tilde{\mathcal{L}} = \boldsymbol{V}_q^T \mathcal{L} \boldsymbol{V}_q$, $\tilde{\mathcal{R}} = \boldsymbol{V}_q^T \mathcal{R} \boldsymbol{V}_q$, $\tilde{B} = \boldsymbol{V}_q^T B$, and $\tilde{C} = \boldsymbol{V}_q^T C$. The main result in [50] is that this reduced order model matches $q-2$ moments and is proven passive under the following conditions:

1. $\tilde{B} = \tilde{C}$
2. $\boldsymbol{z}^T(\mathcal{R} + \mathcal{R}^T)\boldsymbol{z} \geq 0$ for all $\boldsymbol{z}$
3. $\boldsymbol{z}^T(\mathcal{L} + \mathcal{L}^T)\boldsymbol{z} \geq 0$ for all $\boldsymbol{z}$

For inductance extraction, the first condition is satisfied since port models always have every input pair as an output pair. The second and third conditions are satisfied since $\mathcal{R}$ and $\mathcal{L}$ are both symmetric positive definite matrices.

5.8 Beyond Inductance Extraction

This chapter developed the Arnoldi algorithm as method of efficiently generating guaranteed passive reduced order models for the inductance problem. For problems in which accurate inductance is important, these models can replace the single lumped R and L of the simple models of Chapter 2 for improved accuracy.

As we also saw in Chapter 2, decoupling the inductance and capacitance does not adequately predict the response of the interconnect at higher frequencies. In Part II we couple these quantities which requires many of the same issues described previously in this part.

Part II

6

Coupled Capacitance and Inductance Extraction

In Chapter 2 it was observed that the simple circuits of Figures 2-3,2-4, and 2-5 were not adequate to accurately capture even the first resonant frequency of the interconnect shown. In this chapter we develop an approach for low order model generation for interconnect which does not neglect displacement current as in Part I. Instead, it solves the full quasistatic Maxwell's equations which we term "electromagnetoquasistatic" analysis or EMQS.

In Section 6.1 we see that we can extend the Partial Element RL circuit interpretation to the EMQS domain to form a distributed RLC circuit representation of the governing integral equations. A mesh analysis approach is then pursued to exploit many of the developments of the mesh formulation for the inductance problem.

With the return of displacement current (capacitive effects), passive model order reduction is not straightforward as before. As Section 6.2 describes, care must be taken in choosing an appropriate state space realization from the discretization.

Previously in Chapter 5, the computational cost of generating a reduced order model for the inductance problem was on the same order as solving for the inductance and resistance at a single frequency point. In particular, the matrix-vector product cost at each iteration of a single frequency solve of MZM^t in (3.36) was roughly the same as computing a matrix vector product with $\mathcal{R}^{-1}\mathcal{L} = (MRM^t)^{-1}(MLM^t)$ in the Arnoldi algorithm. Thus, a single frequency solution which takes 40 iterations to converge would be roughly the same as computing a 40th order reduced model.

For the electromagnetoquasistatic (EMQS) problem, this is no longer the case. Each matrix-vector product for model order reduction requires a *solution* to a capacitance problem. To make computation of a reduced order model tractable, the computational

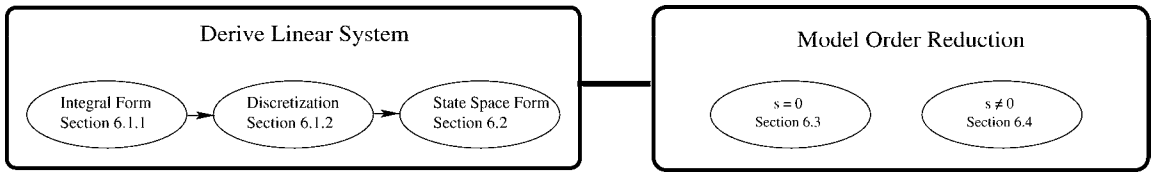


Figure 6-1: Outline of Chapter

cost of these many full capacitive solves must be reduced. An approach which "recycles" the Krylov subspaces to this end is described in Section 6.3.1.

Computing reduced order models with $A = \mathcal{R}^{-1}\mathcal{L}$ represents an expansion of the transfer function of (5.4) about $s = 0$. In Section 6.3.3 we shall see that this choice is not adequate, and the need to expand about nonzero s is important. While expansions about $s = 0$ require a capacitive solve at each step, expansions about nonzero s unfortunately require solutions of the full EMQS equations. Fortunately, the mesh formulation lends itself to effective preconditioning as will be shown in Section 6.4. With a means for efficient multipoint approximations, the chapter concludes by showing that efficient generation of compact reduced order models is possible. An outline of the chapter is illustrated in Figure 6-1.

6.1 Formulation and Discretization

This section derives the linear system solved to determine the admittance relation, $\boldsymbol{Y}_t$. It begins by deriving an integral equation from Maxwell's equations and then discusses the circuit-like discretization to generate a linear system. Following a mesh based approach has a slightly different character for the electromagnetoquasistatic problem and it will be contrasted against nodal analysis.

6.1.1 Integral Equation Formulation

Several integral equation-based approaches have been used to derive $\boldsymbol{Y}_t$ associated with a given package or interconnect structure [55, 7]. The approach followed here is the standard integral equation approach based on a "superposition" of the sources point of view whose derivation can be found in many places [25, 27] and will be reviewed here. These integral formulations are derived without the quasistatic assumption. From Maxwell's equations under Laplace transformation,

$$\nabla \times \boldsymbol{E} = -s\mu\boldsymbol{H} \tag{6.1}$$

$$\nabla \times \boldsymbol{H} = s\epsilon \boldsymbol{E} + \boldsymbol{J} \tag{6.2}$$

$$\nabla \cdot (\epsilon \boldsymbol{E}) = \rho \tag{6.3}$$

$$\nabla \cdot (\mu \boldsymbol{H}) = 0 \tag{6.4}$$

where s is the Laplace transform complex frequency.

Taking the divergence of (6.2) and using (6.3) gives the charge conservation,

$$\nabla \cdot \boldsymbol{J} = -s\rho. \tag{6.5}$$

From this point, we wish to eliminate the field quantities, $\boldsymbol{E}$ and $\boldsymbol{H}$, in favor of the current density, $\boldsymbol{J}$, and applied voltages. From Gauss' Law of magnetic flux, (6.4), the magnetic flux can be written as

$$\mu \boldsymbol{H} = \nabla \times \boldsymbol{A} \tag{6.6}$$

where $\boldsymbol{A}$ is the vector potential. Applying this to (6.1),

$$\nabla \times (\boldsymbol{E} + s\boldsymbol{A}) = 0. \tag{6.7}$$

This implies that there exists a scalar function, Φ, such that

$$-\nabla\Phi = \boldsymbol{E} + s\boldsymbol{A} \tag{6.8}$$

where Φ will be called the scalar potential. We require one final relation to relate the vector potential, $\boldsymbol{A}$ to the current density, $\boldsymbol{J}$. To uniquely determine $\boldsymbol{A}$, we must also specify its divergence. In Chapter 3, the Coulomb gauge was chosen, but here we choose the Lorentz gauge,

$$\nabla \cdot \boldsymbol{A} = -\epsilon\mu s\Phi, \tag{6.9}$$

and using (6.6) in (6.2), and the identity, $\nabla \times (\nabla \times \boldsymbol{A}) = \nabla(\nabla \cdot \boldsymbol{A}) - \nabla^2 \boldsymbol{A}$, we arrive at

$$\nabla^2 \boldsymbol{A} - \mu\epsilon s^2 \boldsymbol{A} = -\mu \boldsymbol{J} \tag{6.10}$$

and thus

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu}{4\pi} \int_{V'} \frac{\boldsymbol{J}(\boldsymbol{r}')e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|} dv' \tag{6.11}$$

where V' is the volume of all conductors and $c = \frac{1}{\sqrt{\mu\epsilon}}$.

To derive a similar relation for Φ, use (6.8) in (6.3) which gives

$$\nabla \cdot \epsilon(-\nabla\Phi - s\boldsymbol{A}) = \rho.$$

Applying (6.9) gives

$$\nabla^2\Phi - \mu\epsilon s^2\Phi = -\rho.$$

An integral relation between the charge density and scalar potential is then

$$\Phi(\boldsymbol{r}) = \frac{1}{4\pi\epsilon}\int_{V'}\frac{\rho(\boldsymbol{r}')e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|}dv'. \tag{6.12}$$

To derive an integral equation for the potentials and currents alone, we note that inside the conductors, by Ohm's law,

$$\boldsymbol{J} = \sigma\boldsymbol{E}. \tag{6.13}$$

where σ is the conductivity. Equation (6.13) can be used to show that ρ is essentially zero on the interior of conductors. Using (6.13) and (6.3) in (6.5),

$$\frac{d\rho}{dt} = -\frac{\sigma}{\epsilon}\rho \tag{6.14}$$

gives

$$\rho = \rho_0 e^{-\frac{\sigma}{\epsilon}t} \tag{6.15}$$

which states that the charge inside a conductor is zero for all time apart from any initial charge which dissipates very quickly. Charge is thus restricted to the boundary and (6.5) becomes

$$\nabla\cdot\boldsymbol{J}(\boldsymbol{r}) = 0, \qquad \boldsymbol{r}\in D \tag{6.16}$$

$$\boldsymbol{n}\cdot\boldsymbol{J}(\boldsymbol{r}) = -s\rho_s(\boldsymbol{r}), \qquad \boldsymbol{r}\in S \tag{6.17}$$

where D is the union of the interior of all conductors and $S = \partial D$ are the surfaces. For regions of S which correspond to terminals, $S_{terminals} \in S$, we allow for external current, $\boldsymbol{J}_t$ and then (6.17) becomes

$$\boldsymbol{n}\cdot(\boldsymbol{J}(\boldsymbol{r}) - \boldsymbol{J}_t) = -s\rho_s(\boldsymbol{r}), \qquad \boldsymbol{r}\in S_{terminals}. \tag{6.18}$$

S, $S_{terminals}$, and D are illustrated in Figure 6-2.

Substituting (6.11) and (6.13), into (6.8) yields the following integral equation,

$$\frac{\boldsymbol{J}(\boldsymbol{r})}{\sigma} + \frac{s\mu}{4\pi}\int_{V'}\frac{\boldsymbol{J}(\boldsymbol{r}')e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|}dv' = -\nabla\Phi(\boldsymbol{r}). \qquad \boldsymbol{r}\in D \tag{6.19}$$

Equation (6.19) with boundary conditions (6.17) and sources injected via (6.18) can now be solved for the conductor current density, $\boldsymbol{J}$, and scalar potential, Φ where Φ is now

$$\Phi(\boldsymbol{r}) = \frac{1}{4\pi\epsilon}\int_{S}\frac{\rho_s(\boldsymbol{r}')e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|}dv', \qquad \boldsymbol{r}\in\mathbb{R}^3. \tag{6.20}$$

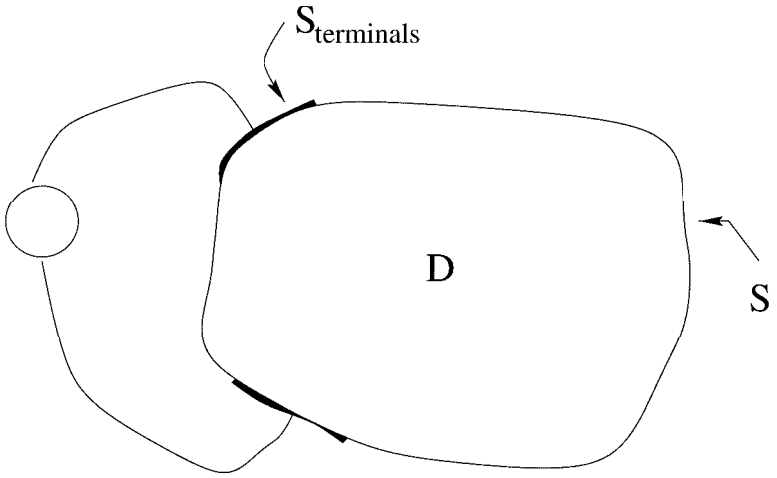


Figure 6-2: The interior D, surface S, and terminal $S_{terminals}$ regions of a conductor

The relation between $\boldsymbol{J}_t$ and Φ on $S_{terminals}$ can be used to compute $Y_t(s)$.

In this work, we consider structures which are small compared to the smallest wavelength of interest, that is,

$$\frac{s}{c}|\boldsymbol{r}-\boldsymbol{r}'| \ll 1 \Longrightarrow e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|} \approx 1. \tag{6.21}$$

Using this approximation, (6.20) and (6.11) become

$$\Phi(\boldsymbol{r}) = \frac{1}{4\pi\epsilon}\int_S \frac{\rho_s(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}dv', \qquad \boldsymbol{r}\in\mathbb{R}^3, \tag{6.22}$$

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu}{4\pi}\int_{V'} \frac{\boldsymbol{J}(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|}dv', \qquad \boldsymbol{r}\in\mathbb{R}^3. \tag{6.23}$$

This approximation provides the convenience that the kernel of the integral operators, $\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|}$, is independent of s and thus the matrices derived in the next section are frequency independent.

6.1.2 Discretization

The integral equations in the previous section are exactly those used in the original PEEC derivation [55]. As was done in Chapter 3, we will first follow the approach given there to generate a discretization of (6.19) and (6.20) and then derive the mesh formulated approach.

To generate a finite dimensional system to numerically solve (6.19), consider using a piecewise constant method of moments [24] scheme as for the inductance problem.

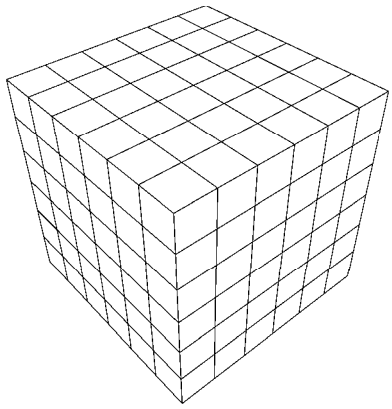


Figure 6-3: A cube of conductor with surface discretized into panels

The discretization for $\boldsymbol{J}$ is identical to the inductance problem and gives the relation in (3.18), repeated here,

$$(R + sL)I_b^f = \tilde{\Phi}_A - \tilde{\Phi}_B \tag{6.24}$$

where $I_b^f \in \mathbb{C}^b$ is the vector of b filament currents,

$$R_{ii} = \frac{l_i}{\sigma a_i} \tag{6.25}$$

is the $b \times b$ diagonal matrix of filament DC resistances,

$$L_{ij} = \frac{\mu}{4\pi a_i a_j} \int_{V_i} \int_{V_j'} \frac{\mathbf{l_i} \cdot \mathbf{l_j}}{|\boldsymbol{r} - \boldsymbol{r}'|} dV' \, dV \tag{6.26}$$

is the $b \times b$ dense, symmetric positive definite matrix of partial inductances, and $\tilde{\Phi}_A$ and $\tilde{\Phi}_B$ are the averages of the potentials over the cross sections of the filament faces.

For the charge, the surface of the conductors is divided into *panels*, as shown for a cube in Figure 6-3 each which holds a uniform charge density. The approximated charge density can then be written as

$$\rho_s(\boldsymbol{r}) = \sum v_i(\boldsymbol{r}) q_i, \qquad \boldsymbol{r} \in S.$$

where $v_i(\boldsymbol{r})$ is one if $\boldsymbol{r}$ is on panel i, zero otherwise.

As in Chapter 3, the filaments are each made branches in a network circuit graph and the junction between filaments are the nodes of the graph. To enforce (6.17), the panels are added to the circuit at nodes on the surface of the conductors. For the cube in Figure 6-3 above, the circuit of filaments and panels at a corner is shown in Figure 6-4 and the placement of panels on one of the surfaces of the cube is shown in Figure 6-5.

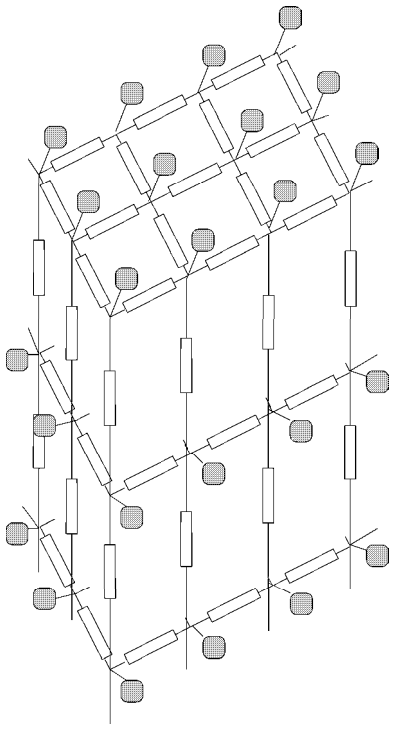


Figure 6-4: Discretization and interconnection of filaments and panels at a corner of the slab. Rectangles are filaments, shaded squares are panels. Reprinted with permission from Nuno Marques

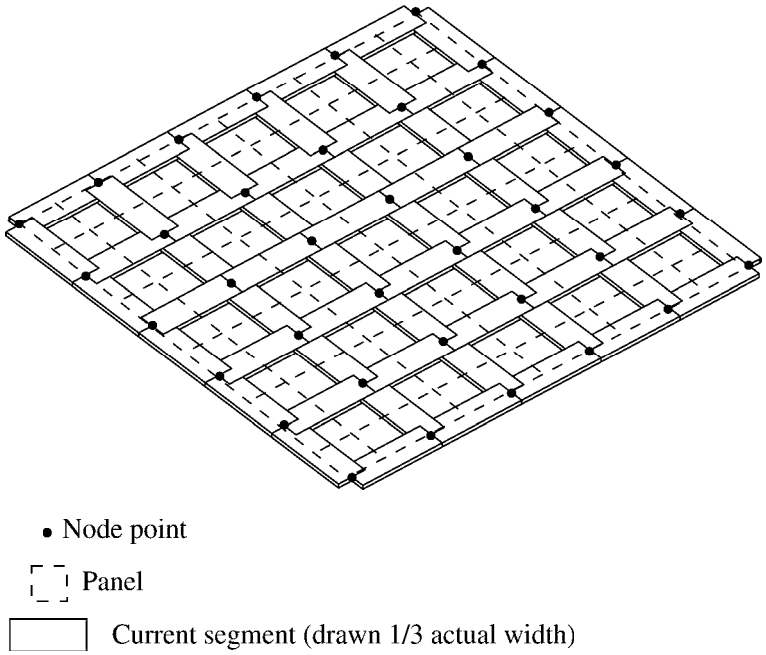


Figure 6-5: The placement of filaments and panels on the top surface of the cube

The last relation is that of the potential, Φ, to the charge, q, from (6.22). Approximating the average over the face, $\tilde{\Phi}_A$, by its value at the appropriate node point, the potential becomes

$$\Phi_n = P' q_p, \tag{6.27}$$

where $\Phi_n \in \mathbb{C}^n$ is the vector of the n node voltages, $q_p \in \mathbb{C}^p$ is the charge on each of p panels, and $P' \in \mathbb{R}^{(n_e+n_i)\times n_e}$ is the potential coefficient matrix given by

$$P'_{ij} = \frac{1}{A_j 4\pi\epsilon_0} \int_{p_j} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}'|} dV' \tag{6.28}$$

where p_j is the surface of panel j, A_j is its area, r_i is the i^{th} node location, $n_e = p$ is the number of node points on the surface and n_i is the number of internal node points. P' can be divided separately into its contribution to the internal and external nodes by

$$P' = \begin{bmatrix} P \\ P^i \end{bmatrix} \tag{6.29}$$

where $P \in \mathbb{R}^{n_e \times n_e}$ and $P^i \in \mathbb{R}^{n_i \times n_e}$.

The above relations, (6.24), (6.27), and (6.17), respectively, give the following linear

system,

$$\begin{bmatrix} Z & 0 & A^T \\ 0 & P' & -I \\ A_e & sI & 0 \end{bmatrix} \begin{bmatrix} I_b^f \\ q_p \\ \Phi_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I_t \end{bmatrix} \tag{6.30}$$

where $Z = R + sL$, I is the identity matrix of appropriate dimension, I_t are the terminal currents, $A^T = [A_e^T \; A_i^T]$ is the nodal incidence matrix providing the differencing of Φ_n and A_e enforces the boundary condition (6.17).

Eliminating Φ_n and q gives

$$(sR + s^2L + A^T P' A_e) I_b^f = A^T P' I_t \tag{6.31}$$

This is the PEEC formulation of [55]. In the original work, the elements were created as circuit elements and sent to a circuit simulator which would effectively assemble and solve (6.30).

It is worth noting that in the interior of conductors, (6.30) does not *explicitly* enforce current conservation as in (6.16). Thus, while an exact solution to the integral equation will satisfy current conservation, there is no guarantee that the discrete version will also. Additionally, the asymmetry of (6.30) makes passive model order reduction difficult since positive semidefiniteness (conditions 2 and 3 in Section 5.7.2) is difficult to determine.

Finally, as for the inductance problem, (6.30) is poorly conditioned for iterative solution, and thus we seek a different formulation more amenable to iterative solution. The advantages for iterative solution will become apparent for multipoint model-order reduction in Section 6.4.

6.1.3 An Alternate Formulation

To derive a different formulation so the we can apply passive model order reduction, for $\boldsymbol{r} \in D$ consider explicitly enforcing current conservation, (6.16),

$$\nabla \cdot \boldsymbol{J}(\boldsymbol{r}, s) = 0, \qquad \boldsymbol{r} \in D \tag{6.32}$$

but no longer enforcing the relation between charge and potential, (6.20). The two choices are equivalent, since with charge only on the boundary, (6.20) is equivalent to

$$\nabla^2 \Phi - \mu\epsilon s^2 \Phi = 0, \qquad \boldsymbol{r} \in D \tag{6.33}$$

and taking the divergence of (6.8) and using the gauge condition, (6.9), and Ohm's law, (6.13), we arrive at

$$\frac{1}{\sigma} \nabla \cdot \boldsymbol{J} = -\nabla^2 \Phi + \mu\epsilon s^2 \Phi$$

Thus (6.33) is equivalent to (6.32).

In the rest of this section, we derive a nodal formulation using this new condition. This nodal form will be similar to Modified Nodal Analysis from circuit simulation [69]. From a nodal form, it is straightforward to derive a mesh formulation as is shown in the next section. It is possible to skip directly to deriving a mesh formulation by deriving mesh quantities as "divergence-free" basis functions and following an approach similar to [3], however such an approach does not illustrate the connection to circuit methods and will not be followed here.

Before modifying (6.30) to enforce $\nabla \cdot \boldsymbol{J}(\boldsymbol{r}, s) = 0$, we first wish to alter it to allow multiple panels to be connected to a single surface node. The panel discretization can then be refined independent of the filament discretization to capture, for instance, the sharp changes in charge density at an edge which does not necessarily correspond to changes in the potential. Equation (6.30) then becomes

$$\begin{bmatrix} Z & 0 & -A_e^T & -A_i^T \\ 0 & P & -A_q^T & 0 \\ 0 & P^i & 0 & -I \\ A_e & sA_q & 0 & 0 \end{bmatrix} \begin{bmatrix} I_b^f \\ q_p \\ \Phi_n^e \\ \Phi_n^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ I_t \end{bmatrix} \tag{6.34}$$

where Φ_n has been divided corresponding to internal and external nodes, A_q *sums* the charges at each node and A_q^T requires that the potential of all the panels at a node are equal. Next, for convenience, replace the panel charge with the current into the panel, $\frac{d}{dt} q_p = I_b^p$. Enforcing current conservation, $\nabla \cdot \boldsymbol{J}(\boldsymbol{r}, s) = 0$, on the interior now involves replacing the set of equations involving P^i with $A_i I_b^f = 0$,

$$\begin{bmatrix} Z & 0 & -A_e^T & -A_i^T \\ 0 & P/s & -A_q^T & 0 \\ A_e & A_q & 0 & 0 \\ A_i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_b^f \\ I_b^p \\ \Phi_n^e \\ \Phi_n^i \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ I_t \\ 0 \end{bmatrix} \tag{6.35}$$

Combining each of the 2×2 blocks in (6.35) into single blocks gives

$$\begin{bmatrix} Z_{EM} & -\tilde{A}^T \\ \tilde{A} & 0 \end{bmatrix} \begin{bmatrix} I_b \\ \Phi_n \end{bmatrix} = \begin{bmatrix} 0 \\ I_t \end{bmatrix} \tag{6.36}$$

which is similar in form to the sparse tableau matrix for the inductance problem, (3.33).

As for the inductance problem, the poor conditioning leads us to derive a mesh analysis version of (6.36).

6.1.4 A Mesh Analysis Formulation

To derive a mesh analysis version of (6.36) from that of Chapter 3, we need only add meshes for the new branch currents associated with panels, I_b^p. Since the panel node voltages in (6.27) are voltages relative to infinity, we can view the panel branches as connecting the panel node to the zero volt node at infinity. Then the panel branch voltages are given by

$$V_b^p = \Phi_{n_e} - 0 = \Phi_{n_e}. \tag{6.37}$$

With this definition for the panel branch voltages, Z_{EM} becomes the constitutive relation for the electromagnetoquasistatic problem

$$V_b = \begin{bmatrix} V_b^f \\ V_b^p \end{bmatrix} = \begin{bmatrix} Z & 0 \\ 0 & P/s \end{bmatrix} \begin{bmatrix} I_b^f \\ I_b^p \end{bmatrix} = Z_{EM} I_b. \tag{6.38}$$

With panels defined as branches in the circuit, we can now define meshes for them which pass through infinity, as shown in Figure 6-6. There are now three types of mesh currents: those involving only filaments on the interior as for the inductance problem, I_m^f, those on the surface which include both a filament and two panels, I_m^s, and those involving only panels for nodes with multiple panels, I_m^p. This categorization leads to three block rows in the new M matrix,

$$MV_b = \begin{bmatrix} M_f & 0 \\ M_{fs} & M_{ps} \\ 0 & M_p \end{bmatrix} \begin{bmatrix} V_b^f \\ V_b^p \end{bmatrix} = \begin{bmatrix} V_s^f \\ V_s^s \\ V_s^p \end{bmatrix} = V_s \tag{6.39}$$

Applying M as before, we get the system,

$$MZ_{EM}M^T I_m = V_s. \tag{6.40}$$

which has a matrix structure similar to that of the mesh formulated inductance problem, (3.36).

Note that in the integral formulation, the sources were known injected currents, $\boldsymbol{J}_t$, and the potential and charge were unknown on $S_{terminals}$. In the mesh formulation, the potential will be specified on $S_{terminals}$ with $\boldsymbol{J}_t$ and the charge as unknowns. $\boldsymbol{J}_t$ will be added as an unknown by adding a branch to the circuit for the source, and requiring a mesh relation for the source as shown on the left in Figure 6-6. This will be described in more detail in Section 6.2.

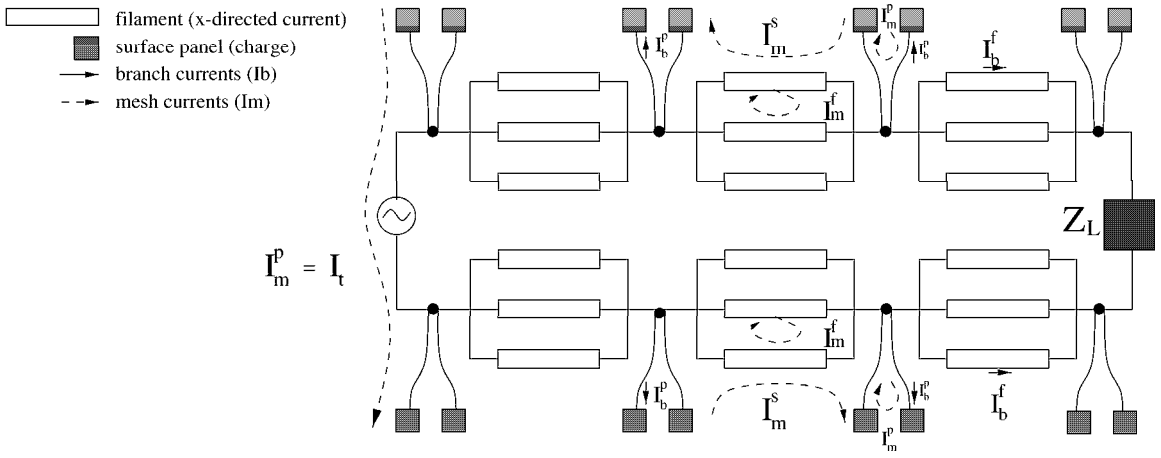


Figure 6-6: A circuit describing the mesh quantities for a 2 conductor TEM line terminated with a load Z_L. Reprinted with permission from Nuno Marques

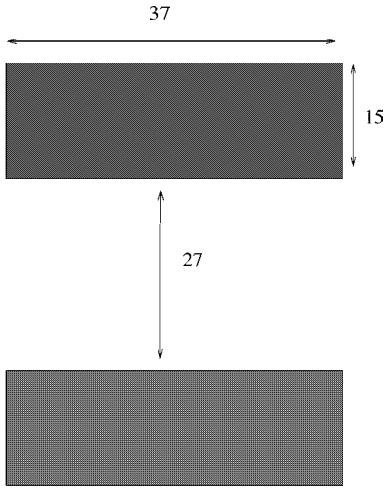


Figure 6-7: The cross section of a two-dimensional transmission line. Units are in microns

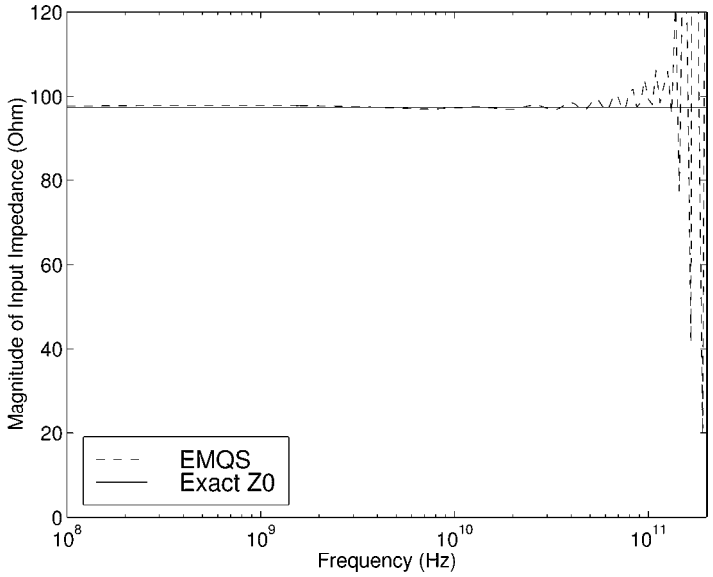


Figure 6-8: Impedance looking into a matched transmission line for the discretized system

6.1.5 Formulation Results

To show that the above formulation gives correct results consider modeling a long two conductor transmission line with this 3D tool. If the line is chosen very long, using 2D TEM transmission line analysis approximates the results well. Figure 6-7 shows the 2D cross section of a two-conductor transmission line derived from a multichip module. Using other electromagnetic tools, the 2D inductance per unit length and capacitance per unit length were computed. With these values, the characteristic impedance of the line was computed as $Z_0 = 97.43\Omega$. By using a matched termination the input impedance is theoretically constant, $Z(f) = Z_0$, for all frequencies f. The two conductors are each 1 cm long and discretized into 558 filaments, and 768 panels. Such a discretization leads to a 2033 state system. As seen from Figure 6-8, the EMQS approximation matches to within 1 percent the exact solution up to nearly 30 GHz. The inability to obtain an error significantly less than 0.1 percent may be due to either resistive effects modeled by the mesh formulation which were not in the 2D simulation, or inaccuracies in the computation of Z_0.

Note that the wavelength at 30 GHz is 1 cm which is the length of the structure modeled. It thus seems that for this example, the applicability of EMQS can extend beyond the small-compared-to-a-wavelength regime. This topic will be addressed in Appendix A.

As a second example, consider analyzing a printed circuit board connector from Teradyne, Inc, as shown in Figure 6-9 and 6-10. There are 4 thin pins surrounding the middle

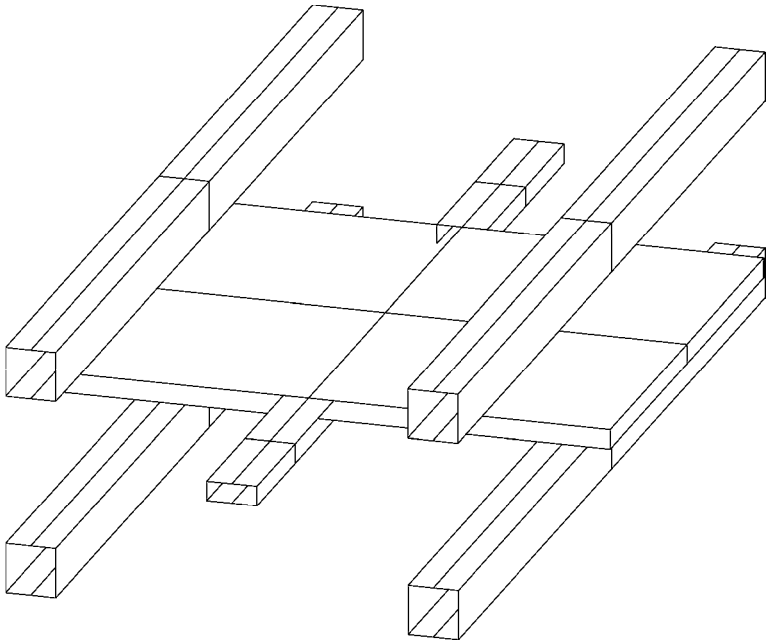


Figure 6-9: A 5 conductor connector

ground conductor which widens in the center to provide shielding. Here the number of states in the original discretized system is 3018 which was generated from 845 filaments and 1182 panels.

One of the pins is driven by a source at one end and terminated with a 50 Ω resistor at the other as shown in Figure 6-11. Frequencies in the gigahertz regime are high enough that the return current in the middle ground conductor will be bunched underneath the pin overhead as shown in Figure 6-11 and previously observed for inductance in Figure 3-22. Note that if the structure were very long, the current distribution along the length would be uniform, except at the junction between sections A and B as shown in Figure 6-11. Therefore, a 2D solution would give accurate results.

To compare to the 2D solution, the connector was stretched by a factor of ten in its length direction. The impedance of the interconnect can be computed by performing two-dimensional transmission-line analysis on sections A and B separately, and then cascading the results. From Figure 6-12 we see that the mesh formulated approach matches well to the 2D analysis for entry (1,1) of the impedance matrix, $\boldsymbol{Z}_t(s) = \boldsymbol{Y}_t(s)^{-1}$.

Next, consider comparing the two methods using the actual length of the connector. The 2D analysis should give exactly the same results as before, just shifted down by a factor of ten in frequency. However, the 3D region is more significant and from Figure 6-13, we see the results diverging.

Finally, to show strong 3D effects, the connector length is shrunk by half, and we

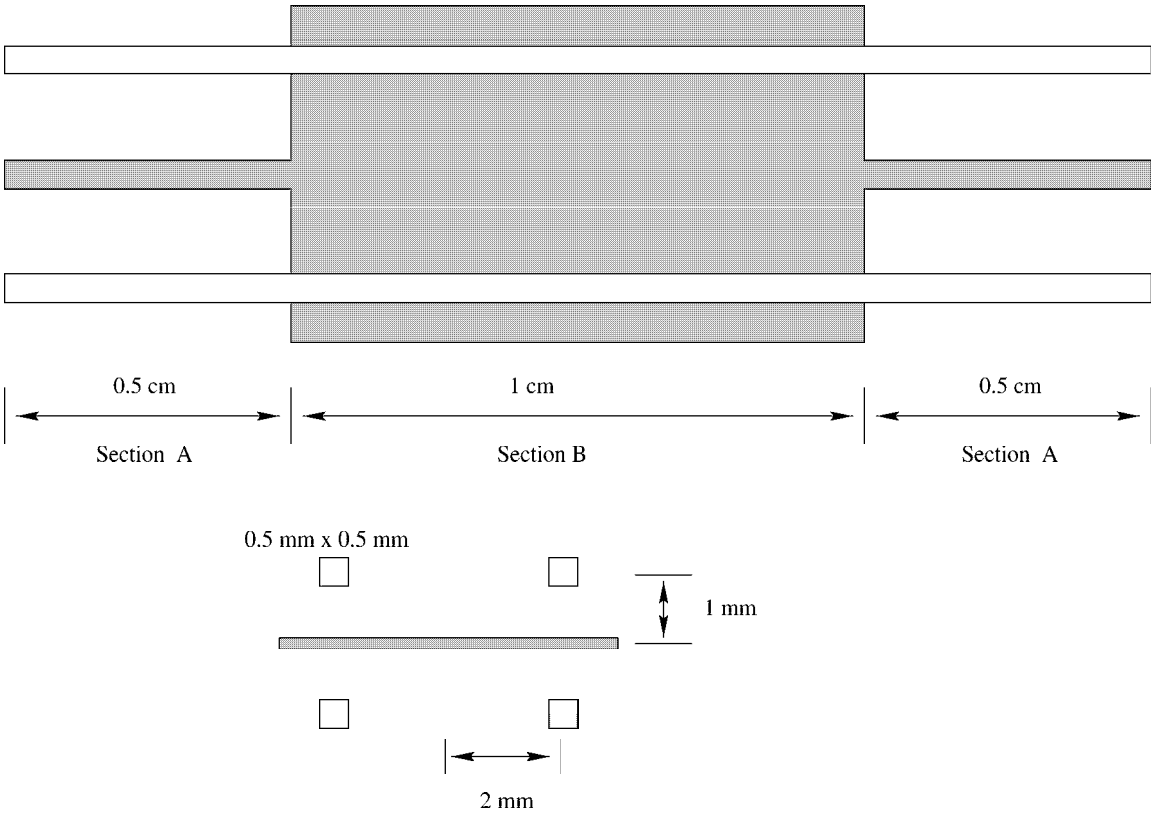


Figure 6-10: Some dimensions of the connector

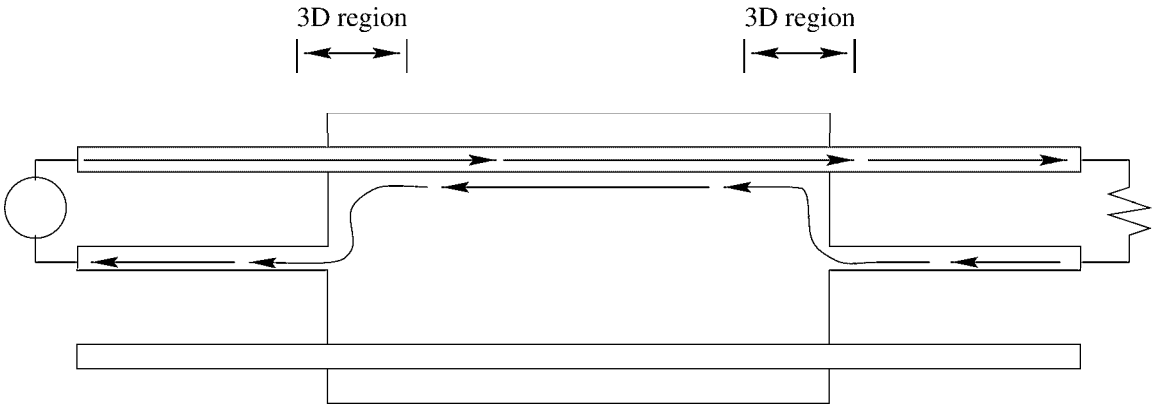


Figure 6-11: Current in connector

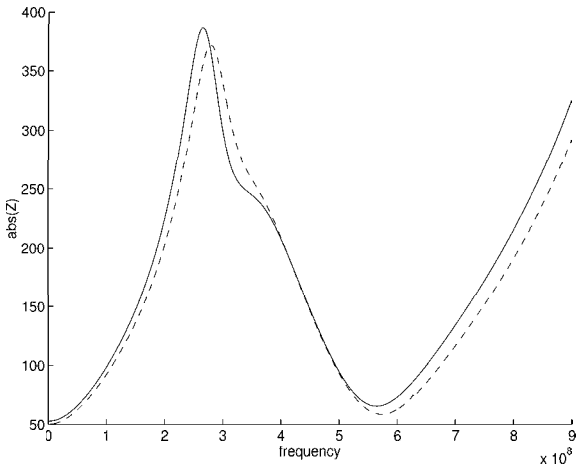


Figure 6-12: Two-dimensional analysis (solid line) compared to EMQS (dashed line) for the connector at 10x its normal length

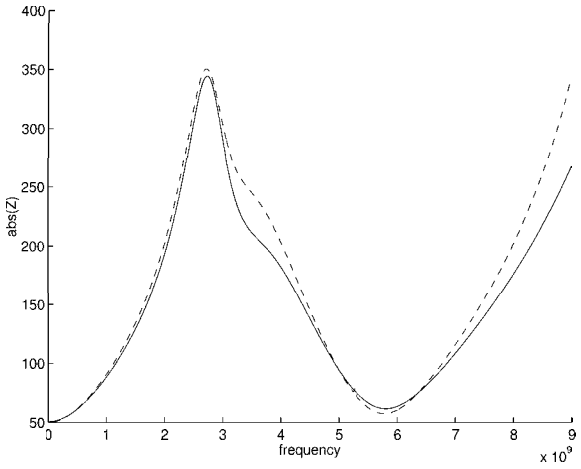


Figure 6-13: Two-dimensional analysis (solid line) compared to EMQS (dashed line) for the connector at its normal length

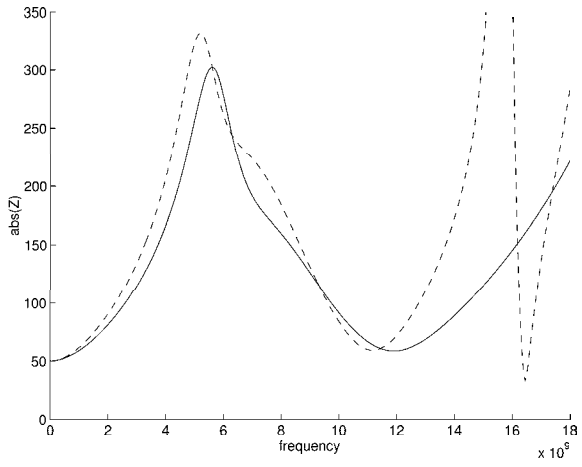


Figure 6-14: Two-dimensional analysis (solid line) compared to EMQS (dashed line) for the connector at half its normal length

see a large difference in the results in Figure 6-14. In particular a resonance, previously above the maximum frequency of the previous plots, has moved to a lower frequency. To explain this physically, the current in the 2D model can "jump" from the middle pins in section A to directly underneath the pin in section B, skipping the lateral flow in the 3D region illustrated in Figure 6-11. For the 3D model, the current must travel this path and this extra distance changes the effective length of the interconnect, thus shifting the resonance downward.

We now have a mesh formulated EMQS approach to which fast iterative techniques could be applied for single frequency solves. But computing the frequency response above via single frequency solves would be very expensive. Also, for coupled circuit simulation, the value of $\boldsymbol{Y}_t(s)$ at a single frequency is of little use. For this reason we turn to model-order reduction in Section 6.2 to generate a low order version of (6.40). The issue of single frequency solution will reappear for multipoint expansions in Section 6.4.

6.2 Deriving a State-Space Realization

We seek to apply methods of model order reduction to (6.40). We thus need to derive a first order state-space realization of the second order system (6.40). Care must be taken to derive a realization that has appropriate properties for passive model order reduction. Additionally, as for $\mathcal{R}$ in Chapter 5, we wish the corresponding $\mathcal{R}$ for (6.40) to be both sparse and nonsingular for expansions about $s = 0$.

6.2.1 A standard approach

To follow an approach similar to the standard approach for deriving a system of first order ordinary differential equations from a single higher order equation, we separate (6.40) into powers of s

$$M \begin{bmatrix} R + sL & 0 \\ 0 & P/s \end{bmatrix} M^t = M\tilde{R}M^t + sM\tilde{L}M^t + s^{-1}M\tilde{P}M^t \tag{6.41}$$

where

$$\tilde{R} = \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} \tag{6.42}$$

$$\tilde{L} = \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} \tag{6.43}$$

$$\tilde{P} = \begin{bmatrix} 0 & 0 \\ 0 & P \end{bmatrix}. \tag{6.44}$$

Defining $s\tilde{I}_m = I_m$, and, as before, $V_s = NV_t$ and $I_t = N^t I_m$ then gives

$$s \begin{bmatrix} M\tilde{L}M^t & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I_m \\ \tilde{I}_m \end{bmatrix} = \begin{bmatrix} -M\tilde{R}M^t & -M\tilde{P}M^t \\ I & 0 \end{bmatrix} \begin{bmatrix} I_m \\ \tilde{I}_m \end{bmatrix} + \begin{bmatrix} N \\ 0 \end{bmatrix} V_t \tag{6.45}$$

$$I_t = [N^t\, 0] \begin{bmatrix} I_m \\ \tilde{I}_m \end{bmatrix} \tag{6.46}$$

where I is the identity matrix. We abbreviate the above as

$$s\mathcal{L}x = -\mathcal{R}x + BV_t \tag{6.47}$$

$$I_t = C^t x. \tag{6.48}$$

To apply the methods of Chapter 5, we must be able to efficiently compute matrix-vector products of the form $\mathcal{R}^{-1}\mathcal{L}x$. Thus we require that $\mathcal{R}$ be nonsingular. Unfortunately, this is not the case for $\mathcal{R}$ in (6.45). To see this, note that if

$$\begin{bmatrix} -M\tilde{R}M^t & -M\tilde{P}M^t \\ I & 0 \end{bmatrix} \begin{bmatrix} I_m \\ \tilde{I}_m \end{bmatrix} = 0 \tag{6.49}$$

then by the second row, $I_m = 0$. Thus from the first row, $\mathcal{R}$ is nonsingular only if $M\tilde{P}M^t$ is also. However, $M\tilde{P}M^t$ contains the singular $\tilde{P}$. The singularity of $\tilde{P}$ does not

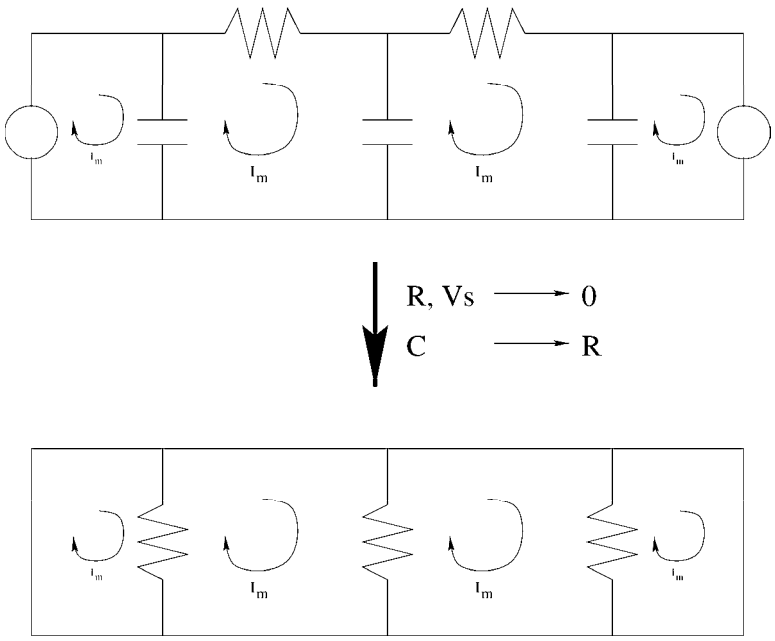


Figure 6-15: PEEC model circuit for only $M\tilde{P}M^t$

necessarily imply the singularity of $M\tilde{P}M^t$ and one may ask whether the particular circuit form of M may make $M\tilde{P}M^t$ nonsingular. To show singularity persists by an example, note that if we simplify the problem by letting P be diagonal, then each of the panels is a capacitor to ground (point a infinity), with $C_i = 1/P_{ii}$ and solving $M\tilde{P}M^t I_m = 0$ is solving a circuit as in Figure 6-6 with panels replaced by resistors and all other circuit elements replaced by shorts. If all meshes without panels are removed so that no row in $M\tilde{P}M^t$ is all zeros, we arrive at a circuit such as shown in Figure 6-15. This circuit could represent the discretization of a single straight wire with a source at both ends.

If all the capacitors are the same, $C_i = C$, then

$$M\tilde{P}M^t = \frac{1}{C}\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{6.50}$$

which admits a nonzero solution, $I_{m_i} = K$, where K is any constant, and thus $M\tilde{P}M^t$ is singular. Physically, with no sources present, the actual current through any resistor must be zero in the bottom circuit of Figure 6-15, and $I_{m_i} = K$ is a still a solution since the actual current through any resistor is zero.

6.2.2 A nonsingular $\mathcal{R}$

The singularity of $M\tilde{P}M^t$ implies that the $\tilde{I}_m$ are not linearly independent. Thus, to derive a nonsingular $\mathcal{R}$, a different $\tilde{I}_m$ must be chosen which are linearly independent. Note that the original realization had state variables

$$\begin{bmatrix} I_m \\ \tilde{I}_m \end{bmatrix} = \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ \tilde{I}_m^f \\ \tilde{I}_m^s \\ \tilde{I}_m^p \end{bmatrix} \tag{6.51}$$

but the state equations involving I_m^f in (6.41) are already first order since they do not involve panels. One could write a state-space system in which entries of I_m are only included in $\tilde{I}_m$ if they involve entries with $\frac{1}{s}$. This corresponds to

$$\begin{bmatrix} I_m \\ \tilde{I}_m \end{bmatrix} = \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ \tilde{I}_m^s \\ \tilde{I}_m^p \end{bmatrix} \tag{6.52}$$

since I_m^f meshes do not involve panels. Unfortunately, the example circuit of Figure 6-15 had no I_m^f meshes, but was still singular.

Since it seems the singularity of $\tilde{P}$ causes difficulty, consider choosing the new state variables as V_b^p instead of $\tilde{I}_m$. The relations will involve P alone,

$$sV_b^p = PI_b^p = P[M_{ps}^T \; M_p^T] \begin{bmatrix} I_m^s \\ I_m^p \end{bmatrix}. \tag{6.53}$$

Additionally, write $Z = R + sL$ of (6.24) as

$$\begin{bmatrix} Z_i & Z_{is} \\ Z_{is}^T & Z_s \end{bmatrix} \begin{bmatrix} I_b^i \\ I_b^s \end{bmatrix} = \begin{bmatrix} V_b^i \\ V_b^s \end{bmatrix} \tag{6.54}$$

where the superscript and subscript i represents filaments only used in meshes on the interior (M_f), and s represents filaments which are also used in meshes with panels (M_{fs}).

Using (6.39), (6.38), and (6.53) in (6.40) gives a first order system,

$$\begin{bmatrix} Z_{mf} & Z_{mfs} & 0 & 0 \\ Z_{mfs}^T & Z_{ms} & 0 & M_{ps} \\ 0 & 0 & 0 & M_p \\ 0 & -PM_{ps}^T & -PM_p^T & sI \end{bmatrix} \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ V_b^p \end{bmatrix} = \begin{bmatrix} V_s^f \\ V_s^s \\ V_s^p \\ 0 \end{bmatrix} \tag{6.55}$$

where $Z_{mf} = M_f Z_f M_f^T$, $Z_{mfs} = M_f Z_{fs} M_{fs}^T$, $Z_{ms} = M_{fs} Z_s M_{fs}^T$, $Z_{mf} = R_{mf} + j\omega L_{mf}$, and similarly for Z_{mfs} and Z_{ms}.

From (5.2), the terminal currents and voltages are related to their corresponding mesh quantities by $V_s = NV_t$, $I_t = N^T I_m$. Letting $B^T = [N^T \; 0]$ and separating out terms multiplying s gives the desired state-space form:

$$s \begin{bmatrix} L_{mf} & L_{mfs} & 0 & 0 \\ L_{mfs}^T & L_{ms} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ V_b^p \end{bmatrix} = \begin{bmatrix} -R_{mf} & -R_{mfs} & 0 & 0 \\ -R_{mfs}^T & -R_{ms} & 0 & -M_{ps} \\ 0 & 0 & 0 & -M_p \\ 0 & PM_{ps}^T & PM_p^T & 0 \end{bmatrix} \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ V_b^p \end{bmatrix} + \begin{bmatrix} V_s^f \\ V_s^s \\ V_s^p \\ 0 \end{bmatrix}$$

$$\begin{aligned} s\mathcal{L}\, x &= -\mathcal{R}\, x + B\, V_t \\ I_t &= B^T x. \end{aligned} \tag{6.56}$$

The $\mathcal{R}$ above can be shown to be nonsingular under the condition that no node is connected to the circuit via only panels. In circuit language, there can be no cut-sets of capacitors. Unfortunately, the point at infinity is a such a node. As an example, consider a loop of conductor similar to the line of conductor from Figure 6-15. The two ends of the loop will be connected to a source as shown in Figure 6-16. $\mathcal{R}$ will be nonsingular if and only if the DC solution ($s = 0$) to the PEEC circuit is unique. At $s = 0$, $s\mathcal{L} = 0$, and letting P be diagonal for simplicity, the PEEC circuit is given in Figure 6-16 with the point at infinity drawn at the center. If the source is zero, then the DC solution is that of $\mathcal{R}x = 0$, where $x^T = [I_m^T \; V_b^{pT}]$. Even if $I_m = 0$, the voltages at nodes A, B, C, and D specified by V_b^p can be set to an arbitrary constant (with the point at infinity 0) and $\mathcal{R}x = 0$ since the sum of voltages around any loop is still zero. Thus $\mathcal{R}$ is singular.

More formally, the potential described above is that of a Laplace's equation with pure Neumann boundary conditions and is thus unique only to a constant. To see this, at $s = 0$, (6.17) is

$$\boldsymbol{n} \cdot \boldsymbol{J} = 0 \tag{6.57}$$

and (6.8) with (6.13) becomes

$$\frac{\boldsymbol{J}}{\sigma} = -\nabla\Phi \tag{6.58}$$

Looking at the normal component of (6.58),

$$\frac{1}{\sigma}\boldsymbol{n} \cdot \boldsymbol{J} = -\frac{\partial\Phi}{\partial n} \tag{6.59}$$

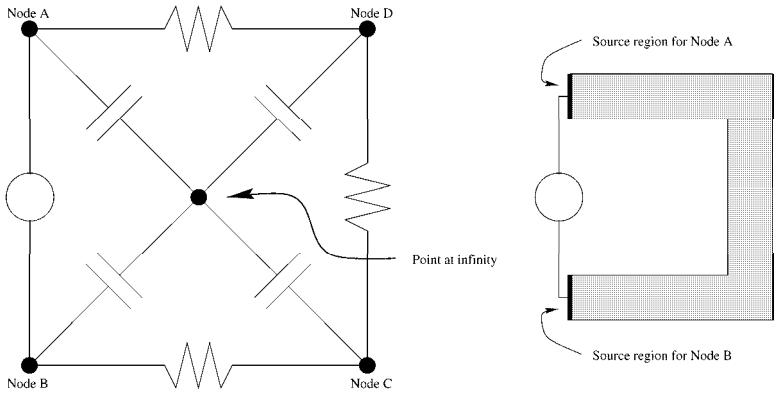


Figure 6-16: A loop of conductor and a coarse representative circuit

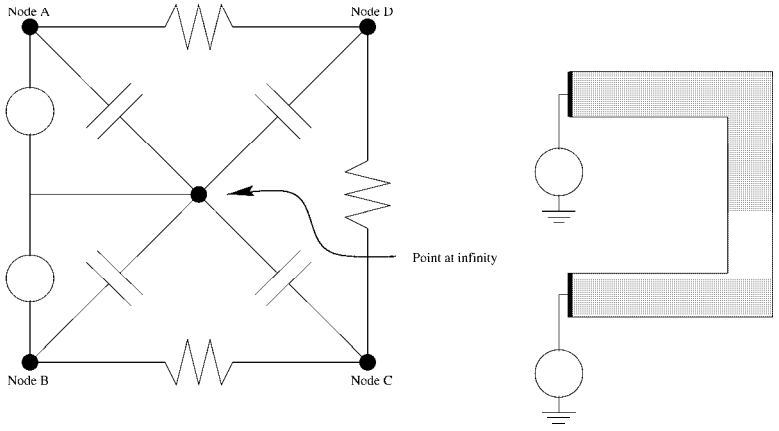


Figure 6-17: The new circuit with proper sources applied

and thus $\frac{\partial \Phi}{\partial n} = 0$ on $S - S_{terminals}$. It is well known that if $\frac{\partial \Phi}{\partial n}$ were specified on all of S, then the solution for Φ is unique only up to a constant. But if Φ is specified on any portion of the boundary, then the solution to Φ is unique. But what is done in Figure 6-16 is to specify a value for the *difference* $\Phi_A - \Phi_B$, and not Φ itself. Thus if Φ is a solution, then so is $\Phi + K$ for any constant K.

To make the solution unique, for each conductor of the geometry, Φ must be specified over some terminal. Such a condition is enforced in circuit terms by using voltage sources to the point at infinity as shown in Figure 6-17. In the figure, the old source is replaced by two sources connected to the point at infinity.

6.2.3 A Realization for Practical and Passive Model Order Reduction

With the nonsingular $\mathcal{R}$ of (6.56), the matrix-vector products of the form $\mathcal{R}^{-1}\mathcal{L}x$ needed for model order reduction are now possible. $\mathcal{R}$, however, contains the dense matrix P which implies that for each matrix-vector product, computing $\mathcal{R}^{-1}y$ requires a dense matrix solve. Additionally, $\mathcal{R}+\mathcal{R}^T$ is not necessarily positive semidefinite as required for passive model order reduction. It is straightforward to modify the system for passive model order reduction as shown next.

Consider multiplying the last row of (6.56) by $\boldsymbol{P}^{-1}$ to give

$$
\begin{aligned}
s\begin{bmatrix} L_{mf} & L_{mfs} & 0 & 0 \\ L_{mfs}^T & L_{ms} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & P^{-1} \end{bmatrix}\begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ V_b^p \end{bmatrix} &= \begin{bmatrix} -R_{mf} & -R_{mfs} & 0 & 0 \\ -R_{mfs}^T & -R_{ms} & 0 & -M_{ps} \\ 0 & 0 & 0 & -M_p \\ 0 & M_{ps}^T & M_p^T & 0 \end{bmatrix}\begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \\ V_b^p \end{bmatrix} + \begin{bmatrix} V_s^f \\ V_s^s \\ V_s^p \\ 0 \end{bmatrix} \\
s\mathcal{L}\,x &= -\mathcal{R}\,x + B\,V_t \\
I_t &= B^T x.
\end{aligned}
\tag{6.60}
$$

Note that B is left unchanged due to its lower block of zeros. To show that $\mathcal{L}+\mathcal{L}^T$ and

$$
\mathcal{R}+\mathcal{R}^T = 2\begin{bmatrix} R_{mf} & R_{mfs} & 0 & 0 \\ R_{mfs}^T & R_{ms} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{6.61}
$$

are both positive semidefinite, note that the top left 2×2 block of $\mathcal{L}$ and $\mathcal{R}$ are of the form WFW^T where F is either R or L from Section 6.1 and $W^T=[M_f^T\ M_{fs}^T]$. Since R and L are both positive definite, so is WFW^T. If P is generated via a Galerkin approach, then it too is positive definite, and since $\mathcal{L}$ is a block diagonal matrix consisting of blocks which are each positive semidefinite, then so is $\mathcal{L}+\mathcal{L}^T$.

Note that the equations of the third row of (6.60) represent *algebraic* relations. These relations are from the meshes I_m^p shown in Figure 6-6. These meshes explicitly enforce that the voltage on the two panels of each mesh are exactly equal. Since these conditions are algebraic, they lead to zero rows in $\mathcal{L}$. It is possible to eliminate these excess state variables and the algebraic conditions by following, for instance, tree-link analysis as described in [39]. Unfortunately application of such a technique would require the use of P^{-1} multiple times, which, since P is large and dense, is computationally inefficient. In addition, such analysis would destroy the structure of $\mathcal{R}$, which is necessary for passive model order reduction as will be described shortly.

Note that the block structure of (6.60) is similar to the nodal form, (6.35) and at this point in the development the advantages of pursing a mesh form are not apparent. Even though both are different realizations of the same system, a nodal form may have benefits over a mesh form. One advantage of (6.35) is that if there are no internal nodes, then the corresponding $\mathcal{L}$ of (6.35) is nonsingular and smaller than the mesh formulated $\mathcal{L}$. The nonsingular $\mathcal{L}$ could be used for expansions about $s = \infty$. The results pursuing model order reduction for (6.35) will not be described here, but the interested reader is referred to [44].

Since R is a diagonal matrix, and M is sparse, then $\mathcal{R}$ is now sparse. However, to form the first block of $\mathcal{L}$ requires order f^2 operations and memory since $L \in \mathbb{R}^{f \times f}$ is dense. Similarly, to form the second block, P^{-1} requires p^2 operations and memory to form P, and then order p^3 operations to invert. For complex geometries with tens of thousands of filaments and panels, such growth rates are severely limiting. In the next section we discuss a more efficient technique for generating reduced order models from (6.60).

6.3 Model Reduction for the Electromagnetoquasistatic Problem

In this section we describe a fast algorithm for computing a reduced order model about the expansion point $s = 0$ for the system in (6.60). To review, the idea of model order reduction is to reduce (6.60), which can be on the order of tens of thousands, to a much smaller system which still captures the dominant behavior of the original system. For moment matching techniques, one wishes to derive a rational function whose moments, or terms in the Taylor series expansion, match that of the original admittance function, $\boldsymbol{Y}_t(s)$, up to some order. From (6.60), the admittance function can be expanded about $s = 0$ as

$$\boldsymbol{Y}_t(s) = \boldsymbol{B}^T \left(\boldsymbol{\mathcal{R}} + s\boldsymbol{\mathcal{L}}\right)^{-1} \boldsymbol{B} = \sum_{k=0}^{\infty} \boldsymbol{m}_k s^k, \tag{6.62}$$

where the moments are obtained from

$$\boldsymbol{m}_k = -\boldsymbol{B}^T (\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{L}})^k \boldsymbol{\mathcal{R}}^{-1} \boldsymbol{B}.$$

Thus we seek an approximation, $\tilde{\boldsymbol{Y}}_t(s) = \sum_{k=0}^{\infty} \tilde{\boldsymbol{m}}_k s^k$, such that $\boldsymbol{m}_k = \tilde{\boldsymbol{m}}_k, k = 1, \ldots, q$. Since $\boldsymbol{Y}_t(s)$ represents a passive circuit, we require $\tilde{\boldsymbol{Y}}_t(s)$ also be passive,

Algorithm 6.3.1 (Iterative Scheme for $\boldsymbol{Pq} = \boldsymbol{v}$).

guess $\boldsymbol{q}^0$
Initialize the search direction
  $\boldsymbol{w}^0 = \boldsymbol{v} - \boldsymbol{P}\boldsymbol{q}^0$
for $k = 1, \ldots$ {
  Select $\boldsymbol{w}^k \in span\{\boldsymbol{w}^0, \boldsymbol{P}\boldsymbol{w}^0, \ldots, \boldsymbol{P}^{k-1}\boldsymbol{w}^0\}$
    such that the new solution
      $\boldsymbol{q}^k = \boldsymbol{q}^{k-1} + \boldsymbol{w}^k$
    minimizes $||\boldsymbol{r}^k|| = ||\boldsymbol{v} - \boldsymbol{P}\boldsymbol{q}^k||$
  if $||\boldsymbol{r}^k|| < tolerance$, return solution $\boldsymbol{q}^k$
}

which can be guaranteed via the numerically stable Arnoldi-based PRIMA model order reduction algorithm [50] described in Section 5.7.2.

6.3.1 Recycled iterative solver

Application of any moment matching scheme about $s = 0$ requires the computation of repeated matrix-vector products with the matrix $(\boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{L}})$ in order to obtain a reduced-order model. For instance, an Arnoldi type algorithm requires $q - 1$ such products to produce an order q model. However, because the partial inductance matrix $\boldsymbol{L}$ and potential coefficient matrix, $\boldsymbol{P}^{-1}$, which appear in $\boldsymbol{\mathcal{L}}$, are both large and dense, many multiplications by $\boldsymbol{\mathcal{L}}$ can be prohibitively expensive. In particular, if done directly, multiplication by $\boldsymbol{P}^{-1}$, would require an initial dense matrix factorization which is $O(p^3)$ operations. For modern packaging structures, for which p exceeds ten thousand, such a factorization is prohibitive.

The expensive factorization can be avoided by noting that the computation $\boldsymbol{q} = \boldsymbol{P}^{-1}\boldsymbol{v}$ is equivalent to solving for the panel charges, $\boldsymbol{q}$, given a set of voltages, $\boldsymbol{v}$. It is thus possible to use a preconditioned, Krylov-subspace iterative method to solve $\boldsymbol{Pq} = \boldsymbol{v}$ as outlined in Algorithm 6.3.1 [59]. Note that the dominant cost of each iteration is the $O(p^2)$ computation of a dense matrix-vector product, $\boldsymbol{Pw}$, to acquire the next vector in the subspace.

In the standard approach, for every product $\boldsymbol{\mathcal{L}x}$, the iterative algorithm would be called to solve $\boldsymbol{P}^{-1}\boldsymbol{v}$, generating a new subspace, $span\{\boldsymbol{w}^0, \boldsymbol{P}\boldsymbol{w}^0, \boldsymbol{P}^2\boldsymbol{w}^0, \ldots\}$, and a new set of search direction, $\boldsymbol{w}_k$. If the number of $\boldsymbol{\mathcal{L}x}$ products is large, the advantage of an

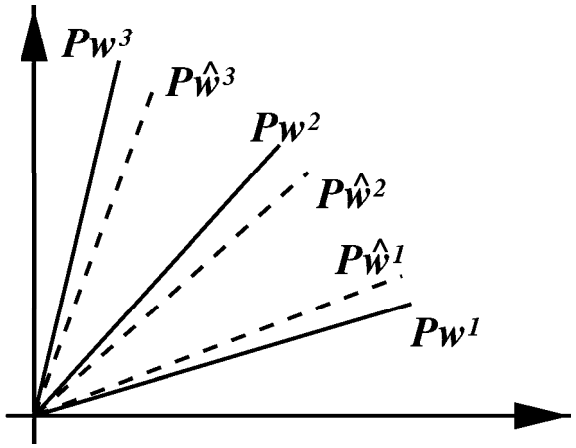


Figure 6-18: Two dimensional illustration of the search direction space for two different calls to the iterative algorithm. Here, the search directions $\boldsymbol{P}\boldsymbol{w}^k$ and $\boldsymbol{P}\hat{\boldsymbol{w}}^k$ are close so the spaces they span are similar.

iterative method would be degraded by the large number of total $\boldsymbol{P}\boldsymbol{w}$ products necessary. However, even though $\boldsymbol{w}^0$ is different for each solve, it may be that the space spanned by $\{\boldsymbol{w}^0, \boldsymbol{P}\boldsymbol{w}^0, \boldsymbol{P}^2\boldsymbol{w}^0, \ldots\}$ is similar, as is illustrated in Figure 6-18. One is thus lead to consider reusing the search directions from the previous solves [64, 52]. While the recycled vectors are not optimal for the next $\boldsymbol{v}$, the cost of computing the solution along those directions is negligible compared to a single $\boldsymbol{P}\boldsymbol{w}$ product. The recycled algorithm using the Krylov-subspace method known as Generalized Conjugate Residual (GCR) [59] is shown in Algorithm 6.3.2.[1]

The $O(p^2)$ operations of the iterative algorithm can be reduced further by using a multipole-accelerated iterative algorithm [47] whose cost and memory has been shown to grow only as $O(p)$. Similarly, the computation of the product $\boldsymbol{M}_f\boldsymbol{L}\boldsymbol{M}_f^t$ can be performed in $O(f)$ operations also via the multipole-algorithm [35].

6.3.2 First Results

In this section we present first results from model order reduction. [2]

Consider generating a reduced order models of order 10, 20, 40, and 80, for the matched 2D transmission line of Figure 6-7. From Figure 6-19 and 6-20, as the model order is increased, a model which matches to higher frequency is generated. For a reduced order

[1] Note that using a recycled version of the GMRES algorithm would have avoided the need to store both sets of $\boldsymbol{w}$ and $\boldsymbol{pw}$ but was not implemented here.

[2] Note that, due to its simplicity of implementation, our implementation uses a collocation approach rather than a Galerkin approach to compute the entries of P. Such an approach is not guaranteed to give a positive semidefinite $P + P^T$ but has yet to cause nonpassive models.

Algorithm 6.3.2 (Recycled GCR Algorithm for $\boldsymbol{P}\boldsymbol{q} = \boldsymbol{v}$).

GCRrecycled(input $\boldsymbol{P}, \boldsymbol{v}, \{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^p\}, \{\boldsymbol{pw}^1, \boldsymbol{pw}^2, \ldots, \boldsymbol{pw}^p\}, p,$ tol;
output $\boldsymbol{q}, \{\boldsymbol{w}^1, \boldsymbol{w}^2, \ldots, \boldsymbol{w}^m\}, \{\boldsymbol{pw}^1, \boldsymbol{pw}^2, \ldots, \boldsymbol{pw}^m\}, m)$ {

$\boldsymbol{r} = \boldsymbol{v}$ /* Initial error */
$k = 0$
Repeat until $||\boldsymbol{r}|| < tol$ {
$k = k + 1$
if (k $\leq$ p)
/* Use previous direction */
$\boldsymbol{w} = \boldsymbol{w}^k$
$\boldsymbol{pw} = \boldsymbol{pw}^k$
else
/* Compute new direction */
$\boldsymbol{w} = \boldsymbol{r}$
$\boldsymbol{pw} = \boldsymbol{P}\boldsymbol{w}$ /* $\leftarrow$ Do dense matrix-vector product */
for $j = 1, \ldots, k$ {
/* Orthogonalize (H is Hermitian transpose) */
$\beta = \frac{(\boldsymbol{pw}^j)^H \boldsymbol{pw}}{||\boldsymbol{pw}^j||}$
$\boldsymbol{pw} = \boldsymbol{pw} - \beta \boldsymbol{pw}^j$
$\boldsymbol{w} = \boldsymbol{w} - \beta \boldsymbol{w}^j$
$\boldsymbol{w}^k = \boldsymbol{w}$
$\boldsymbol{pw}^k = \boldsymbol{pw}$
}
end

/* compute new solution and error */
$\alpha = \frac{(\boldsymbol{pw})^H \boldsymbol{r}}{||\boldsymbol{pw}||}$
$\boldsymbol{q} = \boldsymbol{q} + \alpha \boldsymbol{w}$
$\boldsymbol{r} = \boldsymbol{r} - \alpha \boldsymbol{pw}$
}
/* set number of vectors to return */
if (k > p)
$m = k$
else
$m = p$
end
}

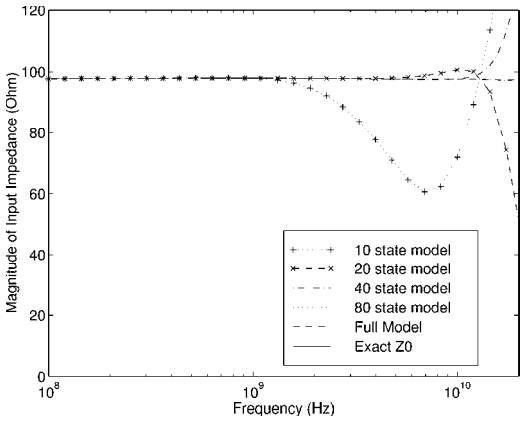


Figure 6-19: Impedance looking into a matched transmission line for various reduced order models and the original discretized system

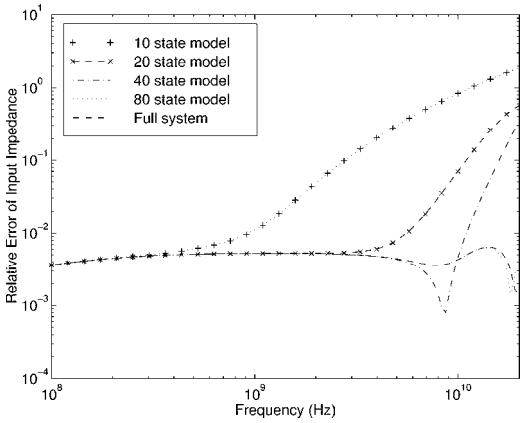


Figure 6-20: Relative error for models for matched transmission line

model with 1 percent error, a 20th order model is valid up to 6 GHz, a 40th order model up to 12 GHz, and an 80th order model past 20GHz (to about 26GHz).

Next, to demonstrate the efficiency of the recycled iterative scheme, consider refining the discretization of the transmission line of the previous example and extracting a 50th order model. Figure 6-21 shows the number of floating point operations (flops) required for direct factorization with back substitution, a non-recycled Krylov-subspace method, a recycled Krylov method, and multipole-accelerated recycled Krylov method, for various levels of discretization. Our implementation, called FASTPEP, uses direct matrix-vector products and thus the multipole-accelerated times are projected based on flop counts from multipole-accelerated capacitance and inductance codes [47, 35]. The residual error tolerance of the iterative algorithm had to be chosen as 10^{-6} so that the difference between models produced by the iterative scheme versus direct factorization differed by less than

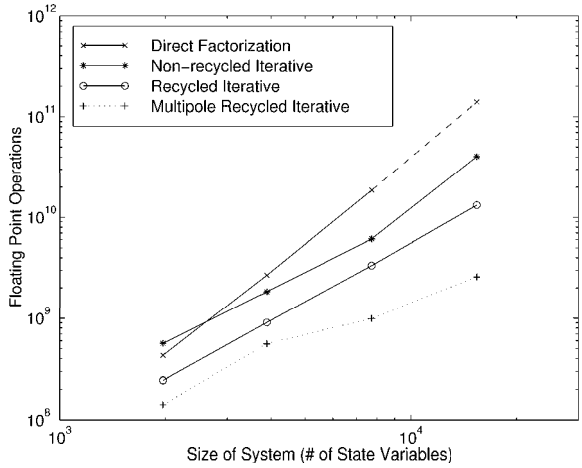


Figure 6-21: Flop count for different methods of computing $\boldsymbol{P}^{-1}\boldsymbol{x}$.

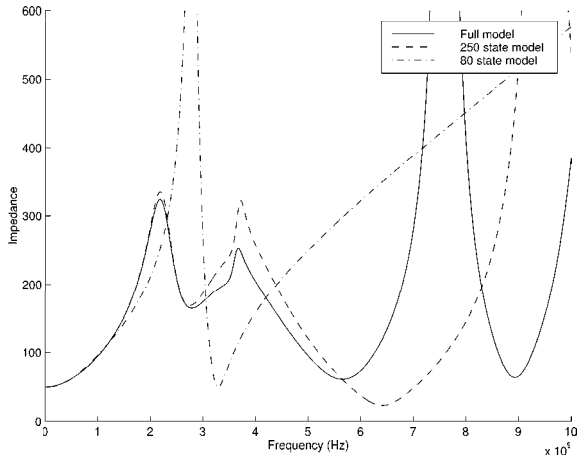


Figure 6-22: Various reduced order models for the connector

1% up to 100 GHz. As can be seen from the figure for an original 15409 state system, the recycled scheme performs an order of magnitude faster than direct factorization, and similarly, the multipole algorithm would provide another order of magnitude speed up. Note that the CPU time comparison would be similar to the flop count comparison for the direct factorization and direct recycled iterative scheme, however the overhead in arranging the multipole computation would shift its curve slightly upward.

As the next example, we generate reduced order models for the 3D connector of Figure 6-9 with slightly different terminating conditions. The results of generating models with 80 and 250 states are compared to the exact response in Figure 6-22. Note the strong improvement from 80 to 250 states.

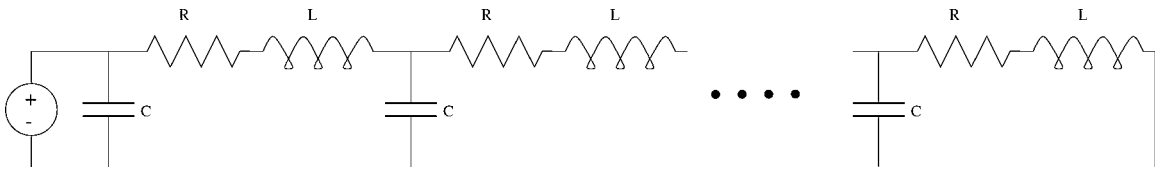


Figure 6-23: A ladder circuit to model 2D TEM transmission line

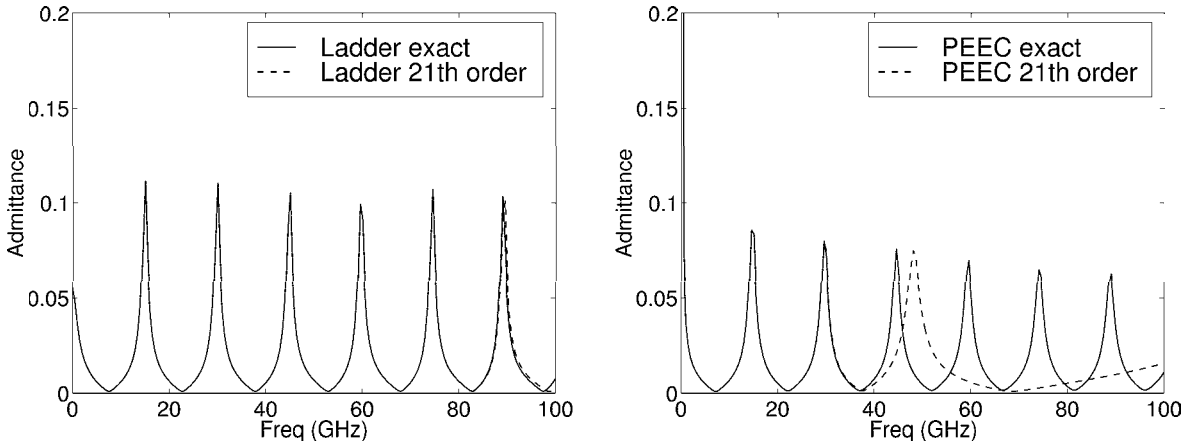
6.3.3 Quality of the Models

The methods of the last section provide efficient generation of a reduced order model, but the quality of the model, with or without acceleration techniques, is discouraging. For instance, there only seem to be a few resonances in the full model of Figure 6-22 yet a 250 state model was required to match the response past just the first resonance.

To investigate this problem, consider the two dimensional line of length 1cm from Figure 6-7. The line is divided into 40 sections along its length. Each section has a 9 filament bundle, and each node has 12 panels leading to a 1704 element circuit. The line is shorted at the far end instead of being matched in order to emphasize the resonances. The admittance is then computed through a number of the resonant frequencies of the line using both the full model and a 21st order reduced model.

For comparison, given the exact per unit length 2D line parameters, L' and C', a similar 40 section ladder circuit is constructed as shown in Figure 6-23. The resistance per unit length is chosen to roughly match the actual resistance at the first resonance (and thus not at DC!). The admittance is computed for both this 80th order model and also a 21st order reduced model. The four admittance functions are shown in Figure 6-24. The solid lines in each figure represent the full PEEC model and the full ladder model. The solid lines show qualitatively the same results: there is a periodic resonance with the first occuring when the 1 cm structure is half a wavelength long. The resonant peaks show a decay for the PEEC model since it captures skin effects. Similarly, one might hope that because the frequency behavior of the two is roughly the same, the model order reduction results for similar order would be the same. However, the 21st order reduced models are very different. The PEEC 21st order model loses accuracy before the third resonance, however the ladder model does not begin to degrade until the sixth.

To understand this phenomenon, Figures 6-25-a and 6-25-b plot the poles of the admittance function for the four cases. In Figure 6-25-a, the poles of the exact admittance lie evenly spaced on a vertical line in the s plane. Since the model order reduction was performed about $s = 0$, one would expect a trend of pole matching starting at the origin and moving outward as shown. In Figure 6-25-b, the full PEEC model has two sets



(a) (b)

Figure 6-24: Responses of various models for the transmission line. a) Exact and reduced ladder circuit. b) Exact and reduced PEEC.

of vertically spaced poles. The complex conjugate pairs with real part in $[-10^9, -10^8]$ are exactly pole-zero cancelled and do not affect the frequency response perhaps due to symmetry in the two conductor geometry. The conjugate pairs with real part near -5×10^{-10} correspond to the vertical line of poles in Figure 6-25-a and are the dominant poles of the system, responsible for the resonances in Figure 6-24-a. Note that they do not lie along a vertical line due to skin effects. In addition, there are also a large number of purely real poles. By noting the scaling of the plot, these real poles are the closest to the origin (see closeup in Figure 6-26). For this reason, moment matching about the origin tends to capture these poles first instead of the poles responsible for the resonant behavior. These real poles result from the discretization of the conductor into bundles of filaments. Since these filaments are mostly in the interior of the conductor, it is difficult to excite these modes from the conductor terminals and so the the effect on the frequency response of this large cluster of poles near the origin is weak; in fact all of these poles are nearly or exactly cancelled by zeros and thus do not have a strong effect on the admittance. In addition, as the order is raised beyond 21, most additional poles are matched near the origin resulting in very slow convergence to the full model.

While the general behavior of moment-matching model order reduction algorithm along the $j\omega$ axis is not as well understood as methods such as Truncated Balanced Realizations [45], we can gain some insight into trends involved in the process by returning to the origins of the Arnoldi algorithm in eigenvalue computation.

To relate the eigenvalues to the poles of the transfer function, note that the poles are the values of s which make $(\boldsymbol{\mathcal{R}} + s\boldsymbol{\mathcal{L}})$ singular in (6.62). Similarly, the eigenvalues of the

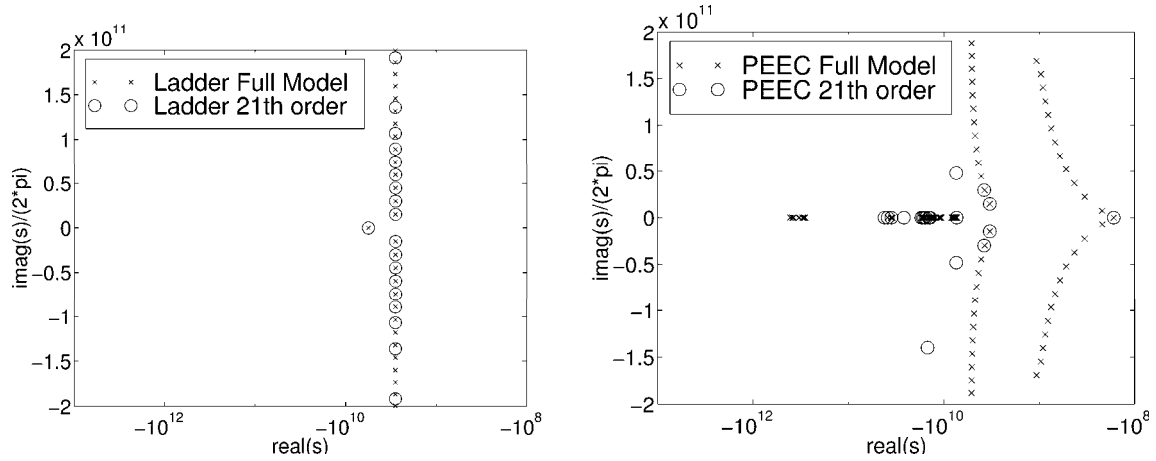


Figure 6-25: Pole locations for a) RLC ladder model and b) the PEEC models. Note axis scaling.

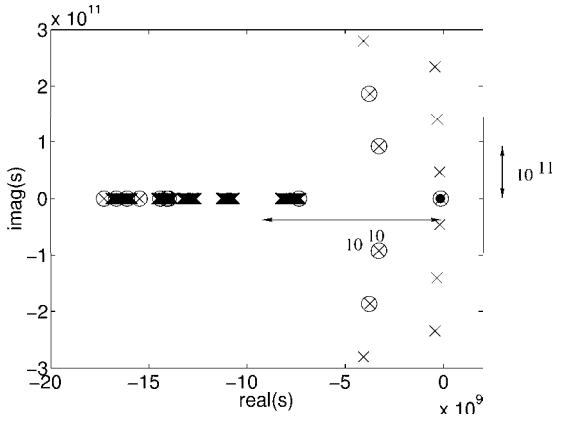


Figure 6-26: Closeup of origin on linear scale for the pole locations of the PEEC models. Solid black circle marks origin.

matrix are precisely those values of λ which make $(\lambda I - \boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{L}})$ singular. Thus $s = \frac{1}{\lambda}$.

The Arnoldi method's origin is in that of eigenvalue decomposition. The eigenvalues of the Arnoldi $\boldsymbol{H}_q$ matrix provide estimates for the eigenvalues of $\boldsymbol{A} = \boldsymbol{\mathcal{R}}^{-1}\boldsymbol{\mathcal{L}}$, and the orthogonal $\boldsymbol{V}_q$ provide for a projection of A into this space, $\boldsymbol{V}_q^T \boldsymbol{A} \boldsymbol{V}_q = \boldsymbol{H}_q$. It is know that Krylov subspace methods such as Arnoldi, produce an $\boldsymbol{H}_q$ which converges to *outlying* eigenvalues, i.e. well separated eigenvalues, more rapidly than those in a cluster [65]. From the eigenvalue plot of Figure 6-27 corresponding to the reciprocal of the values in Figure 6-25, we see that poles nearest the origin correspond to eigenvalues farthest from the origin and well separated (note the x-scale is in log units). Thus, the model order reduction problem above will have an accurate estimate for those poles first,

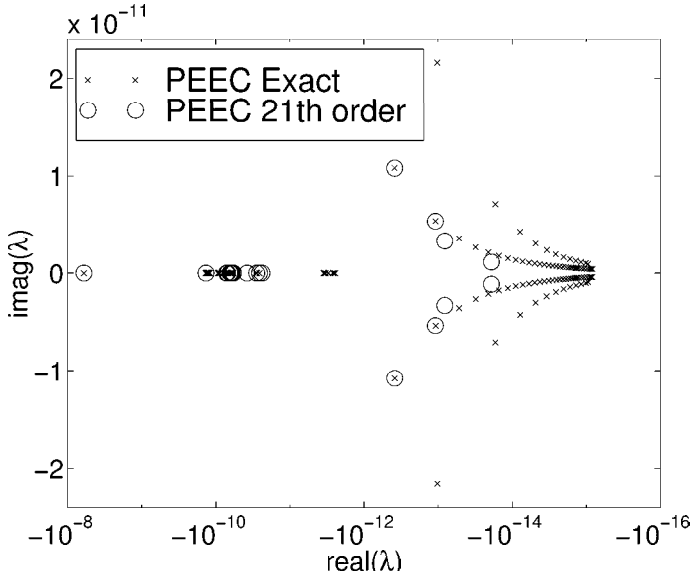


Figure 6-27: The eigenvalues (1/s) of the PEEC model. Note axis scaling

and converge toward the eigenvalues close to the origin last, as seen by the two conjugate pairs of eigenvalues with real part near 10^{-13} and 10^{-14} of the 21^{st} order model which have not converged to poles of the full model. Thus the poles nearest the origin are captured accurately first.

Left out of the above discussion is the influence of the input vectors, $\boldsymbol{B}$. While $\boldsymbol{A}$ by itself determines the poles of the system, $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ determine the position of the *zeros* of the transfer function. As we shall describe in a moment, a pole should not appear in the reduced order model if $\boldsymbol{B}$ indicates that it is exactly or nearly cancelled by a zero. But as discussed before, many of the poles along the real axis are canceled to machine precision or have very small residues compared to the poles along the $j\omega$ axis. Yet, as observed in Figure 6-25, these weak or zeroed poles appear in the model before those with a larger component in $\boldsymbol{B}$.

What we see is that while the influence of $\boldsymbol{B}$ may be important for the first few iterations, its influence is quickly diminished by roundoff errors. To illustrate this, first note that a pole which is exactly cancelled by a zero should theoretically never appear in the reduced order model because a cancelled mode is not in one of either the controllability or observability spaces defined by $\mathcal{K}_n(\boldsymbol{A}, B) = span\{B, \boldsymbol{A}B, \boldsymbol{A}^2B, \ldots, \boldsymbol{A}^nB\}$ and $\mathcal{K}_n((\boldsymbol{A}^T), C) = span\{C, (\boldsymbol{A}^T)C, (\boldsymbol{A}^T)^2C, \ldots, (\boldsymbol{A}^T)^nC\}$, respectively [30]. In the above example, the modes turn out to be in neither space and thus one would expect that cancelled poles could not appear in the Arnoldi model which is based on a Krylov subspace $\mathcal{K}_k(\boldsymbol{A}, B) = span\{B, \boldsymbol{A}B, \boldsymbol{A}^2B, \ldots, \boldsymbol{A}^kB\} \subset \mathcal{K}_n(\boldsymbol{A}, B)$.

However, because of finite precision arithmetic, directions are introduced into B corresponding to cancelled modes. To understand this, let $A = \mathcal{R}^{-1}\mathcal{L}$ and write its eigendecomposition as

$$AX = X\Lambda \tag{6.63}$$

where $X = [x_1 x_2 ...]$, with each x_i as an eigenvector, and Λ is the diagonal matrix of eigenvalues. For illustration, assume A is a normal matrix such that X can be chosen as an orthonormal matrix and that the eigenvalues are distinct and are ordered $\lambda_1 > \lambda_2 > \ldots$.

Let $\boldsymbol{B} = x_1$, so that λ_1 is the only uncanceled mode. Then new directions are introduced in finite precision arithmetic after a matrix-vector product: $Ab = \lambda_1 \boldsymbol{x}_1 + \epsilon \boldsymbol{y}$ where $\boldsymbol{y}$ is a random vector and ϵ is on the order of machine precision. The next vector of the Arnoldi iteration, $p^1 = Ab - \lambda b = \epsilon \boldsymbol{y}$ is no longer 0 and contains components of cancelled modes.

Next, cancelled modes will appear before real ones since at each iteration, the erroneous component of x_i will be magnified by λ_i. For instance, if B contains no x_2 component, $\boldsymbol{B} = x_1 + x_3 + x_4 + \cdots$ then at the iteration, k, at which $(\epsilon\lambda_2)^k > (\lambda_j)^k$, the cancelled mode (λ_2) will have a larger component in p^k than x_j.

6.3.4 Improving the model

The cluster of real, weak poles near the origin is not limited to the transmission line example and has been observed in most examples. We thus seek a geometry independent means of avoiding the matching of the weak poles near the origin. One thought is the expansion about $s = \infty$ which would select those poles last and model generation would involve $\mathcal{L}^{-1}\mathcal{R}$. But for most discretizations, there are multiple panels at the nodes, and thus $\mathcal{L}$ is singular. The nodal formulation approach could be used for such expansions as pursued in [44].

Another approach is to use some multipoint scheme [8, 16]. The idea of such a scheme is to match moments corresponding to multiple values of s rather than just $s = 0$. However for any finite expansion point, matrix solution with $(\mathcal{R} + s_0\mathcal{L})$ is required at *each* iteration, which is computationally expensive since $\mathcal{L}$ has dense blocks. To reduce the computational cost of this solution, fast techniques of iterative solution are developed in the next section.

6.4 Efficient Multipoint Approximation

As observed in the last section, moment matching about $s = 0$ generates particularly poor results due to the large cluster of poles near the expansion point. The next course is to consider expanding about some other point or points. Such multipoint expansions have been explored previously for explicit moment matching in [8], and for the Krylov subspace techniques in [16, 22]. Recently, a provably passive multipoint rational Arnoldi algorithm has been derived for the reduction of RLC circuits with multiple inputs and outputs [11].

For expansions about points $s_0 \neq 0$, the moments become

$$\boldsymbol{m}_k = -\boldsymbol{B}^T \left\{ (\boldsymbol{\mathcal{R}} + \boldsymbol{s}_\circ \boldsymbol{\mathcal{L}})^{-1} \boldsymbol{\mathcal{L}} \right\}^k \boldsymbol{\mathcal{R}}^{-1} \boldsymbol{B}.$$

Thus, to apply any multipoint scheme for the large dense systems of (6.60), one must be able to compute not only $\boldsymbol{\mathcal{L}}x$ rapidly, as in the last section, but also

$$(\boldsymbol{\mathcal{R}} + \boldsymbol{s}_\circ \boldsymbol{\mathcal{L}})^{-1} \boldsymbol{y}. \tag{6.64}$$

Again, (6.64) is too large for direct factorization and one is led to iterative solution.

6.4.1 Iterative Solution

Iterative solution of

$$\boldsymbol{x} = (\boldsymbol{\mathcal{R}} + s_0 \boldsymbol{\mathcal{L}})^{-1} \boldsymbol{y} = \begin{bmatrix} R_{mf} + s_0 L_{mf} & R_{mfs} + s_0 L_{mfs} & 0 & 0 \\ R_{mfs}^T + s_0 L_{mfs}^T & R_{ms} + s_0 L_{ms} & 0 & M_{ps} \\ 0 & 0 & 0 & M_p \\ 0 & M_{ps}^T & M_p^T & s_0 P^{-1} \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{6.65}$$

is particularly expensive because each matrix vector product of an iterative algorithm requires an inner solve for P^{-1}. Thus, the model reduction algorithm would have 3 levels of nested loops: 1) the Arnoldi moment-matching iteration, 2) for each moment matched, an iterative solve of (6.64), and 3) at each iteration for (6.64), a solution with P^{-1}.

Fortunately, this last inner iteration can be avoided by realizing that computation of (6.64) is very close to solving the mesh formulated circuit of (6.60) at a single frequency s_0 given an input vector $\boldsymbol{y}$. Thus we can return to solving the second order form, (6.40), to compute $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \end{bmatrix}$, and then compute V_b^p separately. However, (6.65) differs

from (6.60) since y_4 is not generally zero, yet the last entry in the input for (6.60) is always zero. Thus, to eliminate V_b^p, we use

$$sV_b^p = P[M_{ps}^T \; M_p^T] \begin{bmatrix} I_m^s \\ I_m^p \end{bmatrix} + Py_4. \tag{6.66}$$

in place of (6.53) and arrive at

$$MZ_{EM}M^t \begin{bmatrix} I_m^f \\ I_m^s \\ I_m^p \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 - \frac{1}{s_0} M_{ps} P y_4 \\ y_3 - \frac{1}{s_0} M_p P y_4 \end{bmatrix}. \tag{6.67}$$

After solving (6.67), $x_4 = V_b^p$ can be directly computed from (6.66). This approach avoids the inner computation of $P^{-1}x$ but more importantly, it returns us to the familiar mesh matrix, (6.40), to which we can apply the effective preconditioning techniques of Part I.

6.4.2 Preconditioning

To follow the approach of Section 3.3, consider preconditioning with a block diagonal version of Z_{EM}. Thus, the preconditioner will be a factored version of

$$M \begin{bmatrix} R + s_0\hat{L} & 0 \\ 0 & \hat{P}/s_0 \end{bmatrix} M^t \tag{6.68}$$

where $\hat{L}$ and $\hat{P}$ are block diagonal. The results for a single solve for the backplane connector of Figure 2-1 discretized into 5112 panels and 2592 filaments are shown in Figure 6-28.

The performance is good compared to no preconditioning, but one can improve this preconditioner by noting that for fast capacitance extraction in [47], it was found that block diagonal preconditioning for P is not adequate to capture the strong coupling involved in charge interaction. For that reason, in [47] the overlapped preconditioner, or local-inversion preconditioner as it was named in Section 3.3, was developed. Since we know this preconditioner works well for P, we wish to use it here in (6.68). Note that we are not pre and post multiplying P by M as in Section 3.3 since we want only to find an approximation to the inverse of P.

To apply the local inversion preconditioner in the mesh formulated preconditioner of (6.68), it must be positive definite as discussed in Section 3.3. While this preconditioner is not guaranteed to be positive definite, we have found from experiment that it produces good results implying it must be "close" to positive definite. This can be explained by

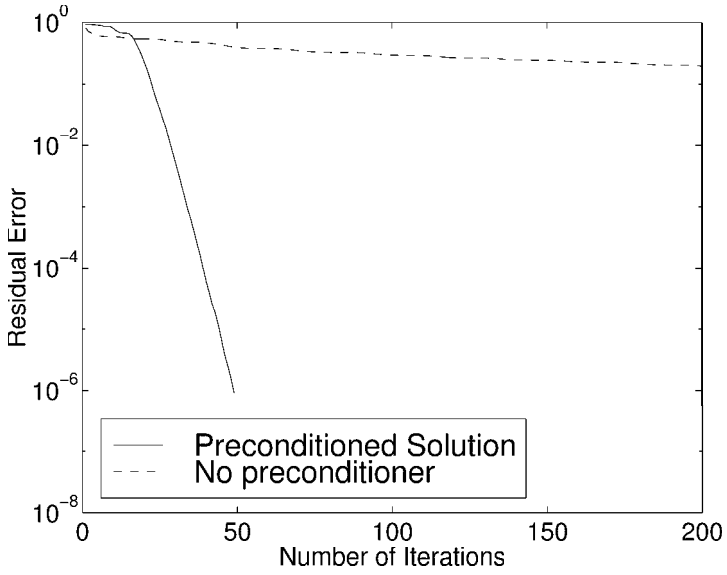


Figure 6-28: Convergence of iterative solver for one solve of $MZ_{EM}M^T$ using a block diagonal preconditioner.

realizing that each row of the preconditioner, C, comes from the inverse of a small P matrix. The P matrix is close to what is known as an "M" matrix whose inverse is known to be diagonally dominant [67]. Since each row of C is likely to be diagonally dominant, then C is likely to be positive definite.

The preconditioner of the form (6.68) requires an approximation for P, but C is an approximation of P^{-1}. Thus, (6.68) would become

$$M \begin{bmatrix} R + s_0 \hat{L} & 0 \\ 0 & \frac{1}{s_0} C^{-1} \end{bmatrix} M^t \tag{6.69}$$

requiring an *inversion* of C, destroying its sparsity. C^{-1} would be too dense to then factor efficiently as a submatrix of (6.68).

Fortunately this inversion can be avoided. The form of (6.40) was desirable because it avoided the inverse of P. Now, to avoid the inversion of C we can return to the first order form, (6.60), for the preconditioning step. More specifically, the preconditioning step requires computing $\boldsymbol{x}$,

$$\left(M \begin{bmatrix} R + s_0 \hat{L} & 0 \\ 0 & \frac{1}{s_0} C^{-1} \end{bmatrix} M^t \right)^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}. \tag{6.70}$$

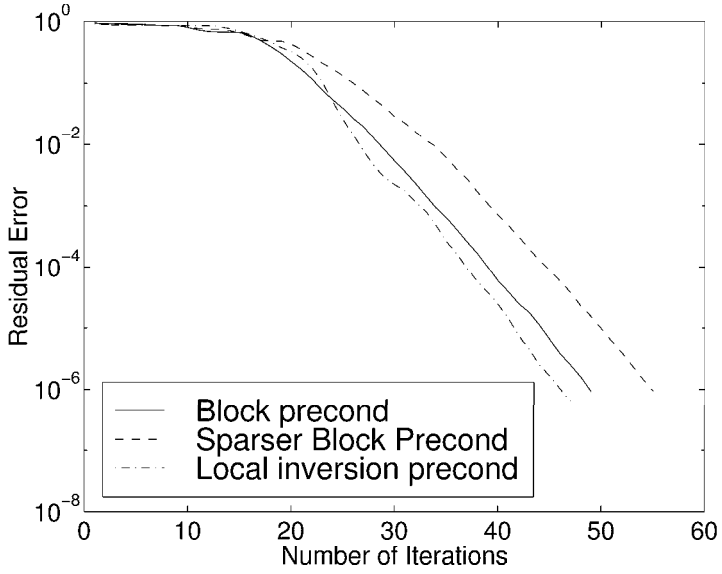


Figure 6-29: Convergence of iterative solver for one solve of $MZ_{EM}M^T$ using different preconditioners

Instead solve

$$\begin{bmatrix} \hat{R}_{mf} + s_0\hat{L}_{mf} & \hat{R}_{mfs} + s_0\hat{L}_{mfs} & 0 & 0 \\ \hat{R}^T_{mfs} + s_0\hat{L}^T_{mfs} & \hat{R}_{ms} + s_0\hat{L}_{ms} & 0 & M_{ps} \\ 0 & 0 & 0 & M_p \\ 0 & M^T_{ps} & M^T_p & s_0C \end{bmatrix}^{-1} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 0 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \tag{6.71}$$

and discard x_4. The matrix is sparse and can be computed rapidly with LU factorization.

With this new preconditioner, the iteration count is smaller as shown in Figure 6-29 and also the number of nonzeros is considerably less as shown in Table 6-1. The table and figure also include a block preconditioner with smaller blocks than in Figure 6-28. This smaller block preconditioner had roughly the same number of nonzeros in the unfactored matrix as the local inversion preconditioner, yet it was worse in all respects, as shown in the table.

The local inversion preconditioner required less time, fewer iterations and had many fewer nonzeros in the factored matrix. Since the CPU time to factor each of these is so small, choosing a denser preconditioner could significantly improve results. Such an optimization will not be pursed here. Note that for a denser preconditioner, the CPU time advantage of the local inversion preconditioner would become considerably more significant.

These results give considerable improvement, however the stagnation for first ten or so iterations requires further investigation. The shell preconditioning ideas of the

Preconditioner	Number of iterations	non-zeros before factor	non-zeros after factor	Factor time (CPU secs)
Block Diag 1	49	288622	894175	14.2
Block Diag 2	55	152974	675278	9.4
Local Inversion	47	143438	276561	4.5

Table 6-1: Nonzeros and CPU time for factorization of only the preconditioner. Local Inversion is for a block diagonal $\hat{L}$ and local inversion based C

Section 4.2 could be attempted here. Also, if multiple moments are to be matched at a given s_0, recycling can further improve results as we see in the next section.

6.4.3 Recycling

Just as for $s = 0$, if many moments are to be matched at a given s_0, then the Krylov subspace from previous moments can be reused. Consider now computing multiple moments at $s_0 = j * 5 \times 10^{11}$.

For each iterative solve, we see that the number of iterations decreases as shown in Figure 6-30. The 23 solves required a total of 422 matrix-vector products, compared to the roughly $23 * 47 = 1081$ that would be required without recycling, representing over a factor of two speedup. This speedup is counterbalanced by the memory consumption in storing the back vectors. For the above problem, storage of the dense $\boldsymbol{L}$ and $\boldsymbol{P}$ matrices with 5112 panels and 2592 filaments requires 263 MB. Storage of the back vectors with 6858 complex entries each requires 46 MB, which is over 17 percent of the total storage. For this small problem, such memory is acceptable, but for larger problems which require multipole or precorrected-FFT acceleration, such consumption is unacceptable. By noting that the most benefit from recycling comes from the first few solves in Figure 6-30, the storage of these back vectors could be stopped when memory consumption is a concern.

6.4.4 Recap

To recap, we noted that the dominant cost of applying Arnoldi-based model order reduction was in the repeated computation of $\boldsymbol{Ax}$ where, for a nonzero expansion point s_0, $\boldsymbol{A} = (\boldsymbol{\mathcal{R}} + \boldsymbol{s}_\circ \boldsymbol{\mathcal{L}})^{-1}\boldsymbol{\mathcal{L}}$. Since the submatrices of $\mathcal{L}$ are L and P^{-1} which are dense matrices of dimension 10^4 to 10^5 a practical algorithm must avoid $O(n^3)$ computation such as the explicit formation of P^{-1} or $(\boldsymbol{\mathcal{R}} + \boldsymbol{s}_\circ \boldsymbol{\mathcal{L}})^{-1}$. In Section 6.3.1 we used an iterative

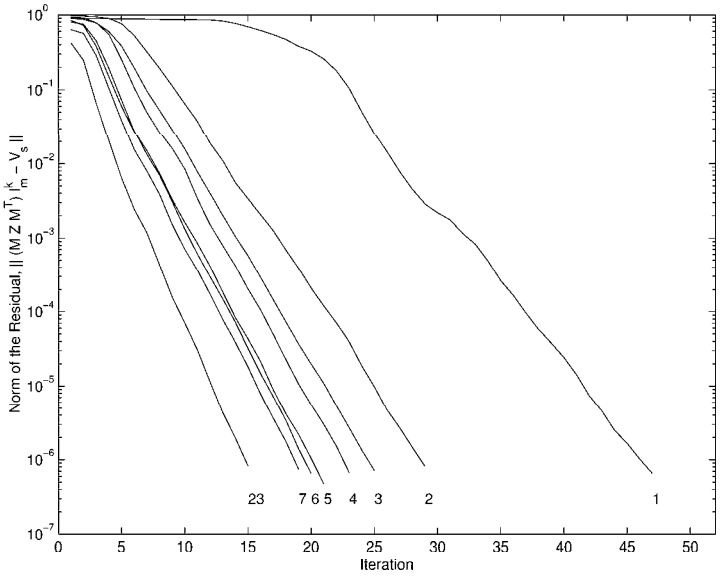


Figure 6-30: Convergence of iterative solver for $MZ_{EM}M^T$ using recycling. Numbers for each line correspond to solve number. With more vectors from recycling, the later solves converge faster.

algorithm to avoid forming P^{-1} in the computation of $\mathcal{L}\boldsymbol{x}$. Applying a preconditioned iterative algorithm was straightforward since it was identical to capacitance computation as explored in [47]. Since many solves must be performed, a recycling algorithm was employed for further speedup. To compute the $(\mathcal{R} + s_0\mathcal{L})^{-1}\boldsymbol{y}$ portion of the $\boldsymbol{Ax}$ product, iterative solution is not as straightforward. Since P^{-1} is contained within $(\mathcal{R} + s_0\mathcal{L})$, iterative solution would require two levels of nested solve. Fortunately, we can return to a pure mesh form for this computation and then the iterative solve is instead for $(MZ_{EM}M^t)^{-1}$. Not only does this form avoid the nested solve, but also provides a good method of preconditioning from the previous work of Part I. To apply these preconditioning ideas to include the added capacitive portion of Z_{EM}, we saw that the best preconditioning approach for capacitance required us to return to the $(\mathcal{R} + s_0\mathcal{L})$ block form for just the preconditioning step. All these pieces provided for iterative solution in fewer than 50 iterations and even fewer with recycling.

6.4.5 Results of Model Order Reduction about $s \neq 0$

With multipoint models now computationally tractable, we can begin to investigate generating low order models. In this section we give results of expansions about nonzero s to give insight into the properties of the resulting models specific to interconnect analysis. The general topic of multipoint model generation via Krylov subspace methods is

addressed in detail in [22].

Expansion points

In this section we explore choosing a single nonzero s as an expansion point and discuss the quality of the resultant reduced order models.

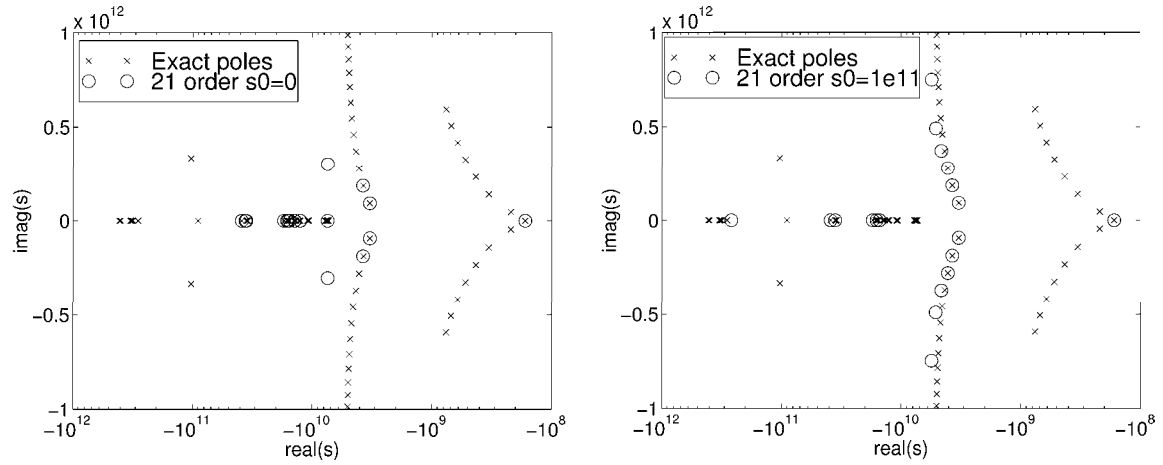
Consider choosing $s_0 \in \mathbb{R}$ and $s_0 > 0$. From Section 6.3.3 where $s_0 = 0$, a pole, s_i, is captured first when the value $\frac{1}{s_i}$ is well separated from the other poles. Thus poles close to the origin were captured first. For $s_0 \neq 0$, poles for which $\frac{1}{s_i - s_0}$ is well separated come first. Since all the poles have $Re(s) < 0$, no pole will be closer than the distance s_0. In particular, the weak poles at the origin appear more as a cluster and we would expect to capture the "separated" eigenvalues away from the origin more readily. Additionally, the choice of a real expansion point implies solution of (6.67) will not involve complex arithmetic.

The results for various s_0 for a PEEC model for a 2D TEM transmission line are shown in Figures 6-31 and 6-32. We see that for $s_0 = 0$, many poles are matched at the origin. $s_0 = 1e11$ does slightly better but still places multiple poles near the origin, $s_0 = 3e11$ does quite well, and $s_0 = 1e12$ does not seem to capture any single pole accurately. To explain $s_0 = 1e12$, the expansion point is too far away, and the entire region of poles appears as a distance cluster for which convergence to any one pole is slow. These results demonstrate that choosing an appropriate real expansion point is difficult.

Instead, consider choosing $s_0 \in \mathbb{C}$. In fact, choosing purely imaginary expansion points is the common choice since it is the response along the $j\omega$ axis which is of interest. As with choosing the origin, the poles nearest the expansion point will tend to appear in the model first. For model reduction for interconnect analysis, we can exploit that the only large dense cluster of poles is at the origin, and thus in choosing an imaginary expansion point away from the origin, the algorithm will not stagnate as for $s = 0$.

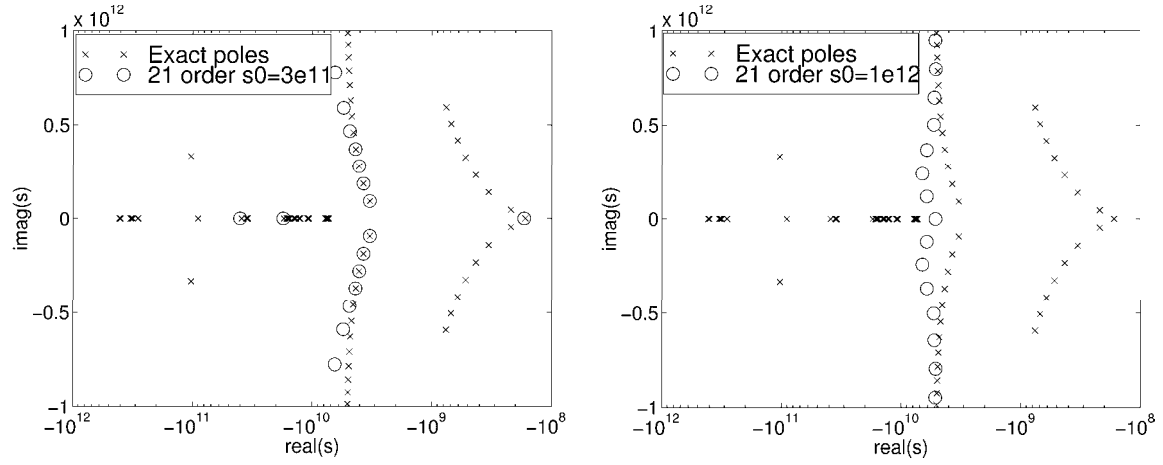
The advantage of complex s_0 comes at a cost. Solving (6.67) will involve complex arithmetic and is thus four times as expensive as the real case. However, to maintain a real reduced order model, moments must be matched at conjugate pair points, s_0 and s_0^*. It was pointed out in [57] that the $x = x_r + jx_i$ and $x^* = x_r - jx_i$ resulting from solving (6.67) at s_0 and s_0^*, respectively, generate only two independent vectors, x_r and x_i, for the Arnoldi algorithm. These directions can be computed with a single solve of (6.67) and thus the cost of a complex expansion point is only double that of a real one.

For the TEM example, we choose $s_0 = \pm j5 \times 10^{11}$ and in Figure 6-33 we see that model order reduction matches the dominant poles starting at $\pm j5 \times 10^{11}$ and moving outward



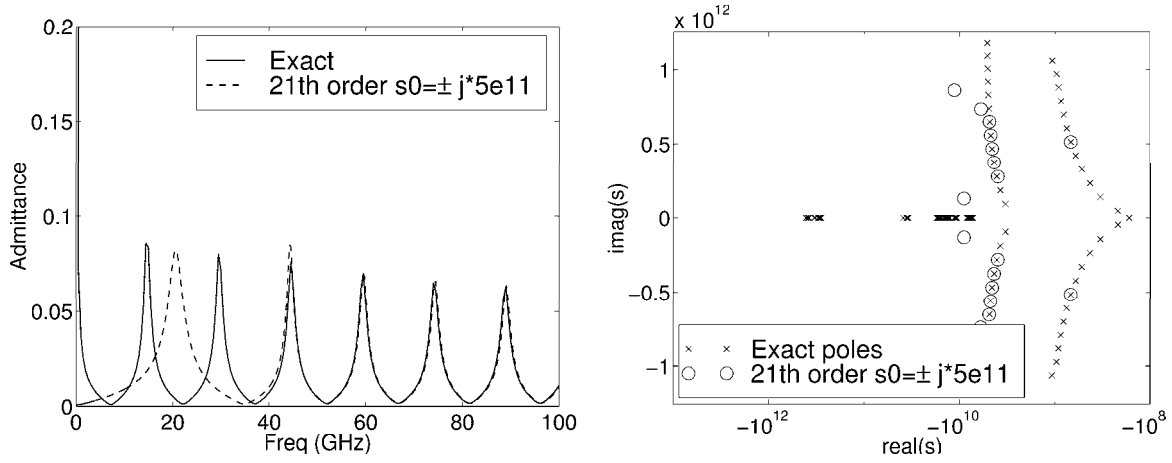
(a) (b)

Figure 6-31: Poles of reduced model for 2D two conductor TEM line for different real expansion points.



(a) (b)

Figure 6-32: Poles of reduced model for 2D two conductor TEM line for different real expansion points.



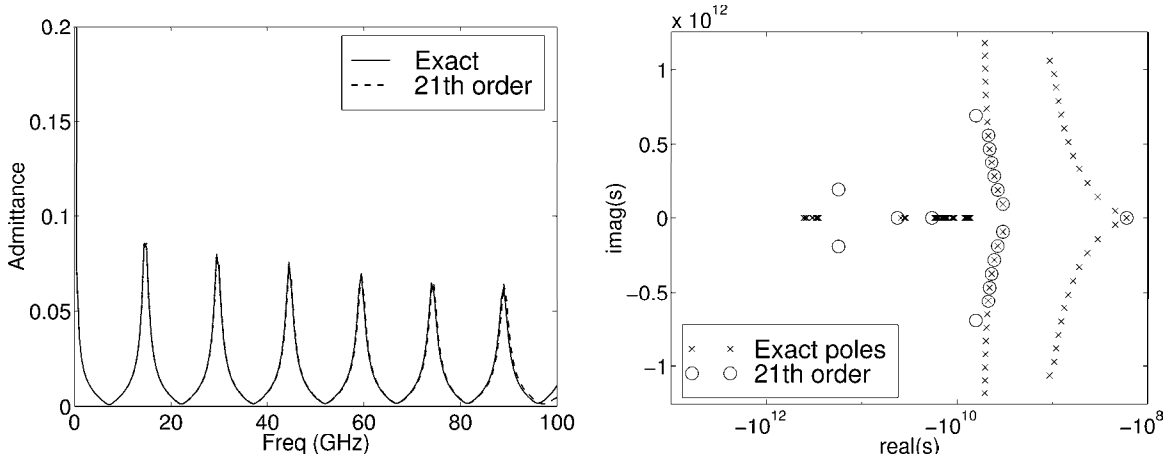
(a) (b)

Figure 6-33: Admittance and poles of reduced model for 2D two conductor TEM line for complex expansion point.

as desired. Note that almost every pole matched corresponds to a pole responsible for the resonant peaks.

To take advantage of the ability to do multipoint expansions, consider matching many moments at multiple points along the imaginary axis. Consider choosing 6 points, $s_0 = 0, \pm j1\times10^{11}, \pm j2\times10^{11}, \pm j3\times10^{11}, \pm j4\times10^{11}, \pm j5\times10^{11}$ and matching $1, 4, 4, 4, 4, 4$ moments at each, respectively. The results for this 21^{st} order model are shown in Figure 6-34 where we see that this approximation accurately captured the dominant poles to a frequency comparable to that for the RLC ladder network of Figure 6-25-a and 6-24-a.

While the results in Figure 6-34 were the underlying goal, the choice of expansion points and number of moments to match at each did not come without trial and error. For instance, consider an 11th order model matching 2 moments at each point instead of 4 in the previous example. The results in Figure 6-35 show that even though the model roughly captured the poles, it did not capture their magnitude well and thus did not give a very accurate frequency response. Comparing to Figure 6-34, perhaps the influence of *all* the weak poles near the origin has an effect at these higher frequency and more moments should be matched at zero. This is not the case as shown in Figure 6-36. We now match 5 moments at $s = 0$ instead of 1, and while the response for the first two resonances near $s = 0$ improved, near the third around 40 GHz, it has worsened. Also, the pole for the second resonance has moved away from its exact value but the magnitude of the resonance is close to the exact. Both these examples indicate that strictly observing



(a) (b)

Figure 6-34: Admittance and poles of reduced model for 2D two conductor TEM line for complex expansion points $s_0 = 0, \pm j1\times10^{11}, \pm j2\times10^{11}, \pm j3\times10^{11}, \pm j4\times10^{11}, \pm j5\times10^{11}$ and matching $1,4,4,4,4,4$ moments at each.

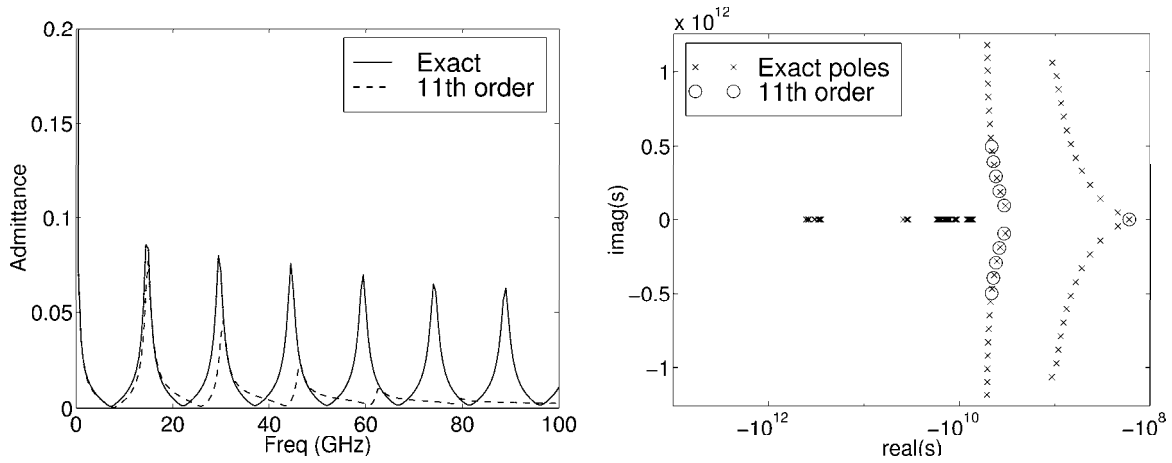
pole locations is not a direct measures of error in the frequency response.

These examples illustrate the need for sophisticated methods of error analysis and expansion point and order selection. For the provably passive multipoint Krylov-subspace Arnoldi schemes used here, some of ideas of Grimme [22] might work here.

The point of the above discussion was to describe the need for automatic multipoint methods, but such methods could not be applied unless the the algorithms of this chapter make problems for $n > 10^4$ computationally possible. Next we show that indeed they do.

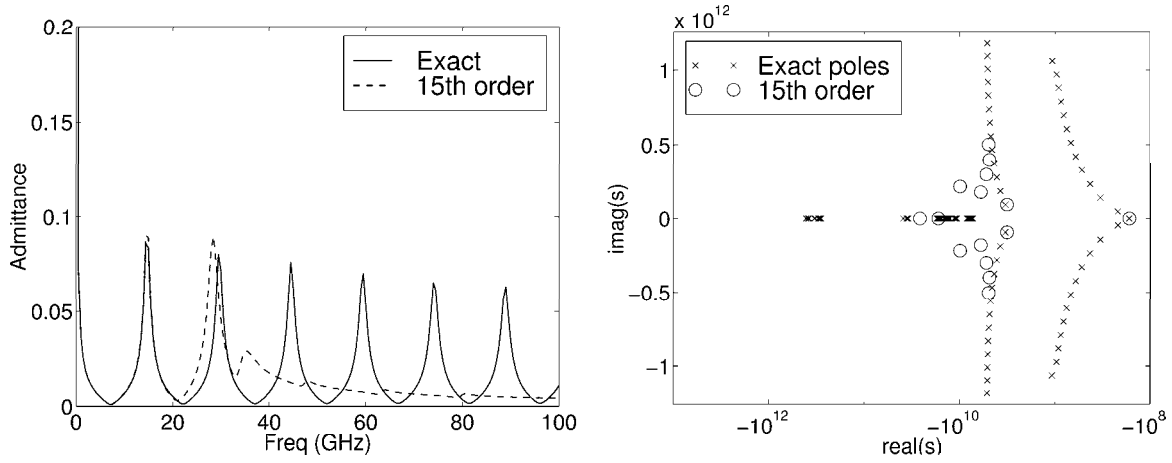
Results for a Practical Example

Finally, consider a real example to show the computational efficiency of multipoint expansions via the algorithms of this chapter. We generate a 50th order model for the half of the backplane connector of Figure 2-1. The discretization generated 1560 panels and 480 filaments. For simplicity, only one input and one output is modeled, $B = b$, corresponding to exciting only one of the middle pins. Assume we desire an accurate frequency response up to $\omega = 10^{11}$. Since poles tend to be matched outward from the expansion point, we match 48 moments about a midpoint $s_0 = j5 \times 10^{10}$ and then two at $s_0 = 0$ to insure accurate capture of the DC behavior. To match moments at multiple points, as in the previous section, we use the algorithm in [11]. The self-admittance of the excited pin is shown in Figure 6-37. The result is compared to a much higher order



(a) (b)

Figure 6-35: Admittance and poles of reduced model for 2D two conductor TEM line for complex expansion points $s_0 = 0, \pm j1\times10^{11}, \pm j2\times10^{11}, \pm j3\times10^{11}, \pm j4\times10^{11}, \pm j5\times10^{11}$ and matching $1, 2, 2, 2, 2, 2$ moments at each.



(a) (b)

Figure 6-36: Admittance and poles of reduced model for 2D two conductor TEM line for complex expansion points $s_0 = 0, \pm j1\times10^{11}, \pm j2\times10^{11}, \pm j3\times10^{11}, \pm j4\times10^{11}, \pm j5\times10^{11}$ and matching $5, 2, 2, 2, 2, 2$ moments at each.

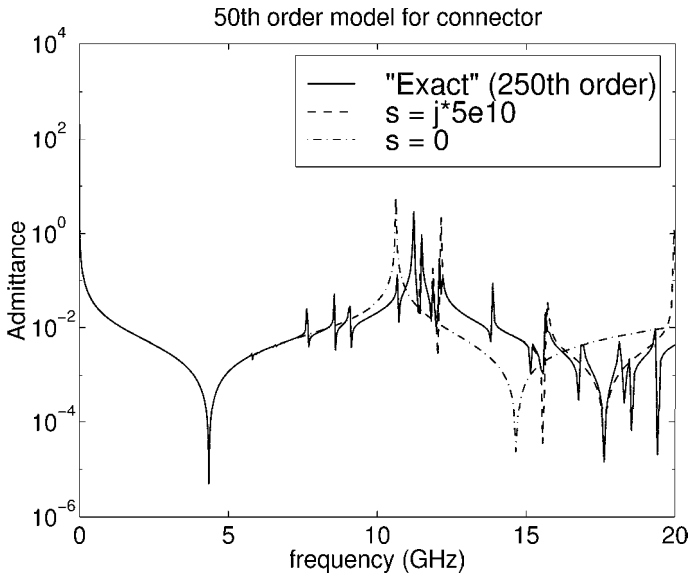


Figure 6-37: Various Reduced order models for the self-admittance of a middle pin from part of the backplane connector

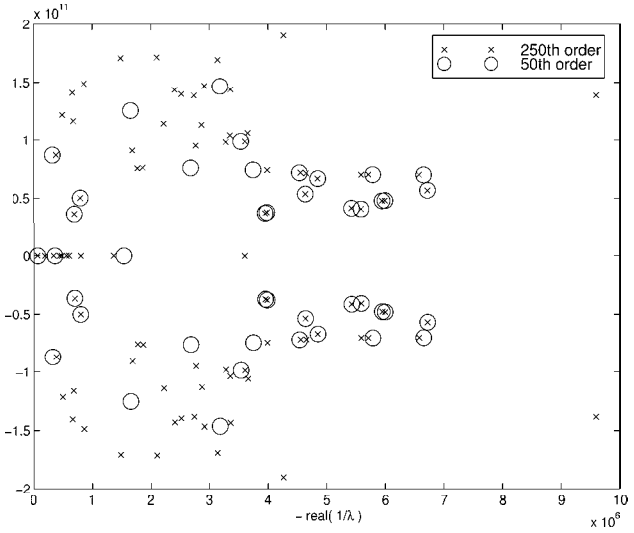


Figure 6-38: Poles of reduced model and higher order model for part of the backplane connector

model of size 250. The 50th order model matches well up to around 15 GHz, compared to the poor response of a 50th order model for which all moments were matched at $s = 0$. The poles captured in the reduced model are shown in Figure 6-38. The many real poles in the original model are greater than 10^7 and are out of range of the plot.

To observe the computational efficiency, Figure 6-39 shows the total CPU floating point operations (flops) required to generate a 50th order model for the *full* connector

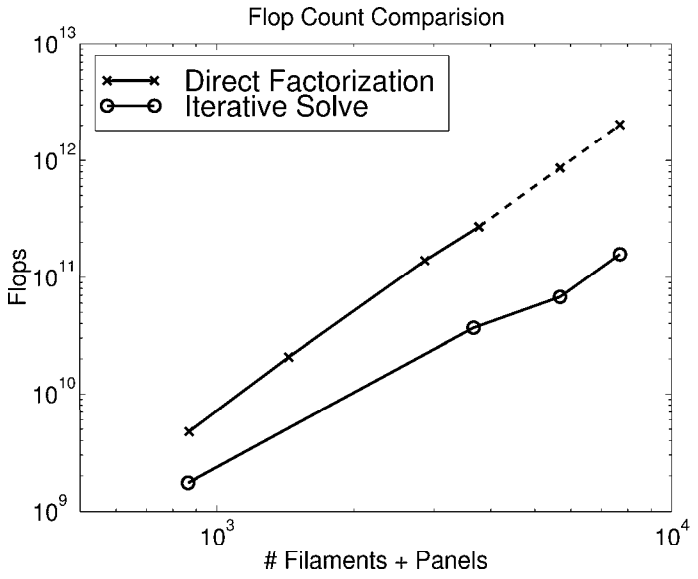


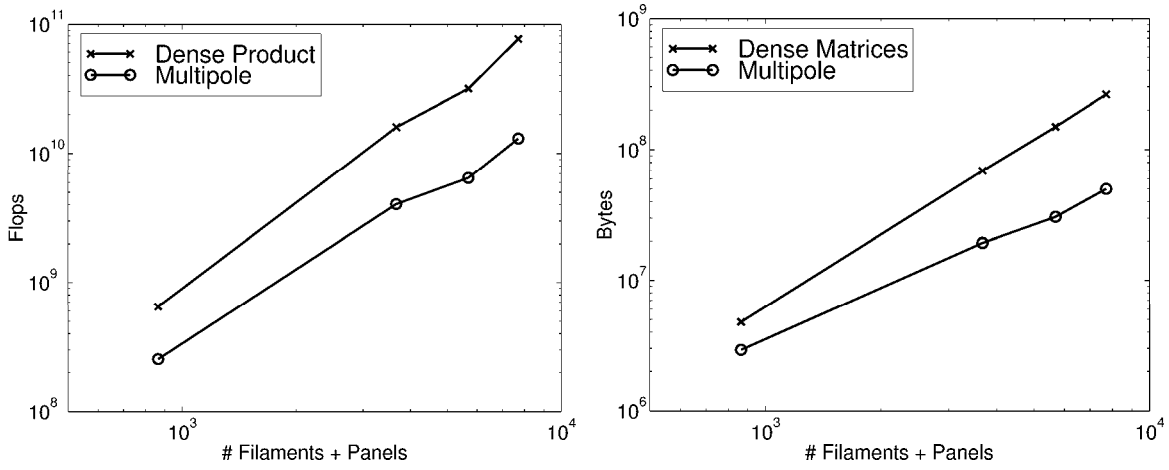
Figure 6-39: CPU flops required to generate a 50th order model.

for various levels of discretization. As can be seen, if (6.67) were solved by direct factorization, the flops would grow as $O(n^3)$, but with the iterative solver, the growth is only $O(n^2)$. Note that for a modest problem size still under 10^4 elements, the iterative algorithm is an order of magnitude faster than direct factorization.

With efficient iterative solution in place, the Multipole algorithm could be directly applied to bring the operation count and memory growth to $O(n)$. The benefits of such an approach are shown in Figure 6-40. We see that using multipole acceleration is roughly a factor of 5 improvement in both time and memory over dense matrix vector products.

6.5 Closing

In this chapter we developed a mesh formulated approach for passive model order reduction of the full quasistatic Maxwell's equations. We found that model reduction about $s = 0$ stagnates due to a common feature of the PEEC method which generates clusters of poles near the origin. The advantage of the mesh formulation became apparent for computing multipoint expansions because the first order state-space form could be reformulated in a pure mesh form for just the computationally intensive portion of the Arnoldi algorithm. From a pure mesh form, effective preconditioning and multipole acceleration could be applied to give an algorithm of nearly $O(n)$ flop and memory growth which, for $n \approx 10^4$, was 50 times faster and consumed 5 times less memory than direct factorization. Such growth rates make such algorithms essential as problems near



(a) (b)

Figure 6-40: Projections using a multipole algorithm for the dense matrix vector product for the 50th order connector model. a) Flop count for the total matrix vector product times. b) Memory consumption.

$n = 10^5$.

Issues remain for the choice of expansion points. Having the exact response and the exact position of the poles for the two conductor problem of Figure 6-34 gave guidance for choosing the s_0. Even with such information, the choice of moments to match at s_0 required some trial and error to derive a good model. But even worse, for real problems the exact information is not available. For $n > 10^4$ computing all the poles would be more expensive than the direct factorization we avoid. For this reason, methods of automatically choosing these points and expansion order are necessary.

7

Summary and Future Work

7.1 Summary

This thesis presented techniques for efficient parasitic extraction to model and simulate complicated three-dimensional interconnect structures. For complicated structures, the size of the linear systems that must be solved exceeds 10^4 and direct factorization approaches were impractical.

The focus was first in a regime for which accurately modeling the resistance and inductance was necessary. As reviewed in Chapter 3, computing the inductance at individual frequency points could be accelerated with a preconditioned iterative solver. When each iteration of the solver is computed with the Fast Multipole algorithm the overall algorithm is reduced to $O(n)$. While the growth rate provided orders of magnitude speedup over direct factorization approaches such as Gaussian Elimination, we saw in Chapter 4 that developing a nonuniform discretization for planar structures can reduce n itself by an order of magnitude.

The algorithms of Chapter 3 and 4 capture the frequency dependent resistance and inductance at specific frequency points only. However, often the end use of interconnect models is for simulation with nonlinear devices. Such simulation must be performed in the time domain and knowledge of the resistance and inductance at individual frequencies is not adequate. Chapter 5 discussed methods of efficiently generating compact yet accurate time domain descriptions of the interconnect. Explicitly computing the terms of the power series to compute a Padé approximant is an ill-conditioned approach which limits the accuracy of the models beyond a very low order. Instead, the Arnoldi approach was developed to generate numerically robust models. Since a q^{th} order Arnoldi model does not match the optimal number of moments of the power series it is not a Padé

approximant. This loss of moments however, did not seem to impact the accuracy. Additionally, the freedom in not matching as many moments permitted the development in [50] of a variant of the Arnoldi algorithm which preserved passivity. The Arnoldi approach as a circuit reduction technique is applicable to not only the RL domain as in Part I but also the RLC domain of Chapter 6.

As the frequencies of interest for these interconnect structures approach or exceed the point where the structures are self-resonant, single lumped models are no longer valid and thus a distributed RLC approach was pursued in Chapter 6. The derivation of the distributed RLC model from Maxwell's equations followed a mesh formulation approach to exploit algorithms for efficient iterative solution. In order to apply Arnoldi-based model order reduction and also to capture the steady state behavior in simulation, care was taken to insure that the first order state-space form of the mesh formulation had a DC solution.

With a mesh formulated approach for passive model order reduction of the full quasistatic Maxwell's equations, we found that model reduction about $s = 0$ stagnates, that is, the reduced order transfer function along the $j\omega$ axis converges slowly to the exact transfer function. This behavior was attributed to the large clusters of poles near the origin, a common feature of the PEEC RLC method. A common solution to such a problem is to match moments about some $s \neq 0$. Computations for the Arnoldi algorithm for $s \neq 0$ however, would be extremely expensive due to the required nested dense iterative solve. The advantage of the mesh formulation then became apparent for computing expansions at $s \neq 0$ because the first order state-space form could be reformulated in a pure mesh form. From a pure mesh form, effective preconditioning and multipole acceleration could be applied to give an algorithm of nearly $O(n)$ flop and memory growth which, for $n \approx 10^4$, was 50 times faster and consumed 5 times less memory than direct factorization. This improvement in performance is modest since these growth rates will make such algorithms essential as problems near $n = 10^5$.

7.2 Future Work

Chapter 6 derived a computationally efficient method of *automatically* generating passive reduced order models for complicated three-dimensional interconnect under the quasistatic assumption. We saw that an accurate low order model could be generated for the backplane connector. However the method is not fully automatic in regard to error control. An accurate model was generated only after comparing the response of the reduced model to the exact model. Unfortunately, the exact response is generally not

available for comparison. It is thus not clear how to fully automate the generation of a good model.

Methods to investigate include the Complex Frequency Hopping [8] approach which provides a heuristic for multipoint model generation, and more recently the work of [22] develops error *estimates* along the $j\omega$ axis which can be used with a multipoint passive block Krylov-method in [11].

But instead of pursing a multipoint approach, perhaps a reformulation of the integral equation can avoid the many weak poles near the origin which stalled the expansion about $s = 0$. For instance, the many weak poles can be connected to the many interior filaments required to capture skin effects. Perhaps a surface integral formulation similar to that of [66] would not have such poles and expansions about $s = 0$ would be sufficient.

Even with a surface formulation for which we can match all moments at $s = 0$, when have enough moments been matched? That is, how can we say precisely when the error is small. Methods involving truncated balanced realizations [45] have long been used in the system theory field for model order reduction. However, such an approach requires the $O(n^3)$ solution of the Lyapunov equation which limits the use of such an approach. Recently, Krylov subspace techniques have been applied to solving the Lyapunov equations for balanced truncation [28] with limited success and perhaps warrant further investigation.

Finally, all the methods of this thesis rely on the quasistatic assumption. Model reduction approaches which use the full Maxwell's equations (full-wave) are much more computationally expensive as investigated in [52]. There is thus a strong need to determine to what extent in frequency the quasistatic assumption is valid. In Appendix A it was roughly investigated in the frequency domain, however this comparison should be more carefully investigated in both the frequency domain and the time domain to determine when this large computational penalty of full-wave calculations must be taken.

A

Quasistatic Versus Full Wave Analysis

In this appendix we approximate the admittance for the full Maxwell's equations and compare it with quasistatic analysis to explore the limits of quasistatic analysis. We will analyze the two-conductor line of uniform cross section previously shown in Figure 6-7. We saw in Figure 6-8 that driving a 1 cm version of this line with a matched load produced good results up to 30 GHz where the structure is one wavelength long. Similarly in Figure 6-24 we saw that if the line is shorted instead of matched, quasistatic analysis captures the location of the resonances correctly.

In this section we will look at this phenomenon in more detail. The width and thickness of the two conductors is 37 μm and 15 μm, respectively, as before. The length is extended to $2cm$ so that at 30 GHz, the structure is two wavelengths long. The structure is terminated with a 10 Ω resistor. In the next experiments, we will vary the separation between the conductors.

To apply the quasistatic formulation of Chapter 6, the conductors are discretized into 200 sections along their length. Skin effect is not modeled and thus each section consists of one filament giving a total of 400 filaments. A similar approach is used for the panels except only one panel on each section is used instead of covering all four sides of the conductor. Even though such a discretization is poor for matching analytic results it is adequate for comparison to full wave as will be done next.

To compute full wave solutions, we no longer assume that the phase term, $e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|} \approx 1$ and must use (6.11) and (6.20) for the potentials. The L and P matrices which would result from this change we write as L^{full} and P^{full} and could be approximated as

$$L_{ij}^{full} = L_{ij} e^{\frac{s}{c} r_0}$$

where L_{ij} is the entry in the quasistatic partial inductance matrix from (6.26) and r_0 is the center to center distance between filaments i and j. The same approach could be used for P^{full}. This center to center approximation of the $e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|}$ term is poor for nearby filaments. Using this approximation, the comparison to the quasistatic solution was found to be poor even for low frequencies for which the quasistatic solution should compare well. For this reason the phase term is neglected for $r_0 \leq \frac{10^{-2}}{24}$,

$$L_{ij}^{full} = \begin{cases} L_{ij}e^{\frac{s}{c}r_0} & \text{if } r_0 > \frac{10^{-2}}{24} \\ L_{ij} & \text{if } r_0 \leq \frac{10^{-2}}{24} \end{cases}$$

Thus this experiment only observes the effect of the "far away" phase terms on the admittance. The length $\frac{10^{-2}}{24}$ corresponds to using a quasistatic approximation for filaments not more than three away along a conductor. Note that at 30 GHz, the neglected phase a distance $\frac{10^{-2}}{24}$ is

$$e^{j\frac{2\pi}{24}} = 0.96593 + 0.25882j$$

which corresponds to a 25 percent error and at 3 GHz

$$e^{j\frac{2\pi}{240}} = 0.99966 + 0.02618j$$

which is 2.6 percent in error.

Using this approximation, the admittance at discrete frequency points was computed for separations $0.01, 0.1, 1cm$ with the conductivity of copper. At this conductivity, the DC resistance of the conducting loop, without the 10 Ω termination, is 1.5Ω. The magnitude, phase, and error are shown in Figures A-1, A-2, and A-3. For comparison, the figures also show the solution for a truncated quasistatic solution where

$$L_{ij}^{trunc} = \begin{cases} 0 & \text{if } r_0 > \frac{10^{-2}}{4} \\ L_{ij} & \text{if } r_0 \leq \frac{10^{-2}}{4} \end{cases}$$

and similarly for P^{trunc}. The truncated solution is included to compare $e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|} \approx 1$ to $\frac{e^{s/c|\boldsymbol{r}-\boldsymbol{r}'|}}{|\boldsymbol{r}-\boldsymbol{r}'|} \approx 0$.

The results show that for small separation in Figure A-1, the quasistatic solution does well and the truncated quasistatic does not. In other words, the distant phase terms can be neglected, but not the entire L_{ij}. When the separation increases to $0.1cm$, quasistatics is superior to truncated and is roughly accurate to one percent until the structure is half a wavelength (7.5 GHz). As the frequency rises, the error is below 10 percent until the structure is a wavelength long (15 GHz). Note that the magnitude of the admittance appears accurate in Figure A-2, but the phase deviates above 15 GHz. Finally, for large

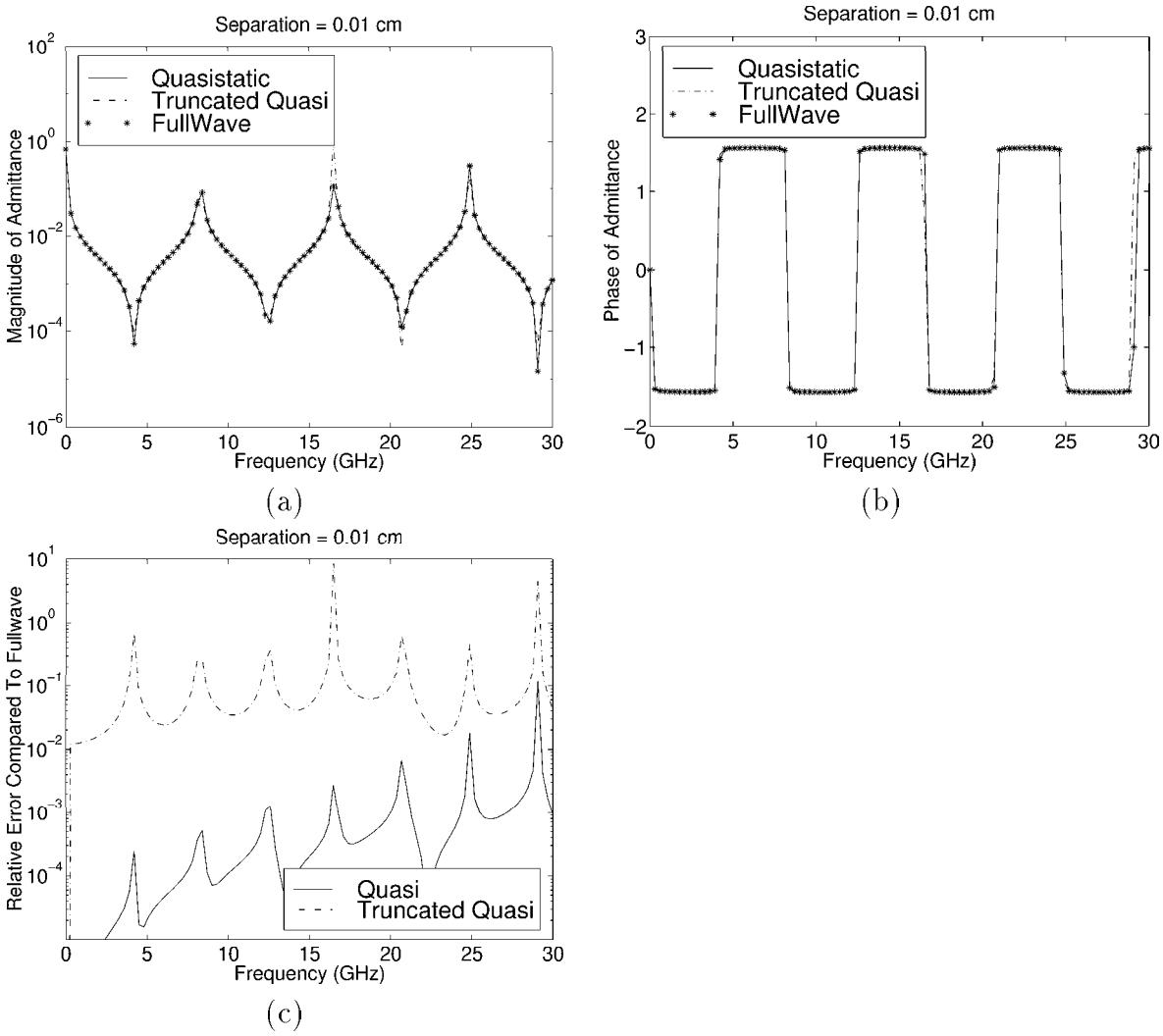


Figure A-1: Magnitude, phase, and error compared to full wave for a two conductor copper ($\sigma = 5.8 \times 10^7 (\Omega m)^{-1}$) transmission line, 1 cm long, separated by $0.01cm$.

separation in Figure A-3, the behavior is captured only qualitatively. The phase is quite different and the height of the resonances is overestimated.

Skin effect was not modeled, and thus the loss was underestimated. Since the maximum error occurs at the resonance points, we next observe if higher loss would damp the resonances and improve the comparison. The conductivity was decreased by two orders of magnitude. At this conductivity, the DC resistance of the conducting loop, without the 10 Ω termination, is 150Ω. The actual skin effected resistance at 30 GHz is 30Ω. Even with the higher loss, the results shown in Figures A-4, A-5, and A-6 show only improvement at the resonances and are otherwise similar as before.

In summary, these results suggest that if error up to a few percent can be tolerated, then quasistatics can be used to capture the admittance parameters for structures on the

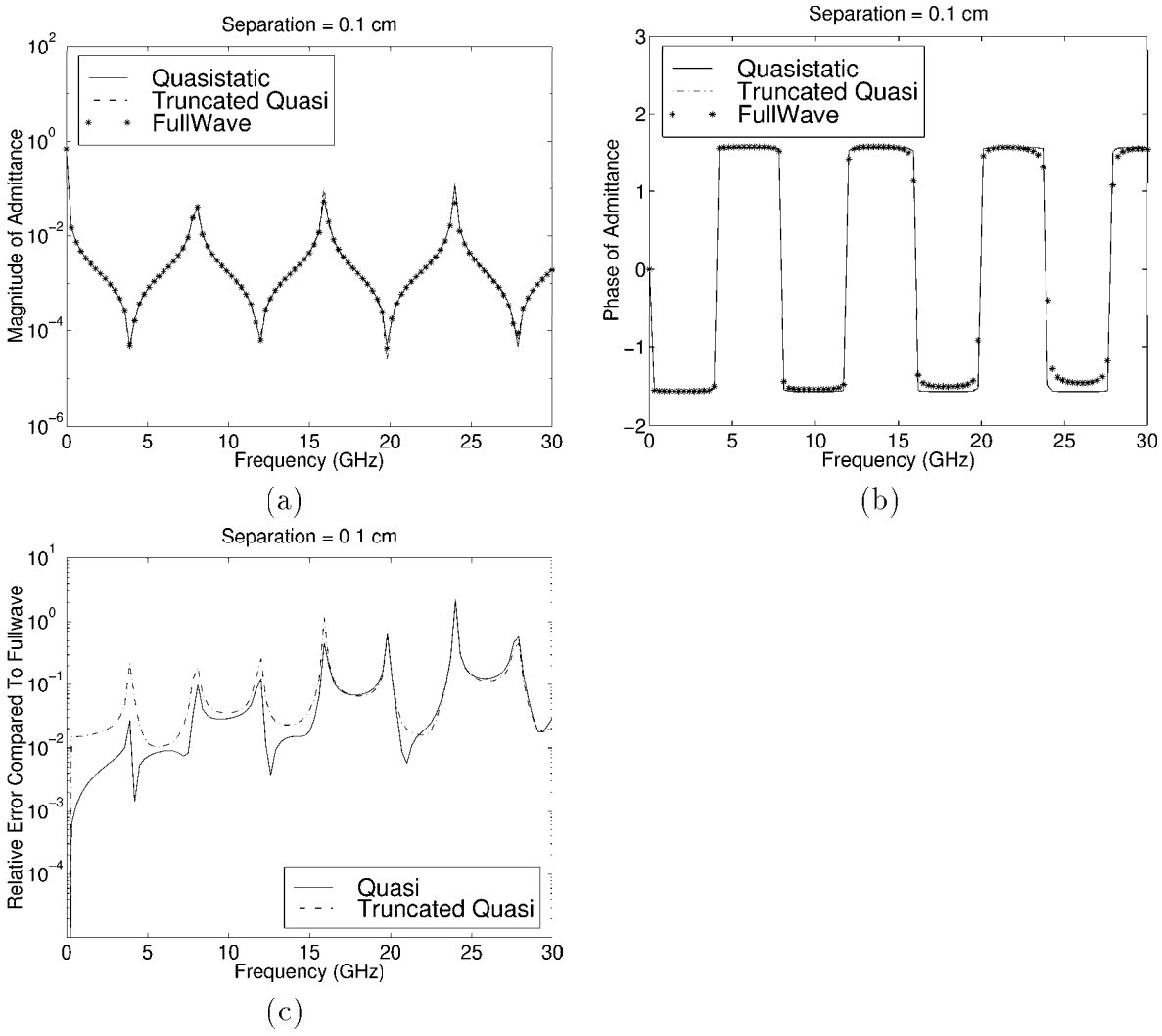


Figure A-2: Magnitude, phase, and error compared to full wave for a two conductor copper ($\sigma = 5.8 \times 10^7 (\Omega m)^{-1}$) transmission line, 1 cm long, separated by $0.1cm$.

order of a wavelength long with separations small compared to a wavelength. For larger separation, quasistatics captures the behavior only qualitatively.

These results rely on the nearby phase terms ($r_0 \leq \frac{10^{-2}}{24}$) being negligible. This assumption should be tested more thoroughly by computing L_{ij}^{full} more accurately for the nearby terms.

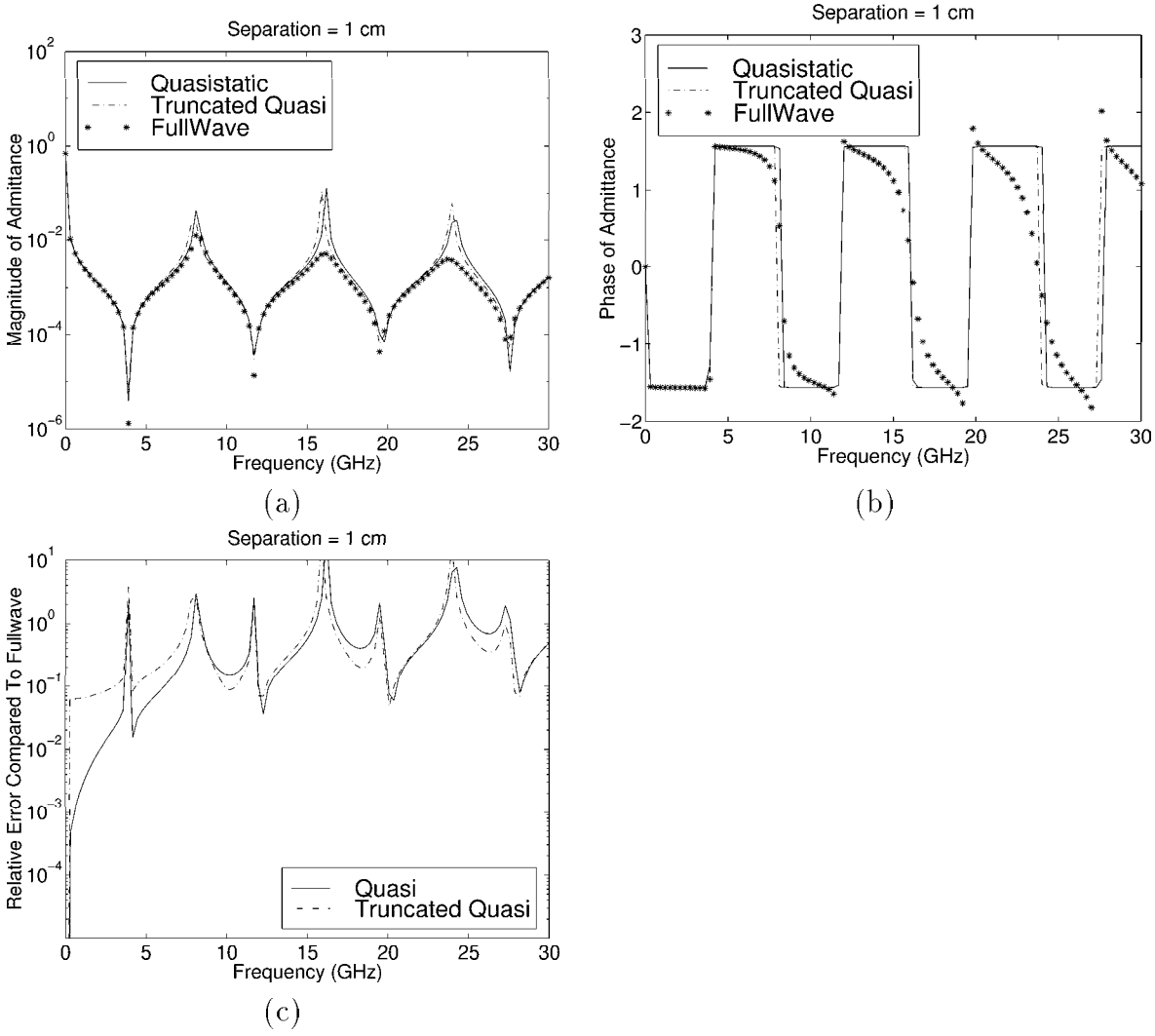


Figure A-3: Magnitude, phase, and error compared to full wave for a two conductor copper ($\sigma = 5.8 \times 10^7 (\Omega m)^{-1}$) transmission line, 1 cm long, separated by $1cm$.

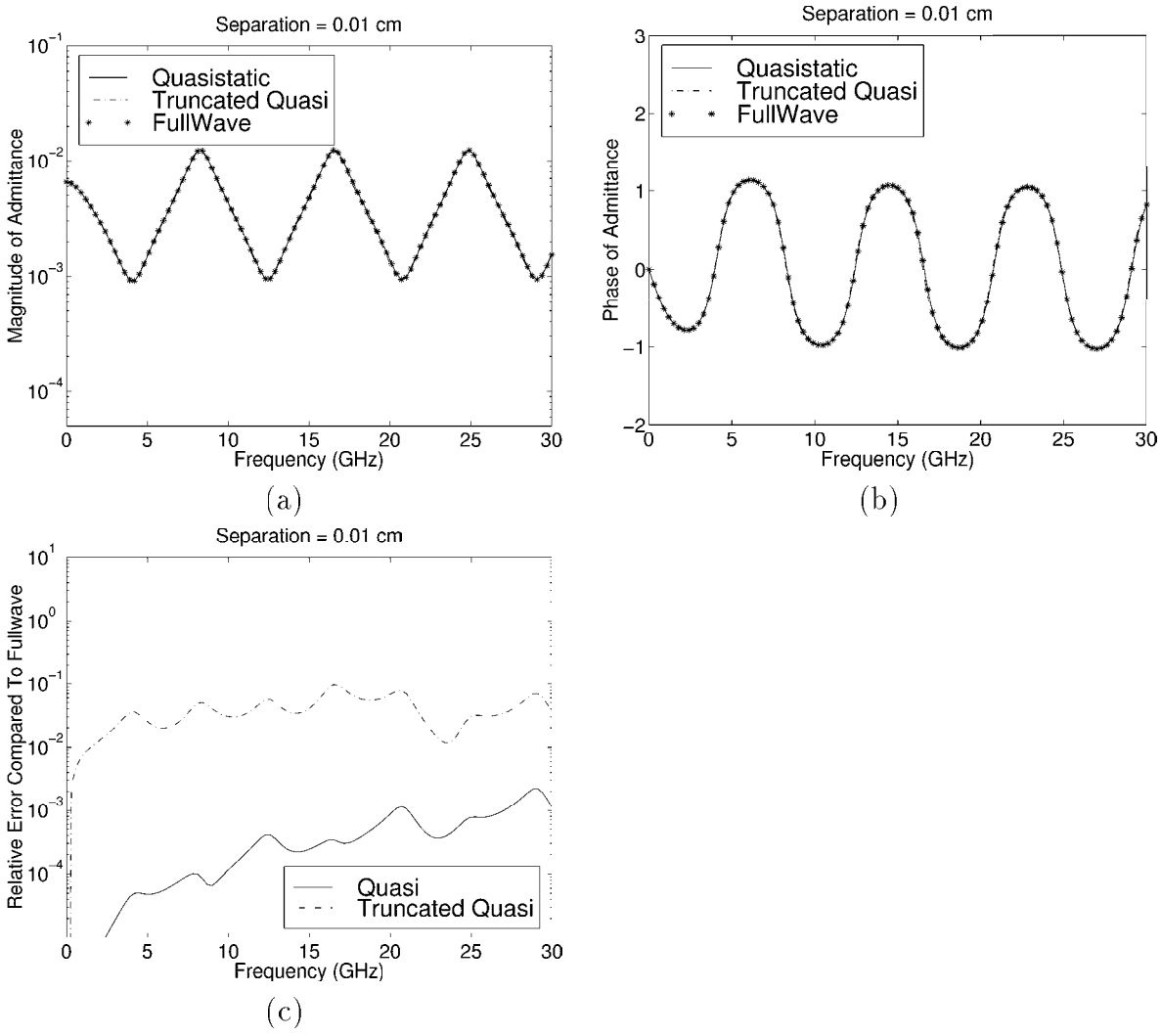


Figure A-4: Magnitude, phase, and error compared to full wave for a two conductor high resistivity ($\sigma = 5.8 \times 10^5 (\Omega m)^{-1}$) transmission line, 1 cm long, separated by $0.01cm$.

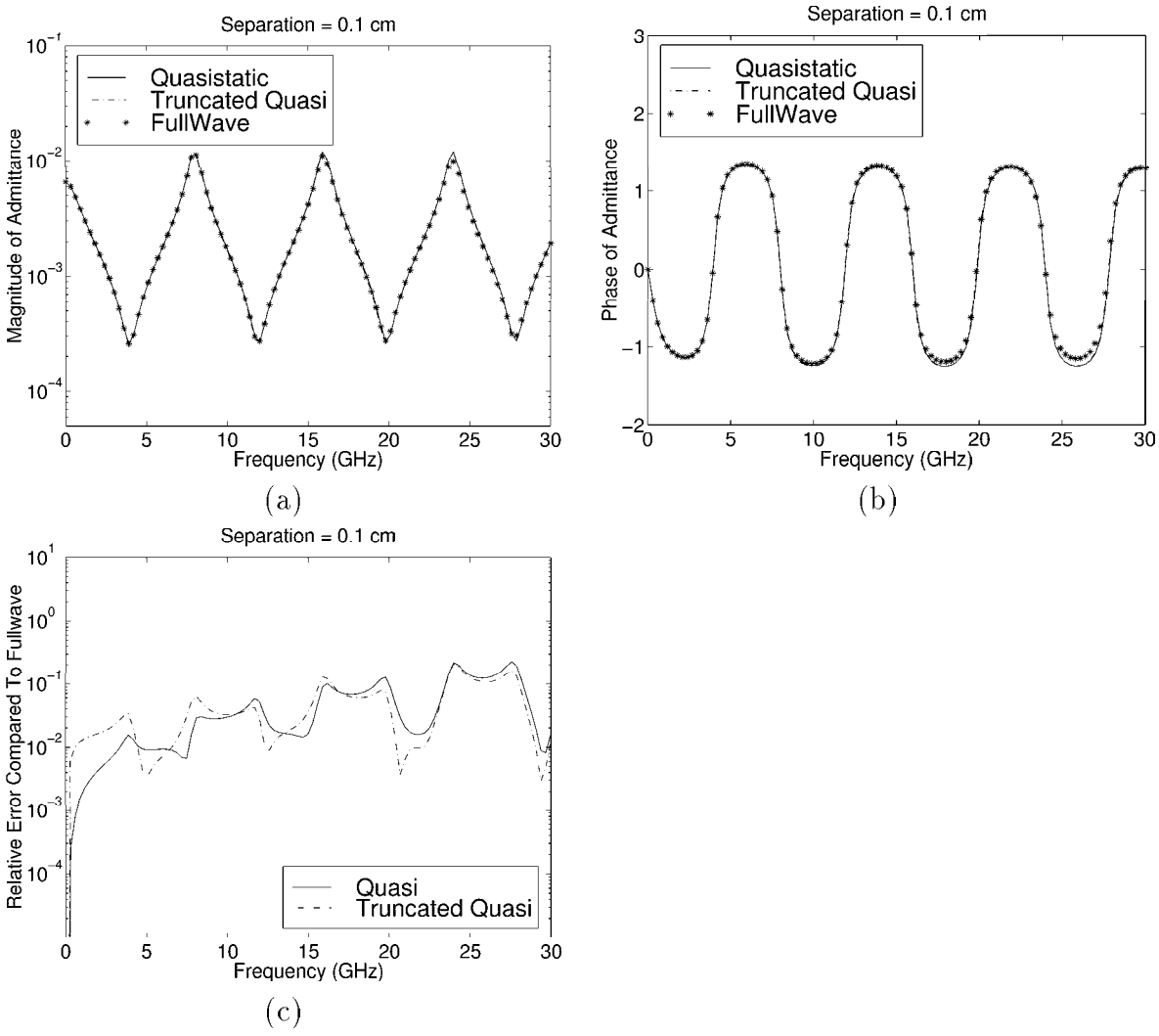


Figure A-5: Magnitude, phase, and error compared to full wave for a two conductor high resistivity ($\sigma = 5.8 \times 10^5 (\Omega m)^{-1}$) transmission line, 1 cm long, separated by $0.1cm$.

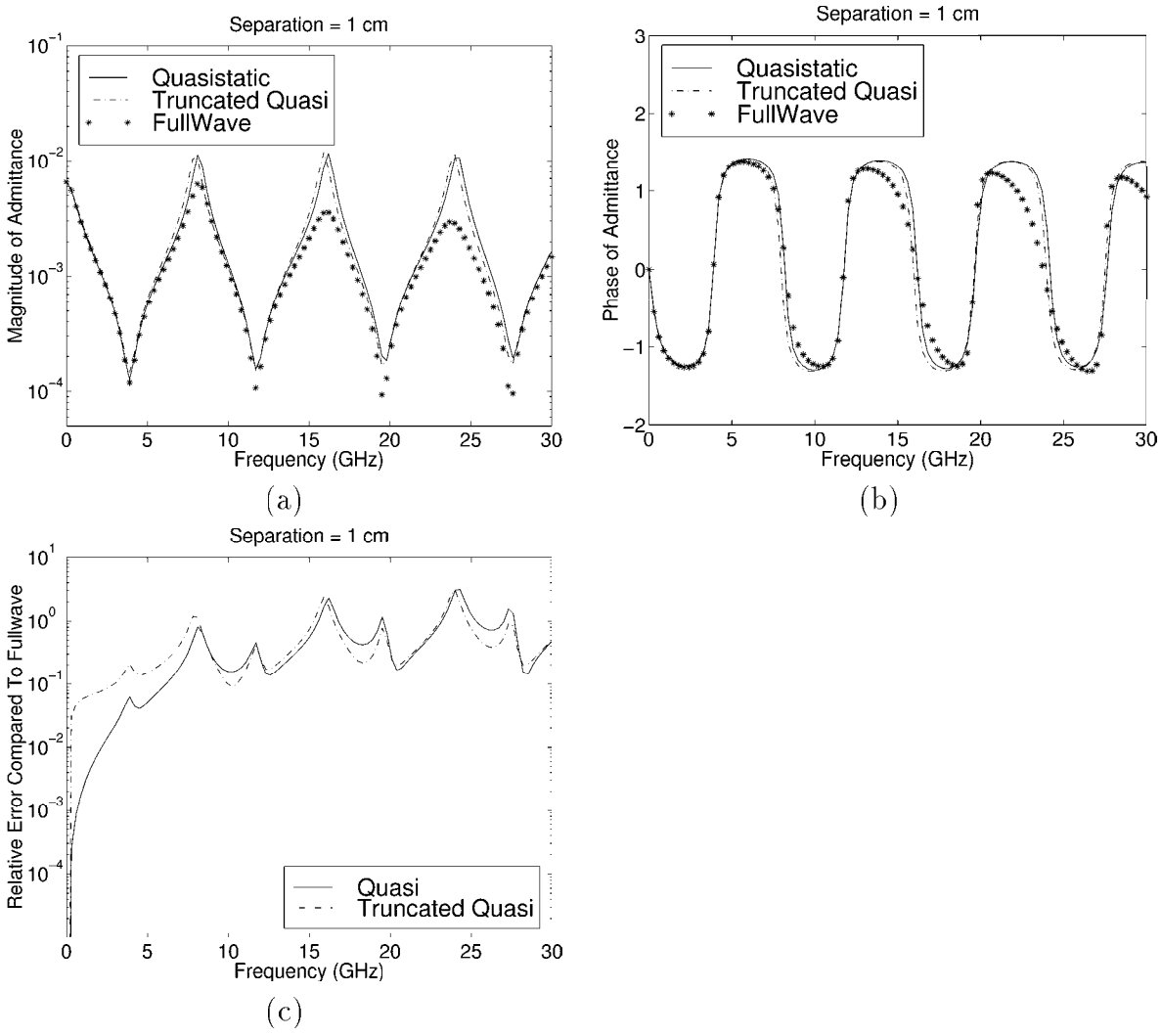


Figure A-6: Magnitude, phase, and error compared to full wave for a two conductor high resistivity ($\sigma = 5.8 \times 10^5 (\Omega m)^{-1}$) transmission line, 1 cm long, separated by $1cm$.

Bibliography

[1] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher. *Computational Fluid Mechanics and Heat Transfer.* Hemisphere Publishing, New York, 1984.

[2] W. E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quart. Appl. Math.*, 9:17–29, 1951.

[3] E. Arvas and R.F. Harrington. Computation of the magnetic polarizability of conducting disks and the electric polarizability of apertures. *IEEE Transactions on Antennas and Propagation*, 31(5):719–725, September 1983.

[4] George A. Baker Jr. *Essentials of Padé Approximants.* Academic Press, New York, NY, First edition, 1975.

[5] P. Benedek and P. Silvester. Capacitance of parallel rectangular plates separated by a dielectric sheet. *IEEE Transactions on Microwave Theory and Techniques*, MTT-20(8):504–510, August 1972.

[6] P. A. Brennan, N. Raver, and A. Ruehli. Three dimensional inductance computations with partial element equivalent circuits. *IBM Journal of Res. and Develop.*, 23(6):661–668, November 1979.

[7] A. C. Cangellaris, J. L. Prince, and L. P. Vakanas. Frequency-dependent inductance and resistance calculation for three-dimensional structures in high-speed interconnect systems. *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, 13(1):154–159, March 1990.

[8] Eli Chiprout and Michael S. Nakhla. Analysis of interconnect networks using complex frequency hopping (CFH). *IEEE Trans. CAD*, 14(2):186–200, February 1995.

[9] T. Chou, J. Cosentino, and Z. J. Cendes. High-speed interconnect modeling and high-accuracy simulation using spice and finite element methods. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 684–690, Dallas, TX, June 1993.

[10] C. Desoer and E. Kuh. *Basic Circuit Theory.* McGraw-Hill, New York, 1969.

[11] Ibrahim M. Elfadel and D. D. Ling. A block rational arnoldi algorithm for multipoint passive model-order reduction of multiport rlc networks. In *International Conference on Computer Aided-Design*, San Jose, California, November 1997.

[12] A. Farrar and A. T. Adams. Matrix methods for microstrip three-dimensional problems. *IEEE Transactions on Microwave Theory and Techniques*, MTT-20(8):497–504, August 1972.

[13] Peter Feldmann and Roland W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *EURO-DAC'94 with EURO-VHDL'94*, September 1994.

[14] L. Fortuna, G. Nunnari, and A. Gallo. *Model order reduction techniques with applications in electrical engineering.* Springer-Verlag, New York, 1992.

[15] K. Gallivan, E. Grimme, and P. Van Dooren. Asymptotic Waveform Evaluation via a Lanczos Method. *Applied Mathematics Letters*, 7(5):75–80, 1994.

[16] K. Gallivan, E. Grimme, and P. Van Dooren. Multi-point padé approximants of large-scale systems via a two-sided rational krylov algorithm. In 33^{rd} *IEEE Conference on Decision and Control*, Lake Buena Vista, FL, December 1994.

[17] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ error bounds. *Int. J. Control*, 36:1115–1193, 1984.

[18] Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* The John Hopkins University Press, Baltimore, Maryland, 1983.

[19] W. B. Gragg and A. Lindquist. On the partial realization problem. *Linear Algebra and its Applications*, 50:277–319, 1983.

[20] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* M.I.T. Press, Cambridge, Massachusetts, 1988.

[21] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.

[22] E. J. Grimme. *Krylov Projection Methods for Model Reduction.* PhD dissertation, University of Illinois at Urbana-Champaign, Electrical Engineering, 1997.

[23] G. D. Hachtel, R. K. Brayton, and F. G. Gustavson. The sparse tableau approach to network analysis and design. *IEEE Transactions on Circuit Theory*, 18:101–113, January 1971.

[24] R. F. Harrington. *Field Computation by Moment Methods.* MacMillan, New York, 1968.

[25] H. A. Haus and J. R. Melcher. *Electromagnetic Fields and Energy.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

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

[27] J. D. Jackson. *Classical Electrodynamics.* John Wiley & Sons, New York, second edition, 1975.

[28] I. M. Jaimoukha and E. M. Kasenally. Implicitly restarted krylov subspace methods for stable partial realizations. *SIAM J. Matrix Anal. Appl.*, 18(3):633–652, 1997.

[29] V. Jandhyala, E. Michielssen, and R. Mittra. Multipole-accelerated capacitance computation for 3-d structures in a stratified dielectric medium using a closed form Green's function. *Int. J. Microwave and Millimeter-Wave Computer-Aided Eng.*, 5(2):68–78, 1995.

[30] Thomas Kailath. *Linear Systems.* Information and System Science Series. Prentice-Hall, Englewood Cliffs, New Jersey, First edition, 1980.

[31] M. Kamon and S. S. Majors. Package and interconnect modeling of the hfa3624, a 2.4 ghz rf to if converter. In *Proceedings of the* 33^{rd} *Design Automation Conference*, pages 2–7, Las Vegas, NV, June 1996.

[32] M. Kamon, N. Marques, L. M. Silveira, and J. White. Automatic generation of accurate circuit models of 3-d interconnect. *submitted to IEEE Transactions on Components, Packaging, and Manufacturing Technology-Part B*, 1998.

[33] M. Kamon, N. Marques, and J. White. FastPep: A Fast Parasitic Extraction Program for Complex Three-Dimensional Geometries. In *IEEE/ACM International Conference on Computer Aided-Design*, San Jose, California, November 1997.

[34] M. Kamon and J. R. Phillips. Preconditioning techniques for constrained vector potential integral equations, with application to 3-D magnetoquasistatic analysis of electronic packages. In *Proceedings of the Colorado Conference on Iterative Methods, Breckenridge, Colorado*, April 1994.

[35] M. Kamon, M. J. Tsuk, and J. White. Fasthenry: A multipole-accelerated 3-d inductance extraction program. *IEEE Transactions on Microwave Theory and Techniques*, 42(9):1750–1758, September 1994.

[36] Mattan Kamon. Efficient techniques for inductance extraction of complex 3-d geometries. Master's thesis, Massachusetts Institute of Technology, Electrical Engineering and Computer Science, February 1994.

[37] K. J. Kerns, I. L. Wemple, and A. T. Yang. Stable and efficient reduction of substrate model networks using congruence transforms. In *IEEE/ACM International Conference on Computer Aided Design*, pages 207 – 214, San Jose, CA, November 1995.

[38] B. Krauter and L. Pileggi. Generating sparse partial inductance matrices with guaranteed stability. In *International Conference on Computer Aided-Design*, November 1995.

[39] E. S. Kuh and R. A. Rohrer. The state-variable approach to network analysis. *Proceedings of the IEEE*, 53:672–686, 1965.

[40] B.-T. Lee, E. Tuncer, and D. P. Neikirk. 3-d series impedance determination using a surface ribbon method. In *Proceedings of the* 4^{th} *Topical Meeting on Electrical Performance of Electronic Packaging*, pages 220–222, Portland, Oregon, October 1995.

[41] Shen Lin and Ernest S. Kuh. Transient Simulation of Lossy Interconnects Based on the Recursive Convolution Formulation. *IEEE Trans. Circuits Syst.*, 39(11):879–892, November 1992.

[42] D. Ling and A. E. Ruehli. Interconnect modeling. In A. E. Ruehli, editor, *Circuit Analysis, Simulation and Design, 2*, chapter 11, pages 211–332. Elsevier Science Publishers B. V., North-Holland, 1987.

[43] R. Lowther, P. Begley, G. Bajor, and A. Rivoli. Substrate parasitics and dual-resistivity substrates. *IEEE Transactions on Microwave Theory and Techniques*, page 1170, July 1996.

[44] Nuno Marques, Mattan Kamon, Jacob White, and L. Miguel Silveira. An efficient algorithm for fast parasitic extraction and passive order reduction of 3d interconnect models. In *to appear in Proceedings of DATE'98, Design Automation and Test in Europe*, February 1998.

[45] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26(1):17–32, February 1981.

[46] K. Nabors and J. White. Fastcap: A multipole accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(11):1447–1459, November 1991.

[47] K. Nabors and J. White. Fast capacitance extraction of general three-dimensional structures. *IEEE Trans. on Microwave Theory and Techniques*, June 1992.

[48] K. Nabors and J. White. Multipole-accelerated capacitance extraction algorithms for 3-D structures with multiple dielectrics. *IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications*, 39(11):946–945, November 1992.

[49] L. W. Nagel. SPICE2: A Computer Program to Simulate Semiconductor Circuits. Technical Report ERL M520, Electronics Research Laboratory Report, University of California, Berkeley, Berkeley, California, May 1975.

[50] Altan Odabasioglu, Mustafa Celik, and Lawrence Pileggi. Prima: Passive reduced-order interconnect macromodeling algorithm. In *International Conference on Computer Aided-Design*, San Jose, California, November 1997.

[51] J. I. Perez-Arriaga, G. C. Verghese, and F. C. Schweppe. Selective modal analysis with applications to electric power systems. part i: Heuristic introduction. part ii: The dynamic stability problem. *IEEE Transactions on Power Apparatus and Systems*, pages 3117–3134, September 1982.

[52] J. R. Phillips, E. Chiprout, and D. D. Ling. Efficient full-wave electromagnetic analysis via model-order reduction of fast integral transforms. In *Proceedings 33rd Design Automation Conference*, Las Vegas, Nevada, June 1996.

[53] Lawrence T. Pillage and Ronald A. Rohrer. Asymptotic Waveform Evaluation for Timing Analysis. *IEEE Trans. CAD*, 9(4):352–366, April 1990.

[54] A. E. Ruehli. Inductance calculations in a complex integrated circuit environment. *IBM J. Res. Develop.*, 16:470–481, September 1972.

[55] A. E. Ruehli. Equivalent circuit models for three-dimensional multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, 22(3):216–221, March 1974.

[56] A. E. Ruehli and P. A. Brennan. Efficient capacitance calculations for three-dimensional multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, 21(2):76–82, February 1973.

[57] A. Ruhe. The rational krylov algorithm for nonsymmetric eigenvalue problems iii: Complex shifts for real matrices. *BIT*, 34:165–176, 1994.

[58] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7:856–869, July 1986.

[59] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, 1996.

[60] J. E. Schutt-Aine and R. Mittra. Scattering Parameter Transient Analysis of Transmissions Lines loaded with Nonlinear Terminations. *IEEE Transactions on Microwave Theory and Techniques*, MTT-36:529–536, 1988.

[61] L. Miguel Silveira, Mattan Kamon, Ibrahim M. Elfadel, and Jacob K. White. A Coordinate-Transformed Arnoldi Algorithm for Generating Guaranteed Stable Reduced Order Models of RLC Circuits. In *IEEE/ACM International Conference on Computer Aided-Design*, San Jose, California, November 1996.

[62] L. Miguel Silveira, Mattan Kamon, and Jacob K. White. Algorithms for Coupled Transient Simulation of Circuits and Complicated 3-D Packaging. *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, 18(1):92–98, February 1995.

[63] J. Tausch and J. White. Mesh refinement strategies for capacitance extraction based on residual errors. In *Proceedings of the* 5^{th} *Topical Meeting on Electrical Performance of Electronic Packaging*, pages 236–237, Napa, California, October 1996.

[64] R. Telichevesky, K. Kundert, and J White. Efficient AC and noise analysis of two-tone RF circuits. In *Proceedings 33rd Design Automation Conference*, Las Vegas, Nevada, June 1996.

[65] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[66] M. J. Tsuk and J. A. Kong. A hybrid method for the calculation of the resistance and inductance of transmission lines with arbitrary cross sections. *IEEE Transactions on Microwave Theory and Techniques*, 39:1338–1347, August 1991.

[67] R. S. Varga. *Matrix iterative analysis.* Prentice Hall, Englewood Cliffs, New Jersey, 1962.

[68] S. A. Vavasis. Preconditioning for boundary integral equations. *SIAM Journal of Matrix Analysis and Applications*, 13(3):905–925, July 1992.

[69] Jiri Vlach and Kishore Singhal. *Computer Methods for Circuit Analysis and Design.* Van Nostrand Reinhold, Berkshire, England, 1983.

[70] W. T. Weeks, L. L. Wu, M. F. McAllister, and A. Singh. Resistive and inductive skin effect in rectangular conductors. *IBM Journal of Res. and Develop.*, 23(6):652–660, November 1979.