6-1

Substrate Resistance Extraction using a Multi-Domain Surface Integral Formulation

by

Anne M. Vithayathil

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

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

©2004 Massachusetts Institute of Technology
All rights reserved

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
JUL 2 6 2004
LIBRARIES

Author ....
Department of Electrical Engineering and Computer Science
April 20, 2004

Certified by......................
Jacob K. White
Professor
Thesis Supervisor

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Students

BARKER

Substrate Resistance Extraction using a Multi-Domain Surface Integral Formulation

by

Anne M. Vithayathil

Submitted to the Department of Electrical Engineering and Computer Science
on April 20, 2004, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

In recent years, mixed-signal designs have become more pervasive, due to their efficient use of area and power. Unfortunately, with sensitive analog and fast digital circuits sharing a common, non-ideal substrate, such designs carry the additional design burden of electromagnetic coupling between contacts.

This thesis presents a method that quickly extracts the electroquasistatic coupling resistances between contacts on a planar, rectangular, two-layer lossy substrate, using an FFT-accelerated multi-domain surface integral formulation. The multi-domain surface integral formulation allows for multi-layered substrates, without meshing the volume. This method has the advantages of easy meshing, simple implementation, and FFT-accelerated iterative methods. Also, a three-dimensional variant of this method allows for more complex substrate geometries than some other surface integral techniques, such as multilayered Green's functions; this three-dimensional problem and its solution are presented in parallel with the planar substrate problem and solution. Results from a C++ implementation are presented for the planar problem.

Thesis Supervisor: Jacob K. White
Title: Professor

Acknowledgments

An Acknowledgements section is supposed to be one page long, but really, what is the cost of printing one extra page? I've been lucky enough to have many supporters, and I'd be remiss to omit one.

I have Prof. Jacob White to thank for the inspiration of this thesis, as well as innumerable suggestions, encouragements and chastisements. Also to be thanked profusely is Xin Hu, a good friend, but more more importantly for the purposes of this thesis, a collaborator in much of the substrate coupling work, in particular the fundamentals and gruntwork of the two-dimensional problem. I must acknowledge that I couldn't have done it without our collaboration.

It is imperative that I acknowledge those sources of funding of which I am aware: a National Science Foundation Graduate Research Fellowship, as well as SRC funding.

Shihhsien Kuo, Jay Bardhan, and Prof. Jacob White's multi-domain approach to the area of biomolecular modeling was much of the inspiration for this thesis, and their paper and assistance provided good guidance. Also, Luke Theogarajan filled in many of my gaps of knowledge regarding fabrication, design and much more. I have Profs. Richard Shi and Vickram Jandhyala, of University of Washington, to thank for initiating me into the mysteries of EDA and CEM, respectively.

I consider myself lucky to have wonderful officemates and groupmates. In particular, I thank Jaydeep Bardhan, Dave Willis, Carlos Pinto Coelho, Rikky Muller, Dimitry Vasilyev, Lily Wang and Xin Wang for putting up with my general slovenliness and answering my random enquiries about random topics at inconvenient times. Groupmates Xin Hu, Joe Kanapka, Tom Klemas, Shihhsien Kuo, Jung Hoon Lee, Michal Rewienski, John D. Rockway (!), Ben Song, Dennis Wu and Zhenhai Zhu were also pestered, about this thesis topic and much more, although I had to go down the hall to do it. Thank you to Chad Collins for bailing me out of all kinds of practical problems. I have appreciated the convivial tone of our group and the 8th floor in general.

I have my friends to thank for dragging me out of the lab, and keeping me sane. In particular, thank you to those of you who (horror!) are not in Engineering and/or (horror!) not at MIT. You guys have provided an amusing reality check, and great a cheerleading squad, even when you couldn't remember what my major is (it's Electrical Engineering, by the way). Speaking of sanity and reality, for the past nine months, I have had the residents of PTZ to thank for keeping me insane and wondering about reality. CJ, I know you will insist on reading this thesis, even though math gives you hives, so thanks to you in particular for putting up with me for the last fifteen years.

Much thanks and appreciation to the Rockway clan for everything you've done for the past 5 years: Joy, Jay and Jeanne. Thanks to all my family: the nuclear Mom, Dad, Joey, Francis, but also my extended family, in particular Darrell Jackson, Linda Shapiro and all of the innumerable Vithayathils who have fed and disciplined me over the years. It's hard to summarize all they have given me in 25+ years, but let's just say, I know where the credit is due for any and all of my accomplishments. Hopefully my Mom won't tape this thesis to the refrigerator, or worse, frame it and talk about a child's innate sense of numerical analysis.

Finally, a big thank you to my biggest (and most forgiving) fan, John Dexter Rockway. Since I've never been the sentimental type, I hope a simple ruv ru and thank you will suffice to express my gratitude.

Contents

1 Introduction **13**

1.1 Background 14

1.1.1 Substrate coupling 14

1.1.2 Previous research 15

1.2 Problem and methodology 15

1.2.1 Problem statement 15

1.2.2 Methodology 16

1.2.3 Comparison 18

1.3 Thesis overview 19

2 Multi-domain surface integral formulation **21**

2.1 Planar problem 21

2.1.1 Geometry 22

2.1.2 Coupled surface integral equations 22

2.1.3 Boundary conditions 25

2.2 Three-dimensional problem 26

2.2.1 Geometry 26

2.2.2 Coupled surface integral equations 26

2.2.3 Boundary conditions 28

2.3 Equation summary 29

2.3.1 Planar problem 29

2.3.2 Three-dimensional problem 30

3 Problem discretization 33

3.1 Discretization of planar problem 33

3.1.1 Geometric mesh and basis functions 33

3.1.2 System matrix equation 36

3.2 Discretization of three-dimensional problem 38

3.2.1 Geometric mesh and basis functions 38

3.2.2 System matrix equation 40

3.3 Iterative method for matrix solution 42

4 FFT acceleration of matrix-vector products 45

4.1 Planar problem 45

4.1.1 Decomposition of matrix-vector product 46

4.1.2 Accelerating dense submatrix-subvector products 47

4.1.3 Accelerated matrix-vector product 48

4.2 Three-dimensional problem 50

4.2.1 Decomposition of matrix-vector product 50

4.2.2 Accelerating planar to planar scattering 50

4.2.3 Accelerating three-dimensional interactions using pFFT 51

4.2.4 Accelerated matrix-vector product 55

5 Computational results 59

5.1 Code summary 59

5.2 Test cases 60

5.3 Results 61

5.3.1 Convergence: iteration counts 61

5.3.2 Computational complexity 62

5.3.3 Computational complexity of program subtasks 62

5.3.4 Computational complexity of iterative method 62

5.3.5 Computational complexity of matrix-vector product 64

List of Figures

1-1 Planar problem geometry. 16

1-2 Three-dimensional problem geometry. 17

2-1 Geometry of a planar substrate 22

2-2 Geometry of a three dimensional substrate 27

3-1 Problem discretization for planar problem 34

3-2 Unknown vector for planar problem 36

3-3 System matrix structure for the planar problem 37

3-4 Breakdown of three-dimensional problem geometry into different components 38

3-5 Unknown vector for three-dimensional problem 40

3-6 System matrix structure for the three-dimensional problem 41

3-7 Preconditioner sparsity pattern for planar problem 43

3-8 Preconditioner sparsity pattern for three-dimensional problem 44

4-1 Matrix-vector product in block form, for the planar problem 46

4-2 Matrix-vector product in block form, for the three-dimensional problem 51

4-3 Three dimensional panel within its pFFT cell with surrounding collocation sphere 53

5-1 Three different top boundary conditions for problem: two-contact (left), random (center), striped (right) 61

Chapter 1

Introduction

In recent years, mixed-signal designs have become more pervasive, due to their efficient use of area and power. Unfortunately, since sensitive analog and fast digital circuits share a common non-ideal substrate, such designs carry the additional design burden of electromagnetic coupling between contacts.

This thesis presents a method that quickly extracts the coupling resistances between contacts on a two-layer lossy substrate, using an FFT-accelerated multi-domain surface integral formulation. This thesis also addresses the same extraction problem when the substrate has some three-dimensional features, such as isolation rings; such features are accommodated by specializing the precorrected-FFT method [2]. Much of the work for the planar problem was originally introduced in [1].

In this chapter, we will first motivate the problem and then briefly identify previous research efforts in this area. The problem of interest will be introduced, as well highlights of our methodology. This method will then be compared with some other available methods. Finally a short overview of the thesis contents is given.

1.1 Background

1.1.1 Substrate coupling

A non-ideal substrate has non-zero conductivity and permittivity, which couples the contacts resistively and capacitively, respectively. At higher frequencies, in excess of a gigahertz, the coupling relationship is better described by a complex coupling impedance. However, for lower frequencies and most substrates, for which the dielectric relaxation time is short, the resistive model is appropriate [5].

Such coupling resistances are detrimental to performance, as switching digital currents will bias sensitive analog components, even when these digital and analog blocks are separated by a large distance. Equation 1.1 describes the resistive substrate coupling for a design with M contacts. V and I are the vectors corresponding to contact potentials and currents, respectively, both of size the $M \times 1$. Y denotes the unknown coupling conductance matrix, of size $M \times M$; $|Y_{i,j}| = \frac{1}{R_{i,j}}$, the coupling resistance between contacts i and j.

$$YV = I \tag{1.1}$$

In order to find the jth column of Y, we apply a 1V potential to the jth contact, and set all other contacts to ground. We assign non-contact surfaces a boundary condition of zero current. Hence, in order to find the entire coupling resistance matrix, we must be able to quickly model the resultant currents due to an applied voltage vector. This modeling will be done M times, once for each column of the conductance matrix. With the conductance matrix found, the coupling effects can then be simulated by designers by using a lumped-element simulator, such as SPICE.

Similar matrix problems and methodologies exist for extracting either electroquasistatic capacitances or full-wave complex impedances, and some of methods will be discussed below; however, this thesis is devoted to the extraction of resistances only.

1.1.2 Previous research

Researchers have modeled substrate coupling with many different approaches, either for capacitance, resistance or full-wave impedance extraction. A finite difference approach is taken in [11], [12]. Although allowing for almost arbitrarily inhomogeneous media, this method requires the discretization of the entire volume of the substrate, plus some of the exterior domain. This volume discretization makes such methods unnecessarily expensive, both in memory and in computational effort. Furthermore, artificial boundary conditions must be employed to truncate the compuational domain.

A finite-element method is presented in [10]. This work relies on a hierarchical extraction using volumetric filaments, and has the same drawback of finite-differnence method: it requires the discretization of the entire volume.

Numerous researchers, including [8],[9], have presented work on multilayer Green's functions. Most of these methods require the discretization of contact surfaces only, albeit with a regular, rectangular grid. Some variations and improvements on these works have also been developed. In [13], a hybrid two-dimensional FEM/three-dimensional BEM method is presented. Reference [14], improves the finite difference method by sparsifying the system matrix via a wavelet basis. Reference [9] implements the multilayer Green's function with an FFT-accelerated matrix-vector product.

1.2 Problem and methodology

1.2.1 Problem statement

This thesis addresses the problem of coupling resistance extraction for a two-layer substrate, where the two rectangular volumes have conductivities σ_u and σ_l. The contacts are restricted to the top and bottom surfaces of the substrate. Although arbitrary boundary conditions could be applied to the bottom surface, this thesis will consider either a grounded or floating backplane. Such a geometry is shown in Figure 1-1, and we will subsequently refer to it as the planar geometry. We describe the coupling in the quasistatic limit, which suffices for operating frequencies to a few gigahertz, and for most substrate material properties.

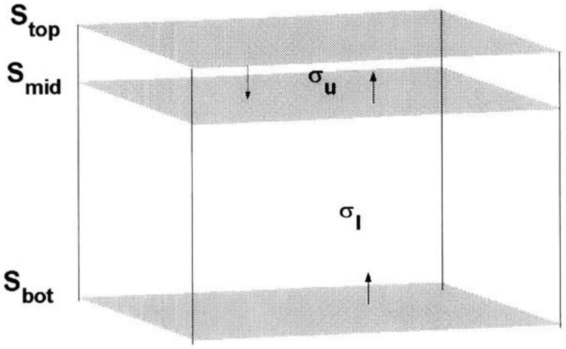

Figure 1-1: Planar problem geometry.

Some three-dimensional features, such as isolation rings, will also be considered. Such a geometry is shown in Figure 1-2. The rings or other three-dimensional features considered here are limited to rings, or other geometric variations in the top surface, such that any indentations do not extend into the lower substrate layer. More general three-dimensional features could be considered with an adaptation of our method, but we restrict the geometry for clarity of presentation. Furthermore, for convenience, we will limit ourselves to contacts only on the top planar surface, and set any indented surface to have a no-current flow condition.

1.2.2 Methodology

The work is motivated by the recent application of a similar FFT-accelerated multi-domain surface integral formulation developed in the area of biomolecular modeling

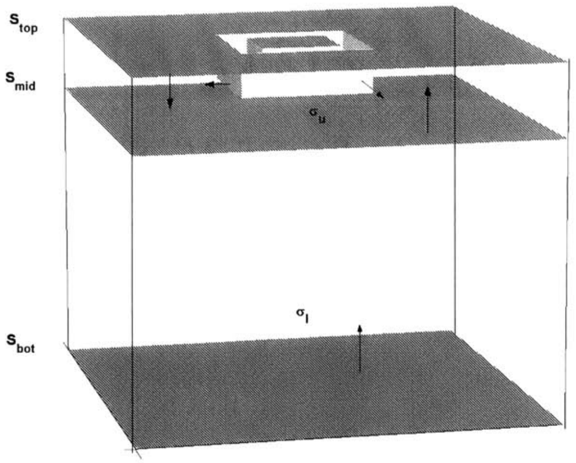

Figure 1-2: Three-dimensional problem geometry.

[4].

Our solution makes use of a surface integral formulation derived from Laplace's equation for each homogeneous layer in the substrate; this system is derived as found in [4], [3], [5]. We discretize this coupled system using the method of moments technique [6] on a uniform, rectangular grid overlaid on the surfaces of the substrate, and its interface between layers. The resultant matrix is solved using a Krylov-subspace iterative method, with a block-diagonal right preconditioner. The Krylov-subspace method requires a fast matrix-vector product. For the planar problem we implement this product using two-dimensional Fast Fourier Transforms (FFTs), as discussed in [7]; the three-dimensional problem requires a matrix-vector product accelerated by an adaptation of the precorrected FFT method (pFFT), as presented in [2].

The methodology is easily extensible to structures with more than two layers, but for greater clarity, such a derivation is omitted and we restrict our work to a two-layer problem. The methodology is also extensible to full-wave analysis, for situations when the operating frequency is greater than a few gigahertz, however, as mentioned previously, we restrict our attention to the resistance problem.

1.2.3 Comparison

The method presented has, like all methods, its advantages and disadvantages. We will see that the method makes use of the free-space Green's function, as opposed to a more specialized multilayered Green's function. This makes the implementation of the method much simpler, as evaluating and integrating this free-space kernel is a relatively easy problem, whereas many multilayered Green's functions require much more sophisticated techniques for their evaluation.

The regular, rectangular meshing also adds to the simplicity of the implementation, as it requires no commercial mesher, but rather, a very simple home-grown meshing code. Unlike volumetric approaches such as finite-difference and finite-element techniques [11], [12], [10], our method requires the meshing of only surfaces, not the entire substrate volume, which makes our method more efficient with respect to memory and computation.

Another advantage is that our method can be adapted to accommodate three-dimensional features such as the isolation rings of our three-dimensional problem.

These advantages are tempered slightly by a few disadvantages. First, unlike many multilayered Green's function techniques, we are required to mesh the entire surface of the substrate, as well as the interfaces between layers. Multilayer Green's function techniques require only the meshing of the contacts. However, for dense designs, where most of the surface is covered by contacts, this consideration does not weigh as heavily.

The simplicity of the meshing presents another disadvantage, because the fineness of the mesh must be chosen to accommodate the smallest feature on the substrate. If this proved prohibitively expensive, however, this disadvantage could be overcome by using a nonuniform mesh, and employing a pre-corrected FFT method.

The surface nature of the method does not allow for irregular material inhomogeneity. However, for many applications, our assumption of a layered medium is appropriate. This disadvantage is shared by all surface integral techniques, including most multilayered techniques [8], [9].

Finally, the method employs a small geometric assumption about the aspect ratio of the substrate. We assume that the thickness of the substrate is much smaller than its length and width. For most substrate coupling applications, this assumption is valid.

1.3 Thesis overview

The second chapter of this thesis introduces the problem geometry in greater detail and derives the multi-domain surface integral equations that describe the coupling of the two-layer structure. The third chapter illustrates the discretization of the surface integral equations, as well as the iterative solution of the resultant matrix equation. We will demonstrate that a solution can be found with an iterative method, requiring only an efficient means for computing the matrix-vector product, and a simple preconditioner. The fourth chapter discusses the acceleration of this matrix-vector product using two-dimensional FFTs. This chapter will also address the impact of three-dimensional features on this matrix-vector product. The fifth chapter presents numerical results of the C++ implementation of our method for the planar problem.

Chapter 2

Multi-domain surface integral formulation

In this chapter we derive the coupled system of integral equations that govern the potentials and conductive currents of our coupling problems, for both planar and three-dimensional geometries. We first derive the equations governing our planar problem. In the second part of the chapter, we adapt the same derivation to handle three-dimensional features such as isolation rings. A summary of the system of equations is given at the end of the chapter.

2.1 Planar problem

In this section we will derive the equations describing the planar problem. We begin by introducing the geometry involved. We then find a surface integral equation for each of the two homogeneous substrate volumes, using Green's second identity. Next, these two equations are coupled using continuity relations. Finally, using imposed Dirichlet and Neumann boundary conditions, we are able to specify the entire system of equations governing the behavior of the substrate.

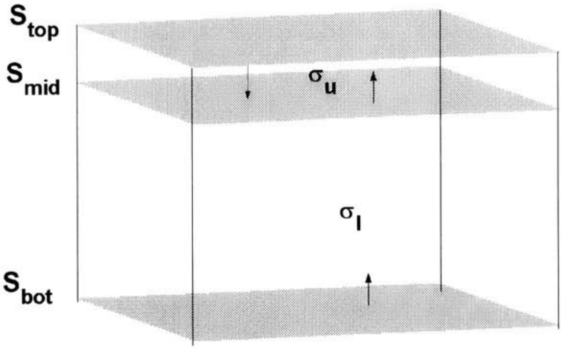

Figure 2-1: Geometry of a planar substrate

2.1.1 Geometry

Consider the planar problem geometry shown in Figure 2-1.

We will refer to the upper and lower homogeneous volumes shown as Ω_u and Ω_l, respectively. $\partial\Omega_u$ and $\partial\Omega_l$ will refer to the surfaces bounding these two volumes, and σ_u, σ_l, to their conductivities.

Note that the top, interface and bottom planar surfaces are denoted as S_{top}, S_{mid}, and S_{bot}, respectively. The normals to these surfaces point inward,as shown in the figure. The normals on the sides of the substrate (not shown in the figure) are oriented inwards.

Note that the width and depth of the substrate are assumed to be much larger than the height of the substrate ($>= 10\times$). This assumption will be used later to simplify the form of the integral equation. Also, the locations of the contacts are assumed to be far from the edges of the substrate.

2.1.2 Coupled surface integral equations

We begin by deriving the surface integral equation for the homogeneous upper volume; this process can then be repeated for the lower volume. We then introduce the continuity equations and boundary equations. A similar derivation is found for a biomolecular modeling problem in [4], and for Laplace's Equation in [5] and [3].

Surface integral equations for a homogeneous volume

For all three-dimensional points r in the upper volume, Ω_u, Laplace's equation for the potential, $\phi(r)$, holds.

$$\nabla^2\phi_u(r) = 0, r \in \Omega_u \tag{2.1}$$

The subscript u emphasizes the fact that the evaluation point, r, is in the upper volume, as opposed to the lower volume.

The fundamental solution to this equation, for a impulse source at the point r', is the Green's function

$$G(r,r') = \frac{1}{4\pi \mid r - r' \mid}. \tag{2.2}$$

Computing the Laplacian of G and multiplying by the potential yields,

$$\phi(r')\nabla^2 G(r,r') = \phi(r')\delta(r,r'). \tag{2.3}$$

Taking the volume integral over $r' \in \Omega_u$, and using the sifting property of the dirac delta function, we have:

$$\int\int\int_{\Omega_u} \phi_u(r')\nabla^2 G(r,r')dV = \phi_u(r), \forall r \in \Omega_u \tag{2.4}$$

Using Green's second identity, and recognizing that $\nabla^2\phi_u(r) = 0$ in the volume,

$$\phi_u(r) = -\int_{\partial\Omega_u} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' + \int_{\partial\Omega_u} \phi_u(r')\frac{\partial G(r,r')}{\partial n}dr', \forall r \in \Omega_u, \tag{2.5}$$

where $\partial\Omega_u$ denotes the boundary of volume Ω_u; note that the normal direction is defined inward. Taking the limit of this relationship as r approaches a point on $\partial\Omega_u$

$$\frac{\phi_u(r)}{2} = -\int_{\partial\Omega_u} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' + \int_{\partial\Omega_u} \phi_u(r')\frac{\partial G(r,r')}{\partial n}dr', \forall r \in \partial\Omega_u^{smooth}. \tag{2.6}$$

In this equation $\partial\Omega_u^{smooth}$ denotes the smooth portions of the boundary surface. The coefficient on the left-hand size of the equations differ for points on a sharp edge or corner, however, we will see later that we are not concerned with such portions.

We make use of our geometrical assumptions to approximate equation 2.6. Under the assumption that the linear dimensions of the problem are much greater than the depth of the substrate, and that the contacts are sufficiently far from the edges of the substrate, we will neglect the contributions to the surface integral that arise from integration of the sides. This allows us to simplify equation 2.6 to

$$\frac{\phi_u(r)}{2} = -\int_{S_{top} \cup S_{mid}} G(r,r') \frac{\partial \phi_u(r')}{\partial n} dr' + \int_{S_{top} \cup S_{mid}} \phi_u(r') \frac{\partial G(r,r')}{\partial n} dr'. \qquad (2.7)$$

Hence, we have the approximate surface integral relationship that governs field points immediately on the interior of the top volume.

The same derivation can be used to find the surface integral relationship governing the lower volume; note that some of the signs are changed as a result of the different normal orientations. A subscript l is used to emphasize the fact that r is a field point immediately on the interior of the lower volume.

$$\frac{\phi_l(r)}{2} = \int_{S_{mid} \cup S_{bot}} G(r,r') \frac{\partial \phi_l(r')}{\partial n} dr' - \int_{S_{mid} \cup S_{bot}} \phi_l(r') \frac{\partial G(r,r')}{\partial n} dr' \qquad (2.8)$$

Continuity relations

Continuity of potential and of current allow us to relate the potential and its normal derivative at the interface surface.

$$\phi_u(r) = \phi_l(r), \qquad r \in S_{mid} \qquad (2.9)$$

$$\sigma_u \frac{\partial \phi_u(r)}{\partial n} = \sigma_l \frac{\partial \phi_l(r)}{\partial n}, \quad r \in S_{mid} \qquad (2.10)$$

Coupled surface integral equations

Using the continuity relations, we eliminate ϕ_l and $\frac{\partial \phi_l}{\partial n}$ at S_{mid} from equation 2.8. The final system of coupled surface integral equations is then given by:

$$\begin{aligned}\frac{\phi_u(r)}{2} = &-\int_{S_{top}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' \\ &+\int_{S_{top}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' \end{aligned} \tag{2.11}$$

and

$$\begin{aligned}\frac{\phi_l(r)}{2} = &\,\alpha\int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' + \int_{S_{bot}} G(r,r')\frac{\partial\phi_l(r')}{\partial n}dr' \\ &-\int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' - \int_{S_{bot}} \frac{\partial G(r,r')}{\partial n}\phi_l(r')dr'. \end{aligned} \tag{2.12}$$

In equation (2.12), α represents the conductivity ratio of σ_u to σ_l.

2.1.3 Boundary conditions

We now have two unknowns on each of the three planar surfaces.

$$\phi_u(r), \frac{\partial\phi_u(r)}{\partial n}, \forall r \in S_{top} \tag{2.13}$$

$$\phi_u(r), \frac{\partial\phi_u(r)}{\partial n}, \forall r \in S_{mid} \tag{2.14}$$

$$\phi_l(r), \frac{\partial\phi_l(r)}{\partial n}, \forall r \in S_{bot} \tag{2.15}$$

The interface surface has two surface integral equations that govern the behavior, however, in order to constrain the system properly, Dirichlet or Neumann boundary conditions, or a mix of both, must be specified for the top and bottom surfaces of the substrate.

The top boundary condition is given by:

$$a(r)\phi_u(r) + b(r)\frac{\partial\phi_u(r)}{\partial n} = f(r), r \in S_{top}, \tag{2.16}$$

where for each position, r, on the top surface, $a(r) = 1$ and $b(r) = 0$ if r is on a contact; $a(r) = 0$ and $b(r) = 1$ otherwise. The function $f(r) = 0$ if r is not on a contact and is either one or zero depending on which column of the coupling conductance matrix is being computed.

The bottom boundary condition is:

$$c\phi_l(r) + d\frac{\partial \phi_l(r)}{\partial n} = 0, r \in S_{bot} \tag{2.17}$$

where $c = 1$, $d = 0$ for a grounded backside substrate and $c = 0$, $d = 1$ for an insulated backside.

2.2 Three-dimensional problem

2.2.1 Geometry

We show the three-dimensional geometry in Figure 2-2, part (i). Note that it is very similar to the original geometry, only now, a trench shape has been carved out of the top substrate layer. We define $\Omega_{u'}$ as the entire homogeneous upper volume, and $\partial\Omega_{u'}$ as its boundary surface. Since the lower volume has not changed, the definitions of Ω_l and $\partial\Omega_l$ given previously will still hold.

To tackle the problem efficiently, it is important to describe the geometry in terms of some subsurfaces. We will first identify each of these subsurfaces.

We define any three-dimensional feature of the geometry, in our case, the 3D trench shape shown in Figure 2-2, part (iv), as S_{3D}. The section shown in part (ii), is defined as before, in the planar case: the top surface S_{top}, the middle surface S_{mid} and the bottom surface S_{bot}. The flat ring shown in part(iii) is defined as $S_\star$; this corresponds to any part of S_{top} that does not actually appear in our geometry. For later convenience, we also define the surfaces $S_{top'} = S_{top} \setminus S_\star$ and $S_1 = S_{top'} \cup S_{3D}$.

2.2.2 Coupled surface integral equations

Now that the geometry has been introduced, we will find the governing integral equations for this geometry. The easist part, since nothing has changed in the lower

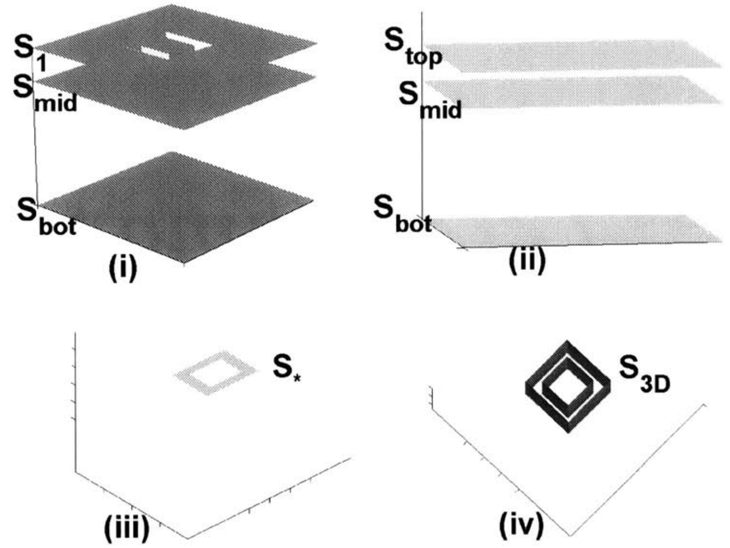

Figure 2-2: Geometry of a three dimensional substrate

volume, is the derivation for the lower volume's integral equation. We still have, as in equation 2.8

$$\frac{\phi_l(r)}{2} = \int_{S_{mid} \cup S_{bot}} G(r,r') \frac{\partial \phi_l(r')}{\partial n} dr' - \int_{S_{mid} \cup S_{bot}} \phi_l(r') \frac{\partial G(r,r')}{\partial n} dr'. \tag{2.18}$$

We now tackle the top volume. The previous derivation holds, up until equation 2.6, which states:

$$\frac{\phi_u(r)}{2} = -\int_{d\Omega_{u'}} G(r,r') \frac{\partial \phi_u(r')}{\partial n} dr' + \int_{d\Omega_{u'}} \phi_u(r') \frac{\partial G(r,r')}{\partial n} dr', \forall r \in \partial\Omega_{u'}^{smooth}. \tag{2.19}$$

We again use the geometric assumptions to approximate the integral as follows.

$$\frac{\phi_u(r)}{2} = -\int_{S_1 \cup S_{mid}} G(r,r') \frac{\partial \phi_u(r')}{\partial n} dr' + \int_{S_1 \cup S_{mid}} \phi_u(r') \frac{\partial G(r,r')}{\partial n} dr' \tag{2.20}$$

For computational convenience later, we break down the top integral over S_1.

$$\frac{\phi_u(r)}{2} = -\int_{S_{top'} \cup S_{3D} \cup S_{mid}} G(r,r') \frac{\partial \phi_u(r')}{\partial n} dr' + \int_{S_{top'} \cup S_{3D} \cup S_{mid}} \phi_u(r') \frac{\partial G(r,r')}{\partial n} dr' \tag{2.21}$$

Using the same continuity equation as before, we are again able to eliminate ϕ_l and $\frac{\partial \phi_l}{\partial n}$ at S_{mid} from equation 2.18.

The final system of coupled surface integral equations for the three-dimensional case is then given by:

$$\frac{\phi_u(r)}{2} = -\int_{S_{top'}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{3D}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' \\ + \int_{S_{top'}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{3D}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' \tag{2.22}$$

and

$$\frac{\phi_l(r)}{2} = \alpha\int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' + \int_{S_{bot}} G(r,r')\frac{\partial\phi_l(r')}{\partial n}dr' \\ - \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' - \int_{S_{bot}} \frac{\partial G(r,r')}{\partial n}\phi_l(r')dr'. \tag{2.23}$$

2.2.3 Boundary conditions

We now have two unknowns on each of the three planar surfaces, and on one nonplanar surface.

$$\phi_u(r), \frac{\partial\phi_u(r)}{\partial n}, \forall r \in S_{top'} \tag{2.24}$$

$$\phi_u(r), \frac{\partial\phi_u(r)}{\partial n}, \forall r \in S_{3D} \tag{2.25}$$

$$\phi_u(r), \frac{\partial\phi_u(r)}{\partial n}, \forall r \in S_{mid} \tag{2.26}$$

$$\phi_l(r), \frac{\partial\phi_l(r)}{\partial n}, \forall r \in S_{bot} \tag{2.27}$$

The boundary equation for $S_{top'}$ is the same as it was for S_{top}:

$$a(r)\phi_u(r) + b(r)\frac{\partial\phi_u(r)}{\partial n} = f(r), r \in S_{top'}, \tag{2.28}$$

where for each position on the top surface, $a(r) = 1$ and $b(r) = 0$ if r is on a contact; $a(r) = 0$ and $b(r) = 1$ otherwise. The function $f(r) = 0$ if r is not on a contact and is either one or zero depending on which column of the coupling conductance matrix is being computed.

The three-dimensional parts are set, for convenience, to be non-contact surfaces, and hence, a no-current boundary condition is imposed.

$$\frac{\partial \phi_u(r)}{\partial n} = 0, \forall r \in S_{3D} \tag{2.29}$$

As before, the bottom is allowed to be either a floating or grounded backplane.

2.3 Equation summary

2.3.1 Planar problem

For this problem we have six unknowns and six equations.

The six unknowns are:

$$\phi_u(r), \frac{\partial \phi_u(r)}{\partial n}, \forall r \in S_{top} \tag{2.30}$$

$$\phi_u(r), \frac{\partial \phi_u(r)}{\partial n}, \forall r \in S_{mid} \tag{2.31}$$

$$\phi_l(r), \frac{\partial \phi_l(r)}{\partial n}, \forall r \in S_{bot}. \tag{2.32}$$

$$\tag{2.33}$$

The six equations are:

$$a(r)\phi_u(r) + b(r)\frac{\partial \phi_u(r)}{\partial n} = f(r), \forall r \in S_{top} \tag{2.34}$$

$$c\phi_l(r) + d\frac{\partial \phi_l(r)}{\partial n} = 0, \forall r \in S_{bot} \tag{2.35}$$

$$\begin{aligned}\phi_u(r) = &- \int_{S_{top}} G(r,r')\frac{\partial \phi_u(r')}{\partial n} dr' - \int_{S_{mid}} G(r,r')\frac{\partial \phi_u(r')}{\partial n} dr' \\ &+ \int_{S_{top}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr', \forall r \in S_{top}\end{aligned} \tag{2.36}$$

$$\phi_u(r) = -\int_{S_{top}} G(r,r')\frac{\partial \phi_u(r')}{\partial n}dr' - \int_{S_{mid}} G(r,r')\frac{\partial \phi_u(r')}{\partial n}dr'$$
$$+\int_{S_{top}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr', \forall r \in S_{mid} \qquad (2.37)$$

$$\phi_u(r) = \alpha\int_{S_{mid}} G(r,r')\frac{\partial \phi_u(r')}{\partial n}dr' + \int_{S_{bot}} G(r,r')\frac{\partial \phi_l(r')}{\partial n}dr'$$
$$-\int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' - \int_{S_{bot}} \frac{\partial G(r,r')}{\partial n}\phi_l(r')dr', \forall r \in S_{mid} \qquad (2.38)$$

$$\phi_l(r) = \alpha\int_{S_{mid}} G(r,r')\frac{\partial \phi_u(r')}{\partial n}dr' + \int_{S_{bot}} G(r,r')\frac{\partial \phi_l(r')}{\partial n}dr'$$
$$-\int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' - \int_{S_{bot}} \frac{\partial G(r,r')}{\partial n}\phi_l(r')dr', \forall r \in S_{bot}. \qquad (2.39)$$

2.3.2 Three-dimensional problem

We now have eight unknowns and eight equations for the three-dimensional problem.

The eight unknowns are:

$$\phi_u(r), \frac{\partial \phi_u(r)}{\partial n}, \forall r \in S_{top'} \qquad (2.40)$$
$$\phi_u(r), \frac{\partial \phi_u(r)}{\partial n}, \forall r \in S_{mid} \qquad (2.41)$$
$$\phi_l(r), \frac{\partial \phi_l(r)}{\partial n}, \forall r \in S_{bot} \qquad (2.42)$$
$$\phi_u(r), \frac{\partial \phi_u(r)}{\partial n}, \forall r \in S_{3D}. \qquad (2.43)$$

The eight equations, ordered in this fashion for later convenience, are:

$$a(r)\phi_u(r) + b(r)\frac{\partial \phi_u(r)}{\partial n} = f(r), \forall r \in S_{top'} \qquad (2.44)$$

$$c\phi_l(r) + d\frac{\partial \phi_l(r)}{\partial n} = 0, \forall r \in S_{bot} \qquad (2.45)$$

$$
\begin{aligned}
\frac{\phi_u(r)}{2} = &-\int_{S_{top'}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{3D}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' \\
&+\int_{S_{top'}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{3D}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr', \forall r \in S_{top'}
\end{aligned}
\tag{2.46}
$$

$$
\begin{aligned}
\frac{\phi_u(r)}{2} = &-\int_{S_{top'}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{3D}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' \\
&+\int_{S_{top'}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{3D}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr', \forall r \in S_{mid}
\end{aligned}
\tag{2.47}
$$

$$
\begin{aligned}
\phi_u(r) = &\alpha\int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' + \int_{S_{bot}} G(r,r')\frac{\partial\phi_l(r')}{\partial n}dr' \\
&-\int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' - \int_{S_{bot}} \frac{\partial G(r,r')}{\partial n}\phi_l(r')dr', \forall r \in S_{mid}
\end{aligned}
\tag{2.48}
$$

$$
\begin{aligned}
\phi_l(r) = &\alpha\int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' + \int_{S_{bot}} G(r,r')\frac{\partial\phi_l(r')}{\partial n}dr' \\
&-\int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' - \int_{S_{bot}} \frac{\partial G(r,r')}{\partial n}\phi_l(r')dr', \forall r \in S_{bot}
\end{aligned}
\tag{2.49}
$$

$$
\begin{aligned}
\frac{\phi_u(r)}{2} = &-\int_{S_{top'}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{3D}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' - \int_{S_{mid}} G(r,r')\frac{\partial\phi_u(r')}{\partial n}dr' \\
&+\int_{S_{top'}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{3D}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr' + \int_{S_{mid}} \frac{\partial G(r,r')}{\partial n}\phi_u(r')dr', \forall r \in S_{3D}
\end{aligned}
\tag{2.50}
$$

$$
\frac{\partial\phi_u(r)}{\partial n} = 0, \forall r \in S_{3D} \tag{2.51}
$$

In the next chapter, we will develop the discretization of these equations; this method will yield an equivalent matrix equation describing the discretized problem.

Chapter 3

Problem discretization

In this chapter, we begin with the planar problem. We introduce the meshing of this geometry, as well as the basis functions used to expand $\phi(r)$ and $\frac{\partial\phi(r)}{\partial n}$ on the surfaces. We discretize our system of equations into an analagous matrix form using these expansions and a centroid-collocation scheme. We subsequently develop the modifications of this procedure needed to model three-dimensional features. The chapter ends with a discussion of the iterative method that will be used to solve the system matrix equation.

3.1 Discretization of planar problem

3.1.1 Geometric mesh and basis functions

The top, middle and bottom surface layers are each discretized with an N_x x N_y grid of regular, rectangular panels, generating a total number of panels, $N_p = 3N_xN_y = 3N_{ppl}$. Such a meshing is shown in Figure 3-1. We label each panel with an integer index, which is shown as a subscript, and with a superscript label to indicate which layer the panel resides on. The integer indexing is done sequentially along the x-direction first, then the y-direction.

Each panel provides the local support for a piece-wise constant basis function,$g_i^{layer}(r)$,

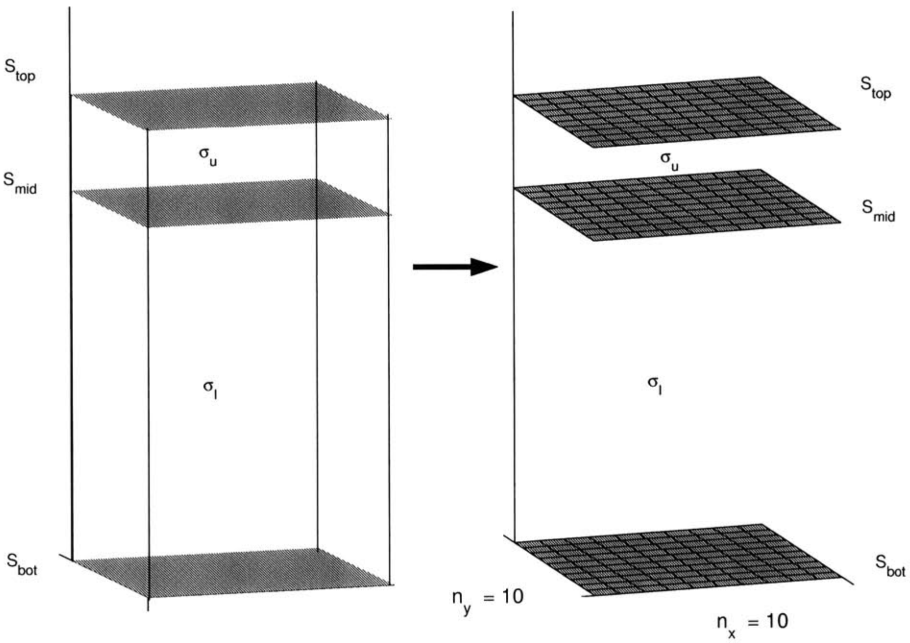
S_{top}
S_{mid}
S_{bot}
σ_u
σ_l
S_{top}
σ_u
S_{mid}
σ_l
S_{bot}
$n_y = 10$
$n_x = 10$

Figure 3-1: Problem discretization for planar problem

as defined in equation 3.1, where the symbol $\diamondsuit_i^{layer}$ indicates the panel indexed by integer i on the layer labelled by *layer*.

$$\begin{aligned} g_i^{layer}(r) &= 1, r \in \diamondsuit_i^{layer} \\ g_i^{layer}(r) &= 0, otherwise \end{aligned} \tag{3.1}$$

On each surface we expand the unknowns $\phi(r)$ and $\frac{\partial\phi(r)}{\partial n}$ in terms of this basis. This is shown in equations 3.2 - 3.7. Notice that our choice assumes that a constant basis is a good representation for the potential and the conductive currents. For this reason, the discretization must be fine enough to ensure this hypothesis is correct. Therefore we must ensure that for a given frequency, the panel size is on the order of $\lambda/10$ or smaller.

$$\phi_u(r) = \sum_{i=1}^{N_{ppl}} p_i^{top} g_i^{top}(r), r \in S_{top} \tag{3.2}$$

$$\phi_u(r) = \sum_{i=1}^{N_{ppl}} p_i^{mid} g_i^{mid}(r), r \in S_{mid} \tag{3.3}$$

$$\phi_l(r) = \sum_{i=1}^{N_{ppl}} p_i^{bot} g_i^{bot}(r), r \in S_{bot} \tag{3.4}$$

$$\frac{\partial\phi_u(r)}{\partial n} = \sum_{i=1}^{N_{ppl}} q_i^{top} g_i^{top}(r), r \in S_{top} \tag{3.5}$$

$$\frac{\partial\phi_u(r)}{\partial n} = \sum_{i=1}^{N_{ppl}} q_i^{mid} g_i^{mid}(r), r \in S_{mid} \tag{3.6}$$

$$\frac{\partial\phi_l(r)}{\partial n} = \sum_{i=1}^{N_{ppl}} q_i^{bot} g_i^{bot}(r), r \in S_{bot} \tag{3.7}$$

The problem is now reduced to finding the six unknown weight vectors of our expansions: p^{top}, p^{mid} , p^{bot}, q^{top}, q^{mid}, q^{bot}(we drop the subscript to denote the vector of weights). In the next section, we derive the matrix equation that governs these unknowns.

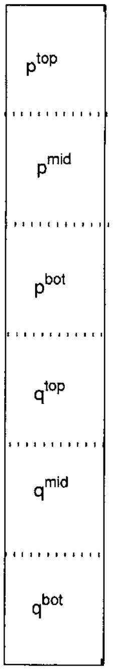

Figure 3-2: Unknown vector for planar problem

3.1.2 System matrix equation

To complete the descretized description, we insert the discretized representation of $\phi(r)$ and $\frac{\partial\phi(r)}{\partial n}$ into equations 2.34 - 2.39, and test at the centroids of the panels, which yields the analogous matrix equation

$$Ax = b. \tag{3.8}$$

The unknown solution vector sought, x, is a column vector, of size $N = 6N_{ppl}$, which has a blocked row structure. Each block row of size N_{ppl} corresponds to an unknown weight vector. This blocked structure is shown in Figure 3-2. The right-hand side, b, is all zeros, except for the first block row, of size N_{ppl}, which contains non-zeros on the panels that have a non-zero applied voltage.

Figure 3-3 shows the block pattern of our system matrix A. Note that each sub-matrix block indicated in the figure is of size $N_{ppl} \times N_{ppl}$. The first two block rows of the matrix correspond to the boundary conditions tested at S_{top} and S_{bot}. The third

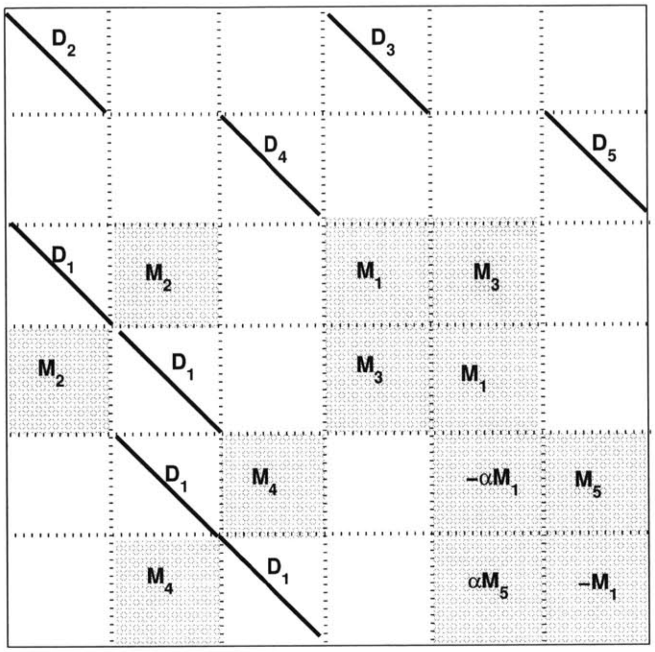

Figure 3-3: System matrix structure for the planar problem

through sixth block rows correspond to testing the surface integral equations 2.36 - 2.39, respectively.

In this block structure, the matrix entries are given by:

$$M_1(i,j) = -\int_{r' \in \Diamond_j^{top}} G(\breve{r}_i^{top}, r')dr' = -\int_{r' \in \Diamond_j^{mid}} G(\breve{r}_i^{mid}, r')dr' \\ = -\int_{r' \in \Diamond_j^{bot}} G(\breve{r}_i^{bot}, r')dr' \tag{3.9}$$

$$M_2(i,j) = \int_{r' \in \Diamond_j^{top}} \frac{\partial G(\breve{r}_i^{mid}, r')}{\partial n} dr' = \int_{r' \in \Diamond_j^{mid}} \frac{\partial G(\breve{r}_i^{top}, r')}{\partial n} dr' \tag{3.10}$$

$$M_3(i,j) = -\int_{r' \in \Diamond_j^{top}} G(\breve{r}_i^{mid}, r')dr' = -\int_{r' \in \Diamond_j^{mid}} G(\breve{r}_i^{top}, r')dr' \tag{3.11}$$

$$M_4(i,j) = -\int_{r' \in \Diamond_j^{mid}} \frac{\partial G(\breve{r}_i^{bot}, r')}{\partial n} dr' = -\int_{r' \in \Diamond_j^{bot}} \frac{\partial G(\breve{r}_i^{mid}, r')}{\partial n} dr' \tag{3.12}$$

$$M_5(i,j) = \int_{r' \in \Diamond_j^{mid}} G(\breve{r}_i^{bot}, r')dr' = \int_{r' \in \Diamond_j^{bot}} G(\breve{r}_i^{mid}, r')dr' \tag{3.13}$$

In these equations, the notation $\breve{r}_i$ indicates the centroid of the ith panel.

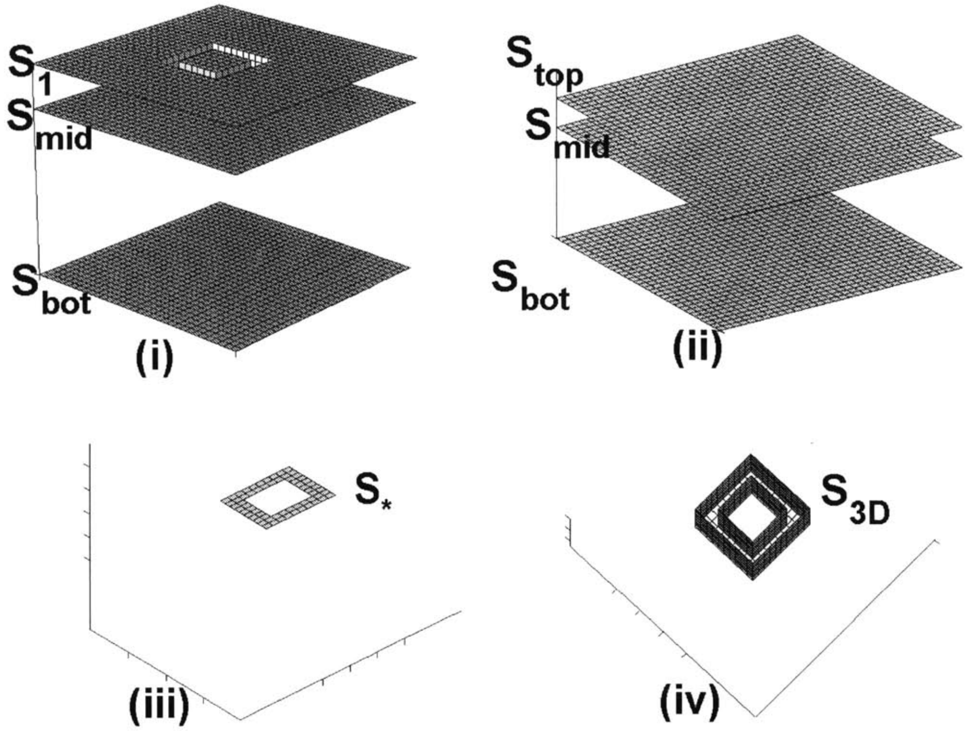

Figure 3-4: Breakdown of three-dimensional problem geometry into different components

3.2 Discretization of three-dimensional problem

3.2.1 Geometric mesh and basis functions

We again begin by discretizing the mesh, as shown in Figure 3-4 (i). As in Chapter 2, we break down the top boundary surface, S_1, as a combination of three surfaces, $S_\star$, S_{3D} and S_{top}, and define $S_{top'} = S_{top} \setminus S_\star$. The meaning of S_{mid} and S_{bot} remain the same as in the planar problem.

We will label all of the panels on S_{top}, S_{mid} and S_{bot} as before, with a layer superscript and an index subscript from 1 to N_{ppl}, with the ordering done consistently along the x-direction first, then the y-direction. Panels on surface S_{3D} will be denoted with a superscript $3D$. The number and ordering of these additional three-dimensional panels is decided by the the mesher, and this ordering, once chosen, does not affect the problem. We will denote the total number of three-dimensional panels as N_{3D}.

Each panel on S_{top}, S_{mid}, S_{bot} and S_{3D} is assigned a piecewise constant basis function, defined as in equation 3.1 .

Our unknowns are the potential and its normal derivative on the surfaces $S_{top'}$, S_{mid}, S_{bot} and S_{3D}. We expand these unknowns as follows.

$$\phi(r) = \sum_{i=1, \Diamond_i \in S_{top'}}^{N_{ppl}} p_i^{top} g_i^{top}(r), r \in S_{top'} \tag{3.14}$$

$$\phi(r) = \sum_{i=1}^{N_{ppl}} p_i^{mid} g_i^{mid}(r), r \in S_{mid} \tag{3.15}$$

$$\phi(r) = \sum_{i=1}^{N_{ppl}} p_i^{bot} g_i^{bot}(r), r \in S_{bot} \tag{3.16}$$

$$\phi(r) = \sum_{i=1}^{N_{3D}} p_i^{3D} g_i^{3D}(r), r \in S_{3D} \tag{3.17}$$

$$\frac{\partial \phi(r)}{\partial n} = \sum_{i=1, \Diamond_i \in S_{top'}}^{N_{ppl}} q_i^{top} g_i^{top}(r), r \in S_{top'} \tag{3.18}$$

$$\frac{\partial \phi(r)}{\partial n} = \sum_{i=1}^{N_{ppl}} q_i^{mid} g_i^{mid}(r), r \in S_{mid} \tag{3.19}$$

$$\frac{\partial \phi(r)}{\partial n} = \sum_{i=1}^{N_{ppl}} q_i^{bot} g_i^{bot}(r), r \in S_{bot} \tag{3.20}$$

$$\frac{\partial \phi(r)}{\partial n} = \sum_{i=1}^{N_{ppl}} q_i^{3D} g_i^{3D}(r), r \in S_{3D} \tag{3.21}$$

The problem is now reduced to finding the eight unknown weight vectors of our expansions : $p^{top'}$, p^{mid}, p^{bot}, p^{3D}, $q^{top'}$, q^{mid}, q^{bot}, q^{3D}. We use $p^{top'}$ and $q^{top'}$ to denote the weight vectors for panels in the intersection of S_{top} and $S_{top'}$. For elements in these vectors, both of length $N_{top'} < N_{ppl}$, the index, i given to the element may not correspond to its position within the vector. The vectors p_{top} and q_{top}, both of length N_{ppl}, have the same meaning as it did in the planar problem. These choices will be motivated in greater detail in the next chapter.

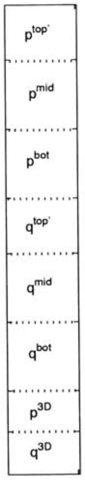

Figure 3-5: Unknown vector for three-dimensional problem

3.2.2 System matrix equation

We now plug our expanded unknowns into equations 2.44 - 2.51 and test the equations at the centroids of the panels. This process again yields a matrix equation, given as

$$\tilde{A}\tilde{x} = \tilde{b}. \tag{3.22}$$

We use the tildes to differentiate from the variables used in the planar matrix equation, which were of a different size and structure.

The unknown solution vector sought, $\tilde{x}$, is a column vector, of size $\tilde{N} = 2N_{top'} + 4N_{ppl} + 2N_{3D}$, that can be broken into blocks corresponding to each of the unknown weight vectors. This blocked structure is shown in Figure 3-5.

Figure 3-6 shows the block pattern of our system matrix $\tilde{A}$. The first two block rows of the matrix correspond to the boundary conditions tested at the $S_{top'}$ and S_{bot}. The third through seventh block rows correspond to testing the surface integral equations 2.46-2.50, respectively. The final row correponds to discretizing equation 2.51, the boundary condition at the surface S_{3D}.

In this block structure, the matrix entries are given by:

$$M_1(i,j) = -\int_{r' \in \Diamond_j^{top}} G(\check{r}_i^{top}, r')dr' = -\int_{r' \in \Diamond_j^{mid}} G(\check{r}_i^{mid}, r')dr'$$

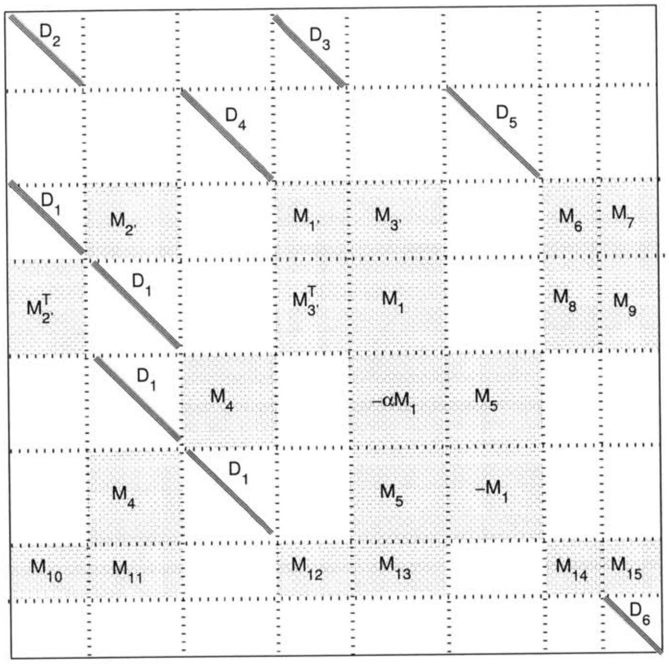

Figure 3-6: System matrix structure for the three-dimensional problem

$$= -\int_{r'\in\Diamond_j^{bot}} G(\breve{r}_i^{bot}, r')dr' \tag{3.23}$$

$$M_{1'}(i,j) = M_1(i,j) \tag{3.24}$$

$$M_2(i,j) = \int_{r'\in\Diamond_j^{top}} \frac{\partial G(\breve{r}_i^{mid}, r')}{\partial n}dr' = \int_{r'\in\Diamond_j^{mid}} \frac{\partial G(\breve{r}top_i, r')}{\partial n}dr' \tag{3.25}$$

$$M_{2'}(i,j) = M_2(i,j) \tag{3.26}$$

$$M_3(i,j) = -\int_{r'\in\Diamond_j^{top}} G(\breve{r}_i^{mid}, r')dr' = -\int_{r'\in\Diamond_j^{mid}} G(\breve{r}_i^{top}, r')dr' \tag{3.27}$$

$$M_{3'}(i,j) = M_3(i,j) \tag{3.28}$$

$$M_4(i,j) = -\int_{r'\in\Diamond_j^{mid}} \frac{\partial G(\breve{r}_i^{bot}, r')}{\partial n}dr' = -\int_{r'\in\Diamond_j^{bot}} \frac{\partial G(\breve{r}_i^{mid}, r')}{\partial n}dr' \tag{3.29}$$

$$M_5(i,j) = \int_{r'\in\Diamond_j^{mid}} G(\breve{r}_i^{bot}, r')dr' = \int_{r'\in\Diamond_j^{bot}} G(\breve{r}_i^{mid}, r')dr' \tag{3.30}$$

$$M_6(i,j) = \int_{r'\in\Diamond_j^{3D}} \frac{\partial G(\breve{r}_i^{top}, r')}{\partial n}dr' \tag{3.31}$$

$$M_7(i,j) = -\int_{r'\in\Diamond_j^{3D}} G(\breve{r}_i^{top}, r')dr' \tag{3.32}$$

$$M_8(i,j) = \int_{r'\in\Diamond_j^{3D}} \frac{\partial G(\breve{r}_i^{mid}, r')}{\partial n}dr' \tag{3.33}$$

$$M_9(i,j) = -\int_{r' \in \Diamond_j^{3D}} G(\breve{r}_i^{mid}, r') dr' \tag{3.34}$$

$$M_{10}(i,j) = \int_{r' \in \Diamond_j^{top}} \frac{\partial G(\breve{r}_i^{3D}, r')}{\partial n} dr' \tag{3.35}$$

$$M_{11}(i,j) = \int_{r' \in \Diamond_j^{mid}} \frac{\partial G(\breve{r}_i^{3D}, r')}{\partial n} dr' \tag{3.36}$$

$$M_{12}(i,j) = -\int_{r' \in \Diamond_j^{top}} G(\breve{r}_i^{3D}, r') dr' \tag{3.37}$$

$$M_{13}(i,j) = -\int_{r' \in \Diamond_j^{mid}} G(\breve{r}_i^{3D}, r') dr' \tag{3.38}$$

$$M_{14}(i,j) = \int_{r' \in \Diamond_j^{3D}} \frac{\partial G(\breve{r}_i^{3D}, r')}{\partial n} dr' \tag{3.39}$$

$$M_{15}(i,j) = -\int_{r' \in \Diamond_j^{3D}} G(\breve{r}_i^{3D}, r') dr' \tag{3.40}$$

3.3 Iterative method for matrix solution

Both the planar and three-dimensional matrix equations are solved using the Generalized Conjugate Residual(GCR) algorithm, a Krylov-subspace based iterative method that requires only one matrix-vector product at each iteration. In the next chapter, we will discuss the details of accelerating this matrix-vector product, and show that its memory usage is $O(N)$ and its computational complexity is $O(Nlog(N))$.

However, two things must be noticed regarding the expense of the iterative method. First, we note that for each iteration, two vectors of size $O(N)$ must be stored, meaning that the memory required by the iterative method grows linearly with the number of iterations, k. This problem could be avoided using a transpose-free quasi-minimal residual method (QMR), however, for this thesis, we restrict our attention to the GCR method.

Second, the computational time required for a solution grows with the number of iterations, k. For these two reasons we will need to ensure that the number of iterations does not grow rapidly with problem size, or k is $O(1)$; otherwise, the problem will become prohibitively expensive to solve.

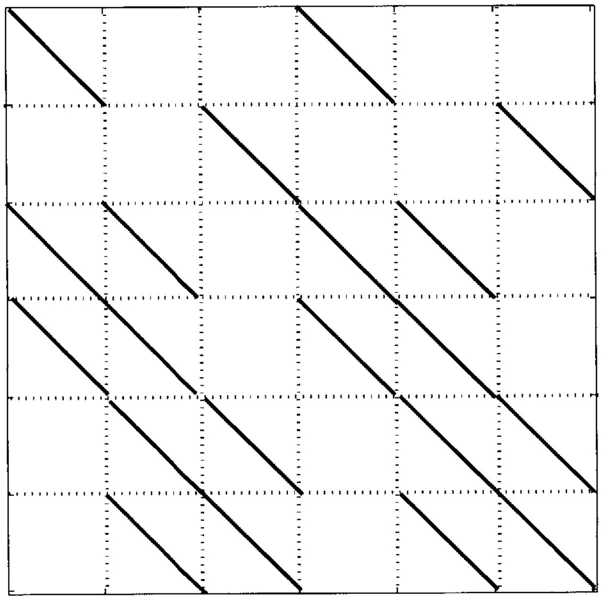

Figure 3-7: Preconditioner sparsity pattern for planar problem

We ensure this rapid convergence by using a block-diagonal right preconditioner. The sparsity pattern of the planar problem preconditioner is shown in Figure 3-7, and the three-dimensional problem preconditioner is shown in Figure 3-8. Note that the memory usage and computational effort needed to store and calculate the LU factorization of this matrix P is minimal, because of its sparsity; this result will be shown later in the final chapter.

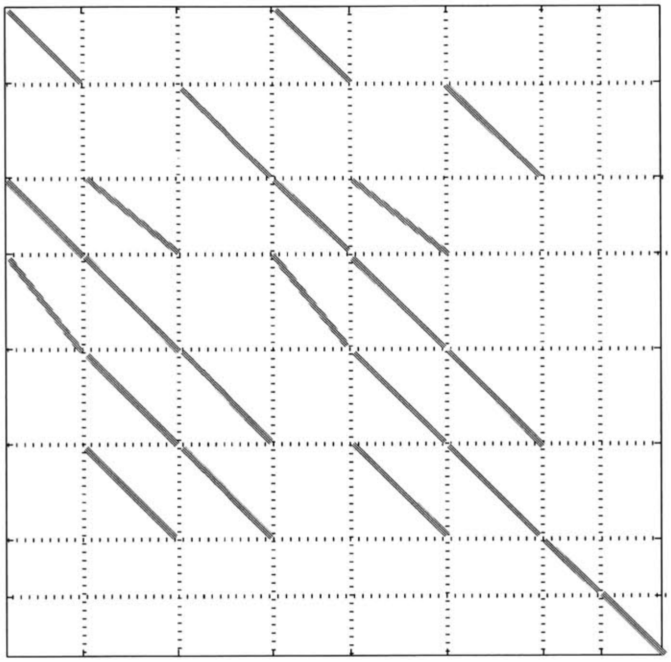

Figure 3-8: Preconditioner sparsity pattern for three-dimensional problem

Chapter 4

FFT acceleration of matrix-vector products

As discussed in the previous chapter, our discretized coupled integral equations give rise to a matrix equation that we will solve using a Krylov-subspace iterative method, GCR. At the heart of this solver is the need for a fast matrix-vector product. We will see in this chapter that the matrix is never explicity formed, but rather, because of the regular grid and translation-invariance of the Green's function, we can use FFTs to accelerate the planar problem's matrix-vector product, and eliminate the need for storing all N^2 matrix entries. We then demonstrate how this idea can be adapted with those of the pFFT method in order to accelerate the matrix-vector product of the three-dimensional problem's matrix-vector product.

4.1 Planar problem

Our goal stated goal for this section is: given a vector v, return $w = Av$ efficiently. We will approach this goal by breaking down the matrix-vector product into submatrix-subvector products. We then discuss the acceleration of one of these submatrix-subvector products using FFTs. Finally, we will give a quasicode description of the

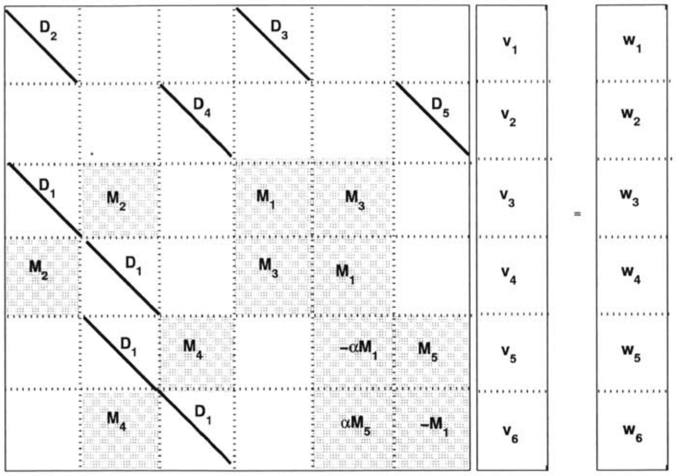
D2
D3
v1
w1
D4
D5
v2
w2
D1
M2
M1
M3
v3
=
w3
M2
D1
M3
M1
v4
w4
D1
M4
−αM1
M5
v5
w5
M4
D1
αM5
−M1
v6
w6

Figure 4-1: Matrix-vector product in block form, for the planar problem

overall matrix-vector product.

4.1.1 Decomposition of matrix-vector product

We approach the matrix-vector product by breaking down the matrix and input vector into blocks, as shown in Figure 4-1. This will allow us to consider the matrix-vector product as a composition of many submatrix-subvector products.

We can rewrite the result of this matrix-multiplication as:

$$w_1 = D_2 v_1 + D_3 v_4 \tag{4.1}$$

$$w_2 = D_4 v_3 + D_5 v_6 \tag{4.2}$$

$$w_3 = D_1 v_1 + M_2 v_2 + M_1 v_4 + M_3 v_5 \tag{4.3}$$

$$w_4 = M_2 v_1 + D_1 v_2 + M_3 v_4 + M_1 v_5 \tag{4.4}$$

$$w_5 = D_1 v_2 + M_4 v_3 - \alpha M_1 v_5 + M_5 v_6 \tag{4.5}$$

$$w_6 = M_4 v_2 + D_1 v_3 + \alpha M_5 v_6 - M_1 v_6 \tag{4.6}$$

$$\tag{4.7}$$

Calculating w_1 and w_2 are very simple as these results involve only submatrix-

4.2 Three-dimensional problem

We now tackle the more complex matrix-vector product associated with the three-dimensional problem. We again approach the problem by decomposing the matrix-vector product into different submatrix-subvector products. We will first look at the submatrix-subvector products which describe interactions between planar source panels and test points on the planar panels. We'll see that a simple modification of our previous algorithm can handle these parts of the matrix-vector product. Next, we will look at the parts of the matrix-vector product that involve panels on the S_{3D} surface, either as source panels, test points, or both. We'll present a modified version of the precorrected FFT method that can be used to calculate such submatrix-subvector products. Finally, we will present the overall matrix-vector product algorithm used for the three-dimensional problem.

4.2.1 Decomposition of matrix-vector product

We show the matrix-vector product, $\tilde{A}v' = w'$ as a composition of blocks in Figure 4-2. The sections corresponding to planar to planar interactions are demarcated with a bold line; everything above this line indicates a part of the matrix-vector product which involves a planar source panel scattered to a test centroid on a planar panel. We will consider this part of the matrix-vector product first. Then we will discuss the parts of the matrix-vector product below this line, which correspond to scattering involving panels on S_{3D}. These parts of the matrix will be handled using an adaptation of the precorrected-FFT method.

4.2.2 Accelerating planar to planar scattering

We will make use of our previously developed planar matrix-vector black box to handle planar panel to planar panel interactions. We pad the vectors v_1' and v_4' with zeros at those indices that correspond to the "deleted panels" on the surface $S_\star$ to form v_1 and

Once these values have been precomputed, we can implement a black-box matrix-vector code, which takes in an input vector v, and returns the matrix-vector product w. Notice that the inverse DFT matrices are factored out of each submatrix-subvector product, and applied only once to each block row in step 3.

Matrix-vector black box

1. Calculate $\tilde{V}_1, \tilde{V}_2, \tilde{V}_3, \tilde{V}_4, \tilde{V}_5, \tilde{V}_6$ by reshaping input subvectors and applying two-dimensional FFTs.

2. Compute $\tilde{W}_3, \tilde{W}_4, \tilde{W}_5, \tilde{W}_6$ defined as:

- **a.** $\tilde{W}_3 = \tilde{H}_2. * \tilde{V}_2 + \tilde{H}_1. * \tilde{V}_4 + \tilde{H}_3. * \tilde{V}_5$
- **b.** $\tilde{W}_4 = \tilde{H}_2. * \tilde{V}_1 + \tilde{H}_3. * \tilde{V}_4 + \tilde{H}_1. * \tilde{V}_5$
- **c.** $\tilde{W}_5 = \tilde{H}_4. * \tilde{V}_3 - \alpha \tilde{H}_1. * \tilde{V}_5 + \tilde{H}_5. * \tilde{V}_6$
- **d.** $\tilde{W}_6 = \tilde{H}_4. * \tilde{V}_2 + \tilde{H}_5. * \tilde{V}_6 + \tilde{H}_1. * \tilde{V}_6$

3. Compute W_3, W_4, W_5, W_6 using two-dimensional inverse FFTs:

4. Lift out the relevant parts of W_3, W_4, W_5, W_6, and reshape into vectors, w_3, w_4, w_5, w_6..

5. Add contributions due to diagonal submatrix, D_1.

- **a.** $w_3 + = D_1 v_1$
- **b.** $w_4 + = D_1 v_2$
- **c.** $w_5 + = D_1 v_2$
- **d.** $w_6 + = D_1 v_3$

6. Compute simple diagonal submatrix-subvector products to find $w_1 = D_2 v_1 + D_3 v_4, w_2 = D_4 v_3 + D_5 v_6$.

2. Zero pad and reshape the subvector v_4 into a matrix, V_4. Calculate its two-dimensional FFT $\tilde{V}_4 = F_{N_x} V_4 F_{N_y}$.

3. Component-wise multiply $\tilde{H}_1$ and $\tilde{V}_4$.

4. Calculate the inverse two-dimensional FFT of the result in 3.

5. Keep the $N_x \times N_y$ submatrix that corresponds to test points on the grid (the results of the convolution will extend past the test points of interest) and reshape these values into a vector of size $N_x N_y$.

The main computational expense lies in the two-dimensional transforms, each with a cost of $O(N_x N_y log(N_x N_y))$. This is much cheaper than the original expense of $O((N_x N_y)^2)$. Now we have an efficient means of computing the dense submatrix-subvector products in our planar problem.

4.1.3 Accelerated matrix-vector product

In order to compute the entire matrix-vector product efficiently, we must compute each of the dense submatrix-subvector products using the previously introduced approach.

For each of these dense submatrices, we need only its corresponding kernel matrix in the discrete fourier domain. Therefore, we will compute $\tilde{H}_1$,$\tilde{H}_2$,$\tilde{H}_3$,$\tilde{H}_4$,$\tilde{H}_5$ just once and store them, in lieu of computing and storing each element of the dense submatrices. The cost for one such computation is only $O(N_x N_y)$ to fill the kernel matrix, and $O(N_x N_y log(N_x N_y))$ to compute its frequency-domain description. The memory cost is $O(N_x N_y)$ to store the entries associated with this $2N_x \times 2N_y$ size matrix. This is much cheaper than the original cost of calculating and then storing a dense matrix, both of which require $O((N_x N_y)^2)$ complexity. We will also precompute the entries of the diagonal matrices, at a cost of $O((N_x N_y)$ memory storage.

subvector products with diagonal matrices, and hence can be done with $O(N)$ complexity and memory. However, all of the other components of w require three dense submatrix-subvector products that require $O(N^2)$ computational cost, if unaccelerated. However, although each of the submatrices involved is dense, each submatrix-subvector product can be thought of a as a discrete two-dimensional convolution. This is a result of the shift invariance of the two kernels $G_o(r, r')$ and $\frac{\partial G_o(r,r')}{\partial n}$, as well as the regular gridding of all of the surfaces. We discuss in the next section how to accelerate one such dense submatrix-subvector product.

4.1.2 Accelerating dense submatrix-subvector products

In this section we will consider the acceleration of one submatrix-subvector product. For illustrative purposes, we choose the product $M_1 v_4$. This submatrix-subvector product can be thought of as the result of a discrete 2D convolution between a $2N_x \times 2N_y$ "kernel matrix" H_1 and a $2N_x \times 2N_y$ zero-padded version of the subvector V_4. This is a result of the shift-invariance of the kernel and the regular grid. As presented in [7], such a convolution of two matrices can be done very inexpensively.

The entries of the kernel matrix are the $N_x N_y$ unique values of M_1 arranged in such a fashion as to represent the two-dimensional discrete "impulse response" due to a uniform source panel.

To convolve the two matrices H_{mat} and V_{mat}, we can use the following result from [7].

$$conv(H_1, V_4) = F_{N_x}^{-1}((F_{N_x} H_1 F_{N_y}) .* (F_{N_x} V_4 F_{N_y}))F_{N_y}^{-1} = F_{N_x}^{-1}(\tilde{H}_1 .* \tilde{V}_4)F_{N_y}^{-1} \qquad (4.8)$$

In this equation, F_n is the one-dimensional discrete Fourier transform (DFT) matrix of size n, and F_n^{-1} its inverse transform. Applying these matrices can be done very cheaply using the Fast Fourier Transform (FFT) algorithm. Therefore, given H_1, we need only do the following:

1. Calculate the two-dimensional FFT of H_1: $\tilde{H}_1 = F_{N_x} H_1 F_{N_y}$.

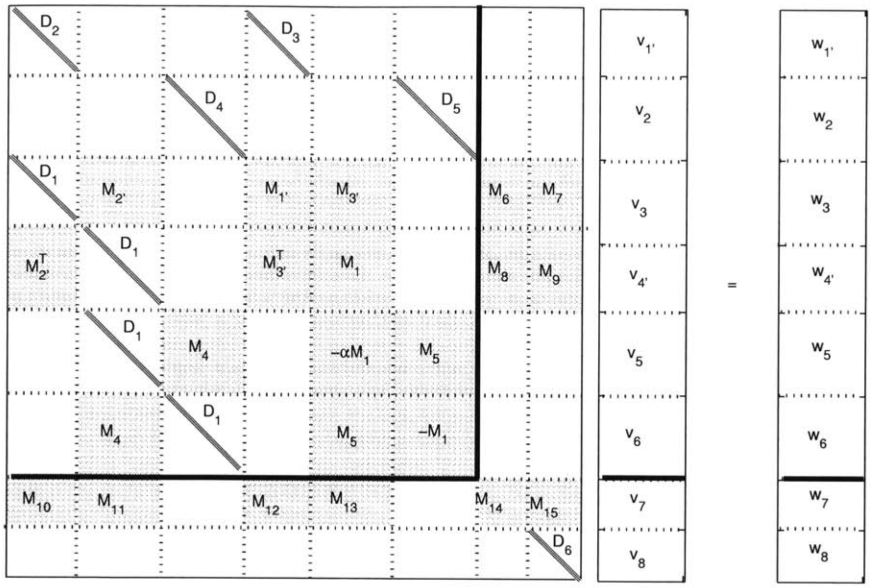
D2
D3
D4
D5
D1
M2'
M1'
M3'
M6
M7
M2'T
D1
M3'T
M1
M8
M9
D1
M4
−αM1
M5
D1
M4
M5
−M1
M10
M11
M12
M13
M14
M15
D6
v1'
v2
v3
v4'
v5
v6
v7
v8
=
w1'
w2
w3
w4'
w5
w6
w7
w8

Figure 4-2: Matrix-vector product in block form, for the three-dimensional problem

v_4, respectively. We then define v as the concatenation of vectors $v_1, v_2, v_3, v_4, v_5, v_6$. We apply the previously developed matrix-vector black-box to this vector to find the vector $w^{planar} = Av$. Later, we will retain only those elements of w_1^{planar} and w_4^{planar} that correspond to testing at panels on $S_{top'}$, that is $w_1^{planar'}$ and $w_4^{planar'}$.

4.2.3 Accelerating three-dimensional interactions using pFFT

We now have to consider the remaining portions of the matrix-vector product. We will consider three different sections, each of which will be handled in a slightly different fashion.

Three-dimensional panel to planar panel scattering via projection

The first part we will consider are the contributions of sources on S_{3D} to test points on $S_{top'}$ and S_{mid}. We will denote these contributions as w_3^{3D} and w_4^{3D}, respectively, and they are given by:

$$w_3^{3D} = M_7 v_7 + M_8 v_8 \tag{4.9}$$

and

$$w_4^{3D} = M_9 v_7 + M_{10} v_8. \tag{4.10}$$

To accelerate the dense submatrix-subvector products involved in these terms, we will borrow the concept of projection from the pFFT method. We first divide the planar top and mid surface uniform grids (S_{top} and S_{mid}) into pFFT cells. Each cell contains a $k \times k$ block of panels on S_{top} and another $k \times k$ block of panels on S_{mid}.

For each three-dimensional panel source, either a monopole source in v_8 or a dipole source in v_7, we will find equivalent uniform monopole sources on the planar panels in the enclosing cell. We choose the weights of these planar sources such that the field is approximated well for points outside of the cell. These weights are chosen by matching the original field due to the three-dimensional source with the field due to the projected sources. This matching is done at at points on a collocation sphere. Reference [2] provides an in-depth explanation of this process.

We show in Figure 4-3 a three-dimensional panel, as well as the surrounding planar panels which comprise its cell. The cell size shown is $k = 4$. The collocation points are also shown in this picture.

This projection from three-dimensional sources to planar sources can be written in terms of four projection matrices. While these matrices are never explicitly formed, it is a useful device to write the projection in this manner.

$$v_4^{proj} = -P_{top}^{proj,d} v_7 + P_{top}^{proj,m} v_8 \tag{4.11}$$

$$v_5^{proj} = -P_{mid}^{proj,d} v_7 + P_{mid}^{proj,m} v_8 \tag{4.12}$$

The process of computing these four projection matrices is the same as the procedure outlined in [2], although here we use uniform sources on the grid panels, instead of grid point charges. As outlined in this reference, the cost of projecting a vector of three-dimensional sources is $O(N_{3D})$. The projection also requires a one-time $O(N_{3D})$ setup cost.

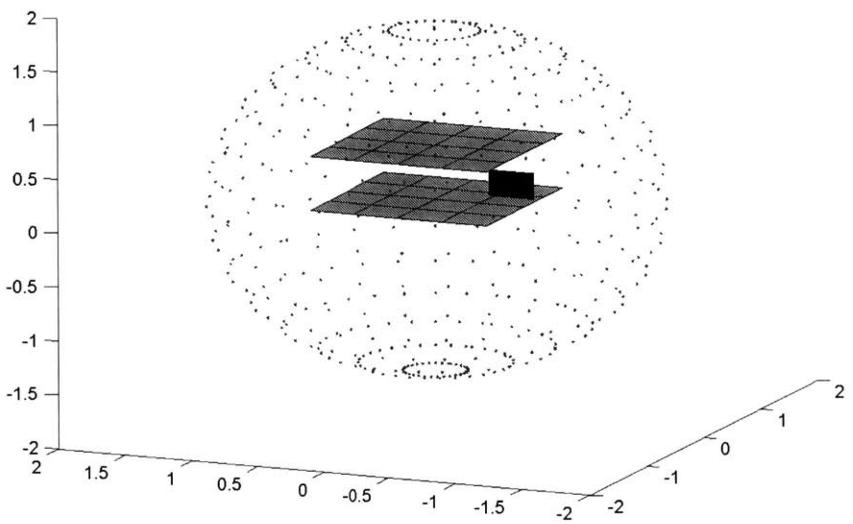

Figure 4-3: Three dimensional panel within its pFFT cell with surrounding collocation sphere

Once these projected source vectors are formed, we can scatter them to planar panel centroids in the same manner as before: by using the FFT to accelerate the scattering. Finally, the near-field interactions of three-dimensional sources to nearby planar panels must be done directly. We add precorrection matrices that eliminate the effect of projected sources and add in this direct calculation. These correction matrices are indicated with a C.

The final equations which describe the three-dimensional to planar scattering are then given by:

$$w_3^{3D} = M_1 v_4^{proj} + M_3 v_5^{proj} + C_{proj,d}^{top} v_7 + C_{proj,m}^{top} v_8 \tag{4.13}$$

and

$$w_4^{3D} = M_3 v_4^{proj} + M_1 v_5^{proj} + C_{proj,d}^{mid} v_7 + C_{proj,m}^{mid} v_8. \tag{4.14}$$

Three-dimensional panel to planar panel scattering via projection

We now work with the parts of the matrix-vector product that correspond to planar sources scattering to the test points on S_{3D}. This part of the product is given as:

$$w_7^{planar} = M_{11} v_1' + M_{12} v_2 + M_{13} v_4' + M_{14} v_5. \tag{4.15}$$

Again, we use a concept from the pFFT method: interpolation. We assume that the potential at a test point on S_{3D} can be interpolated from the potentials at nearby points in S_{top} and S_{mid}. We have these potentials already: w_1^{planar} and w_4^{planar}.

As shown in reference [2], we can write this interpolation as follows, using four interpolation matrices to indicate the process of interpolation. Precorrection terms are also included to correct for any inaccuracies in the near-field interactions, and are indicated with a C.

$$w_7^{planar} = I_{top} w_3^{planar} + I_{mid} w_4^{planar} + C_{interp}^{top} v_{1'} + C_{interp}^{mid} v_2 \tag{4.16}$$

Three-dimensional panel to three-dimensional panel scattering

Finally, we have to compute the components of the matrix-vector product that correspond to three-dimensional sources in v_8 and v_7 scattering to test points on S_{3D}. We will denote this part of the matrix-vector product as:

$$w_7^{3D} = M_{15}v_7 + M_{16}v_8. \tag{4.17}$$

We now combine both the concepts used in the previous section. We will use both projection and interpolation to approximate the far-field interactions for this type of scattering. We will interpolate the results found using projected sources.

$$w_7^{3D} = I_{top}(M_1 v_4^{proj} + M_3 v_5^{proj}) + I_{mid}(M_3 v_4^{proj} + M_1 v_5^{proj}) + C_{both,d}v_7 + C_{both,m}v_8 \tag{4.18}$$

Notice that we already have computed all of the variables in this equation, with the exception of the precorrection matrices.

4.2.4 Accelerated matrix-vector product

We show here the overall computations needed to implement an accelerated matrix-vector black-box for the three-dimensional problem.

In addition to the "kernel matrices" required for the planar problem, we must precompute some values before calling the black-box matrix-vector product for the three-dimensional problem. The cost associated with this setup is $O(N_{3D})$.

Setup computations

1. Projection matrices: $P_{mid}^{proj,d}$, $P_{mid}^{proj,m}$, $P_{top}^{proj,d}$, $P_{top}^{proj,m}$

2. Interpolation matrices: I_{top}, I_{mid}

3. Precorrection matrices: $C^{top}_{proj,m}$, $C^{top}_{proj,d}$, $C^{mid}_{proj,m}$, $C^{mid}_{proj,d}$, C^{top}_{interp}, C^{mid}_{interp}, $C_{both,d}$, $C_{both,m}$

Now we give an overview of the matrix-vector black box algorithm, which takes v' and produces $w' = \tilde{A}v'$.

Matrix-vector black box

1. Compute planar to planar scattering: Find v via zero-padding, and $w^{planar} = Av$ using planar black-box matrix-vector product.

2. Compute three-dimensional to planar scattering

$$v_4^{proj} = P^{top}_{proj,d}v_7 + P^{top}_{proj,m}v_8 \tag{4.19}$$

$$v_5^{proj} = P^{mid}_{proj,d}v_7 + P^{mid}_{proj,m}v_8 \tag{4.20}$$

$$w_3^{3D} = M_1v_4^{proj} + M_3v_5^{proj} + C^{top}_{proj,d}v_7 + C^{top}_{proj,m}v_8 \tag{4.21}$$

$$w_4^{3D} = M_3v_4^{proj} + M_1v_5^{proj} + C^{mid}_{proj,d}v_7 + C^{mid}_{proj,m}v_8. \tag{4.22}$$

3. Compute planar to three-dimensional scattering via interpolation

$$w_7^{planar} = I_{top}w_3^{planar} + I_{mid}w_4^{planar} + C^{top}_{interp}v_{1'} + +C^{mid}_{interp}v_2 \tag{4.23}$$

4. Compute three-dimensional to three-dimensional scattering

$$w_7^{3D} = I_{top}(M_1v_4^{proj} + M_3v_5^{proj}) + I_{mid}(M_3v_4^{proj} + M_1v_5^{proj}) + C_{both,d}v_7 + C_{both,m}v_8 \tag{4.24}$$

5. Combine results, and apply diagonal matrices, to yield subvectors w

$$w_1 = D_2v_1 + D_3v_4 \tag{4.25}$$

$$w_2 = D_4v_3 + D_5v_6 \tag{4.26}$$

$$w_3 = w_3^{planar} + w_3^{3D} \tag{4.27}$$

$$w_4 = w_4^{planar} + w_4^{3D} \tag{4.28}$$

$$w_5 = w_5^{planar} \tag{4.29}$$

$$w_6 = w_6^{planar} \tag{4.30}$$

$$w_7 = w_7^{planar} + w_7^{3D} \tag{4.31}$$

$$w_8 = D_6 v_8 \tag{4.32}$$

$$\tag{4.33}$$

6. Discard elements of w_1 and w_4 that do not correspond to panels on $S_{top'}$ to yield $w_1^{'}$ and $w_4^{'}$ respectively. Concatenate $w_1^{'}$, w_2, w_3, $w_4^{'}$,w_5,w_6,w_7 and w_8 to yield w'.

Chapter 5

Computational results

In this chapter we present computational results from a C++ implementation of our algorithm for the planar problem. All results were run on a 800 MHz Itanium processor with 32GB of RAM.

5.1 Code summary

The solution of the resulting currents due to applied voltages (that is one right-hand side) involves the following steps. In the case of a resistance extraction, where a solution is found for multiple right-hand sides, the first three steps are done only once, while the GCR iterative solve is done once for each right-hand side. Later, we will compare the relative costs of these steps.

Solve for currents

```
Mesh Problem
Precompute convolution FFTs
Precompute preconditioner and its LU factorization
Solve system using GCR
```

The iterative solve involves the following algorithm. Later we compare the relative CPU costs of these steps.

GCR algorithm

```
Compute initial residual, r
At each iteration k,
   Backsolve with LU factorization of preconditioner to find z
   Compute matrix-vector product, Ar
   Update new search direction, by orthogonalizing Ar
        wrt to all k-1 previous search directions
   Check new residual to see if converged
Backsolve solution with LU factorization to find true solution
Return result
```

Finally, the matrix-vector-product black box used at each iteration of GCR has a number of steps associated with it. We will later show the relative costs of these subtasks:

Matrix-vector-product black box

```
Forward transform input vector, via FFTs
Convolution of input vector with Green's Function
     done via frequency domain calcuations
Inverse transformation of results
Applying boundary conditions
```

5.2 Test cases

For testing, we choose a standard geometry of a 0.1 cm x 0.1 cm square substrate with a height of 0.01cm. We choose a top conductivity of 0.01, and bottom conductivity

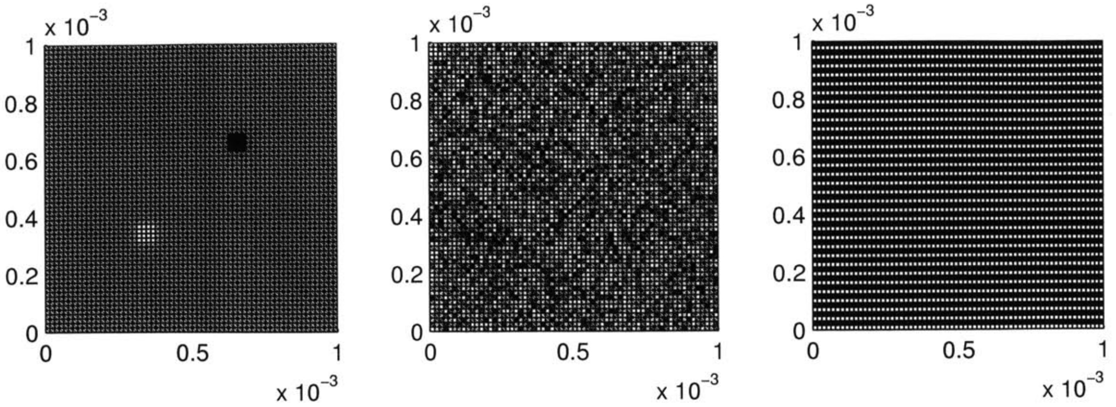

Figure 5-1: Three different top boundary conditions for problem: two-contact (left), random (center), striped (right)

of 0.05, with the interface of these layers at 0.008cm.

To show robustness under different boundary conditions, we show convergence under three types of boundary conditions, illustrated in Figure 5.2. These boundary conditions correspond to different applied voltages on the top surface, either two contacts, randomly selected voltages between 0V and 1V, and alternating stripes of -1V and 1V. Also, the backplane is either given as a ground plane or floating back plane.

5.3 Results

5.3.1 Convergence: iteration counts

First we demonstrate robustness of the solver for different top and bottom boundary conditions. Table 5.1 shows that for all of our boundary condtion types, the iteration counts grow slowly with problem size, indicating that our precondtioner choice is effective.

Table 5.1: Iteration counts for solution of nominal problem under different boundary conditions and discretizations.

Problem Size ($N_x \times N_y$)/ Boundary Conditions	(64x64)	(128x128)	(256x256)	(512x512)	(1024x1024)
Contacts/GND	21	27	36	48	63
Random/GND	31	41	54	71	93
Striped/GND	25	32	43	58	74
Contacts/Float	22	31	42	57	63
Random/Float	28	37	50	66	94
Striped/Float	23	31	42	55	74

5.3.2 Computational complexity

We next demonstrate that our total computational complexity grows slowly with problem size. This result is shown in Table 5.2.

5.3.3 Computational complexity of program subtasks

We demonstrate the relative costs of various aspects of our algorithm. As is readily apparent in Table 5.3, the iterative matrix solver is the costliest task, with all others being neglible for most practical problem sizes.

5.3.4 Computational complexity of iterative method

We next illustrate the relative costs of various aspects of our iterative method. As shown in Table 5.4, the cost of the iterative matrix solver is dominated by the matrix-vector product, although the back orthogonalization becomes more and more expensive with larger problems and their resulting slower convergence. For that reason, it might be better to switch to a different iterative method that does not require such orthogonalization, such as QMR.

Table 5.2: Total CPU times (ms) for solution of nominal problem under different boundary conditions and discretizations.

Problem Size($N_x \times N_y$)/ Boundary Conditions	(64x64)	(128x128)	(256x256)	(512x512)	(1024x1024)
Contacts/GND	1.5e4	8.5e5	5.3e5	3.3e6	2.0e7
Random/GND	2.4e4	1.5e5	9.7e5	6.1e6	3.7e7
Striped/GND	1.8e4	1.1e5	7.0e5	4.5e6	2.6e7
Contacts/Float	1.6e4	1.0e5	6.7e5	4.4e6	2.0e7
Random/Float	2.1e4	1.3e5	8.7e5	2.0e6	3.8e7
Striped/Float	1.7e4	1.0e5	6.8e5	2.1e6	2.6e7

Table 5.3: Total time per task for setup and solution of nominal problem, with Contacts/GND boundary conditions.

Problem Size	#UKs	Meshing Time	FFT Setup	Precond Setup	GCR Solve
(64x64)	2.5e3	1.9e2	4.1e2	1.5e	1.4e4
(128x128)	9.8e4	7.5e2	1.5e3	7.1e2	8.2e4
(256x256)	3.9e5	3.0e3	6.8e3	3.0e3	5.3e5
(512x512)	1.6e6	1.2e4	2.7e4	1.5e4	3.3e6
(1024x1024)	6.3e6	4.8e4	8.9e4	6.1e4	2.0e7

Table 5.4: Total CPU times spent on substasks of iterative solver, GCR, for nominal problem, with Contacts/GND boundary conditions.

Problem Size ($N_x \times N_y$)	#UKs / #UKs	Matrix-vector product	Back orthogonalization	Preconditioner backsolve
(64x64)	2.5e3	8.5e3	4.3e3	6.4e2
(128x128)	9.8e4	4.5e4	3.0e4	3.7e3
(256x256)	3.9e5	2.6e5	2.2e5	2.3e4
(512x512)	1.6e6	1.5e6	1.6e6	1.4e5
(1024x1024)	6.3e6	8.0e6	1.1e7	7.1e5

Table 5.5: Total CPU times spent on subtasks of matrix-vector product function of nominal problem, with Contacts/GND boundary conditions.

Problem Size($N_x \times N_y$)/	#UKs	Forward FFTs	Freq. Domain Calcs.	Inverse FFTs	BCs
(64x64)	2.5e3	2.0e3	4.6e3	1.3e3	2.0e2
(128x128)	9.8e4	1.1e4	2.4e4	7.2e3	1.0e3
(256x256)	3.9e5	7.1e4	1.3e5	4.6e4	5.6e3
(512x512)	1.6e6	4.4e5	6.6e5	2.9e5	3.1e4
(1024x1024)	6.3e6e6	2.4e6	3.5e6	1.6e6	1.6e5

5.3.5 Computational complexity of matrix-vector product

Finally, we demonstrate that the main cost of the matrix vector product lies in the transformations , which have a cost that grows as $O(N_xN_ylog(N_xN_y))$, and the and frequency domain calcuations, which grow as $O(N_xN_y)$. This result is given in Table 5.5.

Bibliography

[1] A. Vithayathil, X. Hu and J.K. White. "Substrate resistance extraction using a multi-domain surface integral formulation." *IEEE SISPAD*, pp.323 - 326, September 2003.

[2] J. Phillips. *"Rapid Solution of Potential Integral Equations in Complicated 3-dimesional Geometries."* Ph.D. thesis, MIT EECS Department, June 1997.

[3] P.K. Kythe. An Introduction to Boundary Element Methods, CRC Press, Boca Raton, FL, 1995.

[4] S. Kuo, M. Altman, J. Bardhan, B. Tidor and J. White. "Fast Methods for Simulation of Biomolecule Electrostatics." *IEEE/ACM Conference on Computer-Aided Design,* June 2002.

[5] E. Charbon, R. Gharpurey, P. Miliozzi, R.G. Meyer and A. Sangiovanni-Vincentelii. Substrate Noise Analysis and Optimization for IC Design, Kluwer Academic Publishers, 2001.

[6] R. F. Harrington. Field Computation by Moment Methods, MacMillan, New York, 1968.

[7] C. Van Loan. Compuational Frameworks for the Fast Fourier Transform, SIAM, New York, 1992.

[8] M. Niknejad, R. Gharpurey, and R. Meyer. "Numerically Stable Green Function for Modeling and Analysis of Substrate Coupling in Integrated Circuits." *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems,* vol. 17, no.4, pp. 305-315, April 1998.

[9] J.P. Costa, M. Chou, L.M. Silveira "Efficient Techniques for Accurate Modeling and Simulation of Substrate Coupling in Mixed-Signal IC's" *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems*, vol. 18 no.5, pp. 597-607, May 1999

[10] T.A. Johnson, R.W. Knepper, V. Marcello, W. Wang "Chip Substrate Resistance Modeling Technique for Integrated Circuit Design" *IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems,* vol. CAD-3, no.2, pp. 126-134, April 1984.

[11] B.R. Stanisic, N.K. Verghese, R.A. Rutenbar, L.R. Carley, D.J. Allstot. *IEEE Journal of Solid-State Circuits*, Vol. 29, No. 3

[12] N. Verghese and D. Allstot. "Rapid Simulation of Substrate Coupling Effects in Mixed-Mode ICs." *Proc. IEEE Custom Integrated Circuits Conference*, pp. 18.3.1-18.3.4, May 1993.

[13] E. Schrik, N.P. van der Meijs "Combined BEM/FEM Substrate Resistance Modeling." *Proc. of the Design Automation Conference*, pp.771-776, June 2002

[14] J. Kanapka, J. White. "Highly Accurate Fast Methods for Extraction and Sparsification of Substrate Coupling Based on Low-Rank Approximation." *IEEE/ACM ICCAD* pp. 417-423, 2001

[15] T. Sakar, E. Arvad and S. Rao. "Application of FFT and the Conjugate Gradient Method for the Solution of Electromagnetic Radiation from Electrically Large and Small Conducting Bodies" *IEEE Trans. Antennas and Propagation,* vol. AP-34, pp. 635-640, 1986.

6318 67