Fast Algorithms for Ill-Conditioned Dense-Matrix Problems in VLSI Interconnect and Substrate Modeling

by

Mike Chuan Chou

B.S. Physics, California Institute of Technology (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

June 1998

Signature of Author ....
Department of Electrical Engineering and Computer Science
April 24, 1998

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

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

Fast Algorithms for Ill-Conditioned Dense-Matrix Problems in VLSI Interconnect and Substrate Modeling

by

Mike Chuan Chou

Submitted to the Department of Electrical Engineering and Computer Science
on April 24, 1998, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Integral equation methods are popular in the electrical simulation of three-dimensional structures since they require only surface meshing, and hence reduce dramatically the number of unknowns. However, they lead to dense matrices which are too expensive to store or factor directly. Iterative solutions based on approximate, matrix-free representations of the original linear system appear to be the only recourse. However, this alternative can also fail if the linear system is ill-conditioned, as is often the case. This thesis investigates the kinds of difficulties which arise and how they can be resolved, using two practical problems in the computer-aided design of VLSI systems.

The first part of this thesis deals with the modeling and simulation of three-dimensional integrated-circuit interconnect in the distributed RC, or electroquasistatic, regime. When a surface integral formulation is first combined with a multipole sparsification method, it is shown that small multipole approximation errors are magnified by the ill-conditioning resulting from the wide range of time constants in the dynamical system. In addition, this ill-conditioning also makes iterative solution impractical because of the large number of iterations required for convergence. A mixed surface-volume approach which effectively resolves both difficulties is proposed and successfully implemented. Results show that the cost of extracting a complete reduced-order model of the interconnect is only several times that of the basic capacitance extraction.

The second part of this thesis deals with the extraction of substrate coupling resistances, which can be formulated as a first-kind integral equation involving only the discretized, two-dimensional substrate contacts. Since first-kind integral equations lead to ill-conditioned linear systems, standard Krylov-subspace iterative solution algorithms are slow to converge. A fast-converging multigrid iterative method for first-kind integral equations is developed to overcome this difficulty. However, for the multigrid implementation to be efficient, the dense matrix representation at each level needs to be sparsified. A multilevel sparsification method based on moment-matching and eigendecomposition which handles edge effects more accurately than previously applied multipole expansion techniques is presented and incorporated into the multigrid solution algorithm. Results on realistic examples demonstrate that the combined approach is up to an order of magnitude faster than the sparsification plus a Krylov-subspace method, and orders of magnitude faster than not using sparsification at all.

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

Acknowledgments

First and foremost, I must thank my thesis advisor, Professor Jacob White. He taught me not only how to formulate new problems and ask the right questions, but more importantly, how to approach life with generosity and compassion. I will never forget the numerous instances in which he offered to help me cope with my personal and family difficulties. Without his constant and unconditional support, this thesis would not have been possible.

I would also like to thank my thesis committee members, Professors Alan Edelman and Steve Leeb, for reading my thesis and offering me their valuable suggestions.

Next, I would like to thank all members of the VLSI-CAD group on the eighth floor for providing me with the stimulating environment which transforms research from a potentially lonely experience to one that's fun and interactive. In particular, I'd like to thank Matt Kamon for being such a great friend and partner as we slogged through an incredibly dense book on integral equations together. Thanks to Joel Phillips for being a consistently insightful source of technical suggestions and criticism. Thanks to Miguel Silveira for being a source of wisdom, and for the numerous e-mail exchanges and several all-night discussions on the substrate problem. He has always been accomodating and supportive in any potential conflicts. I have also been lucky to have office-mates like Xuejun Cai, Jose Monteiro, Abe Elfadel, and Narayan Aluru. Each of them has been great fun, and has offered me generous support in the non-academic areas of my life. Thanks must also go to Frank Wang and Farzan Fallah for being great pals during my last year at MIT.

Life at MIT would have been unbearable without my best friend, Ralph Lin, who has been a source of support and inspiration in the best and worst of times. Many thanks to Sophia Chin for her unconditional love and support during the three years we were together. My sincere gratitude also goes to Phillip and Stephanie Nee, Hwa-Ping Chang, Cynthia Chuang, I-Bin Chen, Mike Kwan, Charles and Chi-Chi Hsu, Chunnan and Amanda Li, George and Susan Chou, Rebecca Xiong, Chris Lowy, and Brett Bochner. They have made MIT a truly unforgettable experience. Special thanks goes to Bei Wang for her friendship, and for helping me get through my last year at MIT.

Last but certainly not least, I'd like to thank Mom and Dad, who gave up a good life and came to the U.S. against all odds, so that my sister and I might have better opportunities to

succeed in life. I hope that this has all been worth it. It is time for them to retire and reclaim the good life which they have put off for fifteen years. It is to them that I dedicate this thesis.

To Mom and Dad, with love.

Contents

List of Figures

List of Tables

1

Introduction

Boundary value problems for Laplace's equation in three dimensions are usually solved via one of two general classes of numerical techniques. The first class requires the discretization of the entire region in which Laplace's equation holds, and includes the well-known finite-difference (FD) and finite-element(FEM) methods. This results in a large but *sparse* system of linear algebraic equations, which can be solved directly using sparse-matrix factorization, or iteratively using either conjugate-gradient style methods or multigrid methods. The most efficient among such techniques require an amount of storage and CPU time proportional to M, where M is the number of nodes, or unknowns, in the discretized region. However, since a huge number of unknowns will be generated, these methods become very expensive when applied to exterior bounary value problems, or to interior problems requiring very fine discretization.

For such problems, one usually resorts to integral equation methods, also called Green's function methods. These methods require only the discretization of relevant boundary features, *i.e.* those associated with a distribution of charges or dipoles. The term "boundary element method" (BEM) is used when the discretized surface is the boundary of an open set in $\Re^3$. Because only two-dimensional surfaces are discretized, integral equation methods lead to much smaller systems of linear algebraic equations than those produced by the FD or FEM methods. See Figures 1-1 and 1-2 for an exterior boundary value problem discretized with the FD or the BEM methods. However, since an integral equation couples every boundary unknown to every other boundary unknown, the resulting matrix is *dense*, and consequently too expensive to store or factor directly even for moderate N, where N is the size of the matrix, or equivalently the number of "boundary elements". Since the cost of direct solution via Gaussian elimination requires $O(N^3)$ operations and $O(N^2)$ storage, integral equation methods had not been considered suitable for "large" problems involving thousands or tens of thousands of unknowns.

The advent of *matrix-free, iterative* methods [1, 2, 3] during the past fifteen years has revived interest and activity in the application of integral equation methods for large problems. The

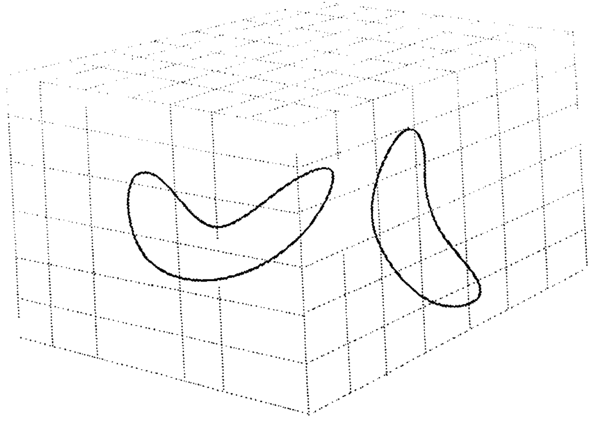


FIGURE 1-1: Finite-difference discretization.

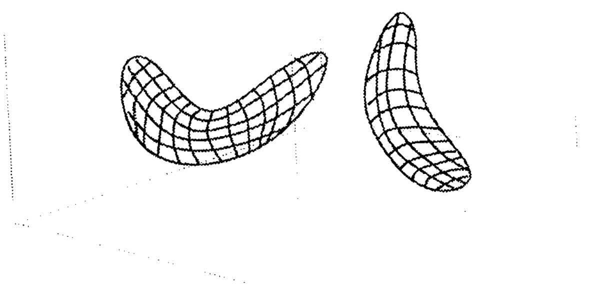


FIGURE 1-2: Boundary-element discretization.

central idea behind this is best illustrated with the solution of a linear system of equations

$$P \cdot q = v \tag{1.1}$$

where P is a large and *dense* matrix which is too cumbersome to store or factor directly. Consider solving (1.1) using an *iterative* algorithm. A generic iterative method is illustrated in Algorithm 1.

```
Algorithm 1 ( Generic Iterative Algorithm for Solving $P \cdot q = v$ ).

Set $k \equiv 1$, initial guess $q^{(1)}$ arbitrary.
Repeat {
    Compute matrix-vector product $v^{(k)} = P \cdot q^{(k)}$ .
    Determine residual $r^{(k)} = v^{(k)} - v$.
    Update current guess $q^{(k+1)} = q^{(k)} + \mathcal{F}\left(r^{(k)}\right)$.
    Set $k = k + 1$.
} Until residual norm $\|r^{(k)}\| < \epsilon$.
```

Common among all iterative methods is that the *residual* $r^{(k)}$ is needed to produce an incremental correction to the current guess through $\mathcal{F}\left(r^{(k)}\right)$, where $\mathcal{F}(\cdot)$ denotes some linear operation associated with a specific iterative method. The most important observation to be made about this approach is that if an efficient "black box" algorithm exists to compute the matrix-vector product $P \cdot q^{(k)}$ given arbitrary input $q^{(k)}$, then the residual $r^{(k)}$ is easily obtained, and (1.1) may be solved without explicit construction or storage of the matrix P. As a simple example, let the elements of P be defined by $P_{ij} = \delta_{ij} + u_i \cdot w_j$, where u, w are constant, length-N vectors, and δ is the Kroneker delta. Then P is a dense matrix given by

$$P = \begin{bmatrix} 1+u_1w_1 & u_1w_2 & \cdots & u_1w_N \\ u_2w_1 & 1+u_2w_2 & \cdots & u_2w_N \\ \vdots & \vdots & \ddots & \vdots \\ u_Nw_1 & u_Nw_2 & \cdots & 1+u_Nw_N \end{bmatrix}. \tag{1.2}$$

If the product $P \cdot q^{(k)}$ is computed by direct matrix-vector multiplication, the cost is N^2 operations. However, if we first compute the inner product $\alpha \equiv w^T q^{(k)}$, and then compute $P \cdot q^{(k)} = \alpha \cdot u + q^{(k)}$, the cost is now only $2N$ operations. This "algorithm" is motivated by the representation

$$P = I + u\ w^T, \tag{1.3}$$

where I is the identity. It is easy to see that (1.3) is equivalent to (1.2). We emphasize here that (1.2) is a *matrix* equation in which P is an array of numbers, with reference to a specific basis set, and that (1.3) is an *operator* equation in which P is defined as a sequence of linear operations without any regard to a basis. It is a minor abuse of notation that P is used to denote both a matrix and a linear operator. The field of numerical linear algebra has gradually moved away from the traditional matrix representation and toward the much more powerful operator point of view. We have just given here a simple example of the "matrix-free" approach. Alternatively, we can view (1.3) as a "sparsified" form of (1.2).

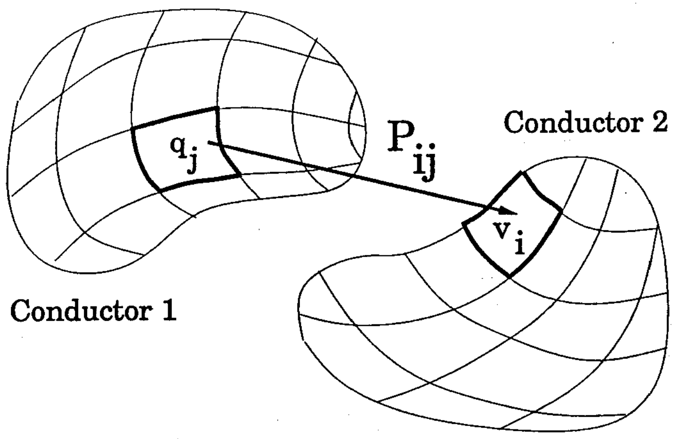


FIGURE 1-3: Every panel interacts with every other panel.

Now, consider solving a potential integral equation arising from the capacitance extraction problem [4]

$$\psi(x) = \int_S \sigma(x') \frac{1}{4\pi\epsilon \|x - x'\|} da', \quad x \in S, \tag{1.4}$$

where S is the collection of conductor surfaces, $\|\cdot\|$ is the Euclidean distance, da' is the differential conductor surface area, ϵ is the dielectric constant. This is an integral equation

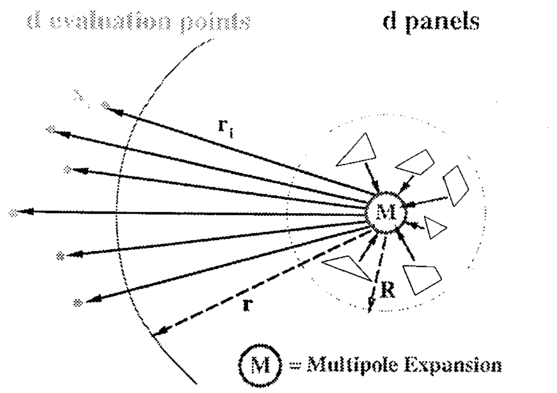


FIGURE 1-4: Multipole Expansions.

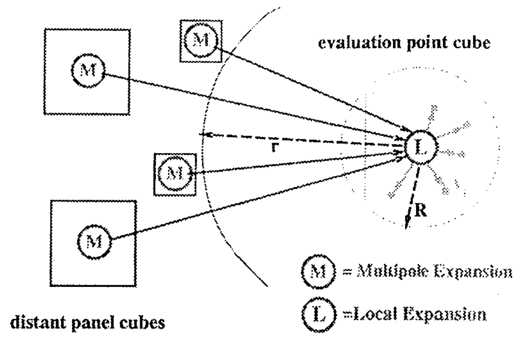


FIGURE 1-5: Local Expansions.

of the *first kind* [5] with a $(1/r)$ *kernel*, or Green's function. The problem is to solve for the unknown surface charge density $\sigma(x')$, given the prescribed conductor surface potentials $\psi(x)$. A common approach [6] is the piecewise constant collocation scheme, in which the conductor surfaces S are approximated by a set of N *panels* $\{p_i\}$. The charge q_i on each panel p_i is assumed to be uniformly distributed. If the potential $\psi(x)$ in (1.4) is evaluated at the *centroid* of each panel, the results is a linear system of the form (1.1) with matrix entries P_{ij} defined by

$$P_{ij} = \frac{1}{a_j} \int_{p_j} \frac{1}{4\pi\epsilon \|x_i - x'\|} da', \tag{1.5}$$

where x_i is the centroid of panel p_i and a_j is the area of panel p_j. See Figure 1-3 for an illustration of a two-conductor problem.

Since charge at any given panel will produce a non-zero potential at all panels, $P_{ij} \neq 0$ and the matrix P is *dense*. Hence we seek a fast, matrix-free algorithm to compute $P \cdot q$ given arbitrary input q. Since $v = Pq$ is simply the potential distribution resulting from the charge distribution q, this feat is accomplished if there is a fast way to compute potentials due to a collection of charges. The *fast-multipole* method (FMM) [7, 8], developed by Greengard and Rokhlin for the $(1/r)$ kernel, is an algorithm that computes *approximate* values $\tilde{v}$ for the N potentials in $O(N)$ operations and with $O(N)$ storage, given a fixed error bound $\|\tilde{v} - v\| < \epsilon$. The basic idea is to exploit the fact that $(1/r)$ is *smooth* away from the point $r = 0$. A collection of charges in a cluster of radius R may be represented by a single charge at the center, with strength equal to the sum of the charges, if the potential due to this cluster is to be evaluated at a distance r far from the cluster $r >> R$. This is a *monopole* expansion. Better accuracy is achieved with higher-order representations, called *multipole expansions*. This is illustrated in Figure 1-4. Similarly, a Taylor series, or *local expansion*, can be used to approximate the potentials at evaluation points inside a cluster of radius R, if all sources are at least a distance $r >> R$ away, as depicted in Figure 1-5. By keeping the ratio r/R at a fixed constant, the multipole and local expansions can be applied at various length scales in a hierarchical manner [8, 3]. The fast-multipole algorithm has been used successfully in combination with

Krylov-subspace iterative methods, such as GMRES [9], to solve potential integral equations [1, 3, 10, 11].

Part I of this thesis investigates integral formulations for a *dynamic* problem, namely, the simulation and macro-modeling of three-dimensional VLSI interconnect in the distributed RC, or electroquasistatic, regime. Consider the dynamical system

$$\begin{aligned} \frac{d}{dt}x(t) &= A \cdot x(t) + b\,u(t), \\ y(t) &= c^T \cdot x(t), \end{aligned} \tag{1.6}$$

where $x(t) \in \Re^N$ is the vector of state variables, $A \in \Re^{N \times N}$ is the system matrix, $b \in \Re^N$ and $c \in \Re^N$ are constant vectors, $u(t)$ is a scalar excitation, and $y(t)$ the scalar output. Examples of dynamical systems include the simple RC tree shown in Figure 1-6, and a three-dimensional interconnect shown in Figure 1-7. In each case, the system is driven by an input voltage source $u(t)$ and the output $y(t)$ is taken as voltage at a particular node.

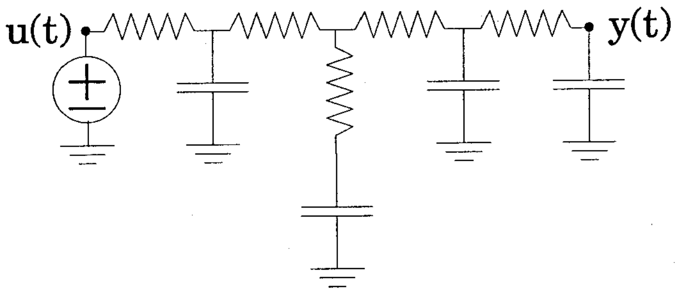


FIGURE 1-6: Simple dynamic system.

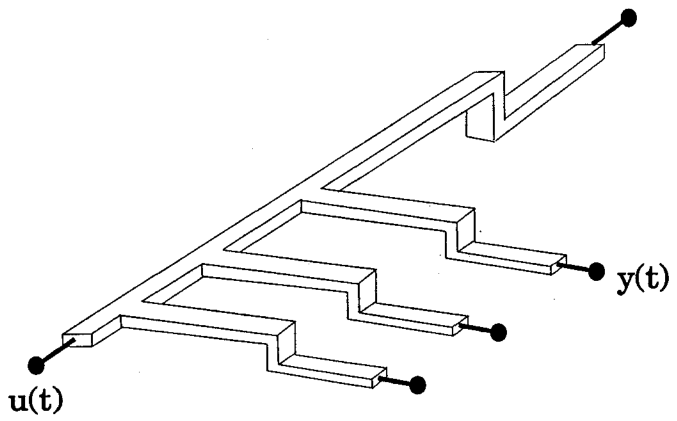


FIGURE 1-7: 3D interconnect.

Suppose that $u(t) = 1$, then the steady-state $\hat{x}$ is the solution of the linear system

$$A \cdot \hat{x} = -b. \tag{1.7}$$

Matrix equations similar to (1.7), but with different right-hand sides, also need to be solved for the problem of *model-order reduction*, the result of which is to produce a much smaller representation, or macromodel, of the originial linear circuit or interconnect. If the dynamical system (1.6) is derived from a circuit model in which each node has connections to only a few neighboring nodes, or if (1.6) results from a FD or FEM discretization of a three-dimensional region, then the matrix A is *sparse.* In such cases, it is feasible to solve (1.7) by LU factorization [12, 13]. However, if (1.6) results from an integral formulation, where only conductor surfaces are meshed, then A is *dense* and cannot be stored or factored directly. This is the case we are concerned with in this thesis. When a matrix-free, iterative method is employed to solve (1.7), two major difficulties are encountered, both resulting from the *ill-conditioning* in A. The first difficulty is that when a black-box algorithm such as the fast-multipole method is used to perform the matrix-vector multiplication, the multipole approximation error $\|\widetilde{v} - v\|$, usually negligible, becomes amplified by the *condition number* of A. The second difficulty is that Krylov-subspace based iterative algorithm (*e.g.* GMRES) converge slowly for ill-conditioned linear systems [14, 15], and that the number of iterations required grows with increasing condition number [14]. Iterative solution becomes very expensive if many iterations, or matrix-vector multiplies, are necessary.

The fundamental cause of the ill-conditioning is that the time constants of the dynamical system span a wide range, especially for long conductors. This is an essential feature in the "physics" of the problem, and is independent of the mathematical formulation used to solve it. However, we discovered that the ill-conditioning can be *isolated* with a mixed surface-volume formulation. The problem is decomposed into two parts: an exterior Laplace problem solved via the boundary-element method, and an interior Laplace problem solved via the finite-difference method. The ill-conditioning can be *isolated* by explicitly solving an *interior* problem with mixed boundary conditions. We show that for a small amount of additional work, the error magnification is eliminated. Also, the interior solution leads to a *preconditioner* which accelerates GMRES convergence by virtually removing the effect of time constants on matrix condition. This removal of ill-conditioning caused by time constants reduces the cost of solving a dynamic problem to one of solving a static problem. Realistic examples are given to demonstrate that constructing a full, time-dependent *reduced-order model* is only several times more costly than basic capacitance extraction. In addition, the multipole-accelerated code is used to compare the popular, one-dimensional diffusion equation against three-dimensional models for the case of long RC lines.

In Part II of this thesis, we turn to fundamental issues on the *convergence* of iterative algorithms for solving integral equations of the *first-kind,* which are of the form

$$\phi(x) = \int_S K(x;x')\sigma(x')\, da'. \tag{1.8}$$

In contrast, integral equations of the second kind [5] take the form

$$\phi(x) = \sigma(x) + \int_S K(x;x')\sigma(x')\, da'. \tag{1.9}$$

Standard results from functional analysis and the theory of Sobolev spaces [5] show that eigenvalues for first-kind integral equations with continuous or weakly singular kernels have an accumulation point at zero, whereas eigenvalues for second-kind integral equations with compact operators are bounded *away* from zero. The implication of this is that second-kind equations generate *well-conditioned* linear systems with bounded condition numbers, and that first-kind equations generate *ill-conditioned* linear systems, whose condition numbers continue to grow with mesh refinement. Since Krylov-subspace based algorithms converge rapidly for well-conditioned linear systems, GMRES has been the method of choice for the iterative solution of second-kind equations [1, 16]. When applied to solving first-kind equations, GMRES converges slowly, and *preconditioners* are necessary to reduce the number of iterations by reducing the effective matrix condition. However, the preconditioner derived in [3] for the problem of capacitance extraction, while effective, does not stop the number of iterations required from growing with increasingly fine discretizations.

To overcome this difficulty, we develop a *multigrid* iterative method for first-kind integral equations, and demonstrate that the convergence rate, and hence iteration count, is fixed, *i.e.* independent of discretization. Multigrid, or multilevel, methods operate by first decomposing the original problem into a set of sub-problems, each associated with a specific length scale, or *level*. Then, a relaxation, or smoothing, scheme is applied to each sub-problem to reduce error components at that length scale. The sub-problems "communicate" with one another via restriction and prolongation operators, collectively called intergrid transfer operators. Since the work associated with relaxation at each level decreases *geometrically* as the problem is *coarsened*, the total work required for going through each level once, or for one *multigrid sweep*, is bounded by a small multiple of the work at the finest level. Furthermore, since the relative error reduction resulting from a relaxation iteration at each level is uniform across all levels, the error reduction for a multigrid sweep is equal to the error reduction at a single level. Hence the multigrid convergence rate is independent of discretization. Although it is possible to formulate multigrid methods as multilevel *preconditioners* used to accelerate other iterative solvers such as GMRES, we shall not take such a view, since we will later demonstrate that GMRES with multigrid preconditioning converges only *slightly* faster than the stand-alone multigrid algorithm. This implies that the multigrid preconditioner turns the original matrix into something very "close" to the identity, in which case simple classical relaxation schemes work nearly as well as Krylov-subspace methods.

The vehicle for our multigrid development is the problem of substrate coupling resistance extraction for mixed-signal IC's. This is formulated as a first-kind integral equation involving only the two-dimensional substrate contacts. The two core components of a multigrid method are carefully developed. The first component, the smoothing operator, is cast as a *fixed-point iteration*, in which a sequence of *local* problems are solved to reduce the short-range, or high-frenquency, portion of the error. The second component, interpolation and prolongation, requires first that the original problem be *formulated* at different length scales. We accomplish this using a hierarchical set of *basis functions* in a Galerkin discretization. Since a coarse-level basis set forms a subspace of a fine-level basis set, interpolation and prolongation operators are simple to construct. In order to make the multigrid algorithm practical, it is necessary to *sparsify* the dense matrix-vector operations at each level in the discretized integral equation. Previous attempts on sparsification based on multipole approximations [17, 18] are inaccurate since they fail to model the edge effects of the substrate. We develop here a sparsification algorithm based on eigendecomposition, which accounts for the edge effects explicitly. At coarser levels, a moment-matching algorithm is developed to represent the problem on a correspondingly coarse and *regular* grid, on which eigendecomposition can be applied more cheaply. Numerical experiments are given to demonstrate that the resulting multigrid method achieves a constant convergence rate independent of discretization. For a realistic chip layout, the sparsified multigrid approach is up to an order of magnitude faster than a Krylov-subspace method plus sparsification, and orders of magnitude faster than not using sparsification at all.

Part I

The Transient Interconnect Problem

2

Overview of the Transient Interconnect Problem

When analyzing high-performance integrated circuit designs, it is well-known that the single lumped resistor-capacitor model of interconnect is insufficiently accurate. It has been shown [19] that reasonably accurate electro-quasistatic, or transient interconnect, simulations could be performed by computing the time evolution of the electric field both inside and outside the conductors via a finite-difference discretization of Laplace's equation. More recently, a boundary-element approach [20] based on Green's theorem was proposed, which performs the caculation using the same surface discretization used for ordinary capacitance extraction, thereby avoiding the large, exterior domain mesh and computation. However, the latter approach generates dense matrix problems, which require $O(N^3)$ operations to solve directly, and at least $O(N^2)$ to solve iteratively, where N is the number of surface unknowns. Therefore it is necessary to accelerate such methods when solving large problems. The direct application of the $O(N)$ fast-multipole algorithm on the boundary-element formulation produces unacceptable results because the multipole errors are magnified by the ill-conditioning in the linear system, which results from the wide range of time constants in the dynamics. To overcome this difficulty, we derive a mixed surface-volume formulation, and show how it prevents the magnification of the multipole error. In this formulation, the interior finite-difference method is used to solve Laplace's equation *inside* the conductors, and the boundary-element method is used to solve the *exterior* Laplace problem.

For three-dimensional interconnect structures to be included along with the actual transistors in a coupled, SPICE-level circuit simulation, it is necessary to construct low-order macromodels whose terminal behaviors essentially capture the complicated 3-D field interactions among the interconnect. Most model order reduction techniques, such as Asymptotic Waveform Evaluation [21] and the more recent Pade-via-Lanczos [22] and Arnoldi [23] algorithms, have been successful because it is feasible to carry out an LU decomposition of the associated *sparse* system matrix, after which each solve can be performed cheaply. For problems involving large, *dense* matrices, direct factorization is computationally intractable. Iterative methods can also be expensive if many solution iterations, or matrix-vector product computations, are required for convergence, as is the case for ill-conditioned linear systems. We show how the surface-volume formulation can be modified slightly to allow effective *preconditioning*, which produces rapid convergence in the iterative solution.

The outline of our exposition is as follows. Chapter 3 deals with the integral formulation of the transient-interconnect problem. The surface-integral formulation is briefly outlined in Section 3.1. The phenomenon of ill-conditioning is described in Section 3.2. The surface-volume formulation is then derived in Section 3.3, and the resulting error control demonstrated in Section 3.4. Chapter 4 deals with the problem of efficient model-order reduction, or macro-modeling. The guaranteed stable Arnoldi algorithm for model-order reduction is reviewed in Section 4.1. The modified surface-volume formulation and preconditioning techniques are presented in Section 4.2. Section 4.3 describes the method-of-images for including ground-planes. Examples of model-order reduction are presented in Section 4.4, where we show that the cost associated with generating a q-th order model is order N, and is less than that of peforming q capacitance extractions. In Chapter 5, we compare the popular diffusion model of the distributed RC line to three-dimensional calculations.

3

Problem Formulation and Error Control

3.1 The Surface Integral Formulation

For the transient interconnect problem, the system is assumed to be in the electro-quasistatic (EQS) regime. The scalar potential $\psi(x,t)$ satisfies Laplace's equation in all of space except on conductor surfaces, where charge can accumulate [20]

$$\nabla^2\psi(x,t) = 0, \qquad x \notin S, \tag{3.1}$$

where S is the union of all conductor surfaces. Since Laplace's equation (3.1) holds both inside and outside of the conductors, all charges in the system reside on the conductor surfaces S. Therefore, the potential ψ is related to the conductor surface charge density, ρ_s, through the superposition integral,

$$\frac{\partial\psi(x,t)}{\partial t} = \int_S \frac{1}{4\pi\epsilon\|x-x'\|}\frac{\partial\rho_s(x',t)}{\partial t}da', \tag{3.2}$$

where the regions inside and outside the conductors are assumed to have uniform permittivity ϵ. Charge conservation [24] at the surface yields the continuity condition

$$\frac{\partial\rho_s(x)}{\partial t} = J^{\text{ internal}}(x) - J^{\text{ external}}(x), \tag{3.3}$$

where $J^{\text{ internal}}$ and $J^{\text{ external}}$ are the normal current densities taken just inside and just outside the conductor surface. Inside a conductor, the current obeys the constitutive relation

$$J^{\text{ internal}}(x) = -\sigma\frac{\partial\psi}{\partial n}(x), \tag{3.4}$$

where σ is the conductivity and n is the outward normal to the surface S.

Combining (3.2), (3.3), and (3.4) results in an integral formulation

$$-4\pi\tau\frac{\partial\psi(x,t)}{\partial t} = \int_S \frac{1}{\|x-x'\|}\frac{\partial\psi}{\partial n'}(x',t)da' + \frac{1}{\sigma}\int_S \frac{1}{\|x-x'\|}J^{\text{ external}}(x',t)da', \tag{3.5}$$

where $\tau = \epsilon/\sigma$ is the dielectric relaxation time of the conductors, x is a point on a conductor surface, n' is the outward normal to the conductor surface, and $\|x - x'\|$ is the Euclidean distance betweeen x and x'. Careful application of Green's theorem [25] [20] to the first integral on the right-hand side of (3.5) yields

$$-4\pi\tau\frac{\partial\psi(x,t)}{\partial t} = 2\pi\psi(x,t) + \int_S \psi(x',t)\frac{\partial}{\partial n'}\frac{1}{\|x-x'\|}da' + \frac{1}{\sigma}\int_S \frac{J^{\text{ external}}(x',t)}{\|x-x'\|}da'. \tag{3.6}$$

Let $S_{\text{ contact}}$ be the subset of S which is in contact with external voltage sources, and let $S_{\text{ free}} = S\backslash S_{\text{ contact}}$ be the non-contact, or free, surfaces. Then $\psi(x,t)$ for $x \in S_{\text{ contact}}$ is known *a priori.* Since there is no external current flow at non-contact surfaces, we also have *a priori* that $J^{\text{ external}}(x,t) = 0$ for $x \in S_{\text{ free}}$.

To numerically solve (3.6) for ψ at non-contact surfaces and for $J^{\text{ external}}$ at contact surfaces, the conductor surfaces are broken into N small tiles, or panels. It is then assumed that on each panel l, there is a constant potential ψ_l and a constant external supply current density J_l. A collocation scheme [6], in which (3.6) is enforced at the centroid of each panel, is used to generate a system of N equations. The result is a $N \times N$ dense linear system

$$-4\pi\tau\frac{d}{dt}\Psi(t) = (2\pi\mathbf{I} + \mathbf{D})\Psi(t) + \mathbf{P}\frac{1}{\sigma}\mathbf{J}^{\text{ ext}}(t), \tag{3.7}$$

where $\Psi \in \Re^N$, $\mathbf{J}^{\text{ ext}} \in \Re^N$ represent the discretized panel potentials and external supply current densities. The elements of the dense matrices $\mathbf{D} \in \Re^{N\times N}$ and $\mathbf{P} \in \Re^{N\times N}$ are

$$\mathbf{P}_{kl} = \frac{1}{a_l}\int_{\text{panel}_l}\frac{1}{\|x'-x_k\|}da', \tag{3.8}$$

$$\mathbf{D}_{kl} = \int_{\text{panel}_l}\frac{\partial}{\partial n'}\frac{1}{\|x'-x_k\|}da', \tag{3.9}$$

where x_k is the center of the k-th panel, and a_l is the area of the l-th panel. Mathematically, $\mathbf{P}_{kl}$ is the potential at x_k due to a unit *charge* distributed uniformly over panel l. Similarly, $\mathbf{D}_{kl}$ is the potential at x_k due to a unit *dipole* oriented along the normal to and distributed uniformly over panel l. The integrals in (3.9) and (3.9) are often referred to as *single-layer* and *double-layer* integrals [5], respectively.

Suppose N_c of the N surface panels are connected to voltage contacts whose potentials $\Psi_c \in \Re^{N_c}$ are known but whose supply currents $\mathbf{J}_c^{\text{ext}} \in \Re^{N_c}$ are unknown. It is then clear that (3.7) is an index-one differential-algebraic equation (DAE), solvable with backward-differencing formulas (BDF). In addition to the N_c elements of $\mathbf{J}_c^{\text{ext}}$, the unknowns also include the $N_f = (N - N_c)$ elements of $\Psi_f \in \Re^{N_f}$, which correspond to the non-contact panel potentials. Discretization of (3.7) in time with the backward-Euler method yields the linear system

$$\mathbf{H}\begin{pmatrix}\Psi_f \\ \mathbf{J}_c^{\text{ext}}\end{pmatrix}_{t=(m+1)h} = \frac{4\pi\tau}{h}\Psi_{t=mh} - \left(\frac{4\pi\tau}{h}\mathbf{I} + 2\pi\mathbf{I} + \mathbf{D}\right)\begin{pmatrix}0 \\ \Psi_c\end{pmatrix}_{t=(m+1)h}, \tag{3.10}$$

where h is the timestep. The matrix, or linear operator, $\mathbf{H} \in \Re^{N \times N}$, is defined by the transformation rule

$$\mathbf{H}\begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} = \left(\frac{4\pi\tau}{h}\mathbf{I} + 2\pi\mathbf{I} + \mathbf{D}\right)\begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix} + \mathbf{P}\begin{pmatrix} 0 \\ \mathbf{w} \end{pmatrix}, \tag{3.11}$$

where $\mathbf{v} \in \Re^{N_f}$ and $\mathbf{w} \in \Re^{N_c}$.

Since $\mathbf{H}$ is defined in terms of $\mathbf{P}$ and $\mathbf{D}$, the unknowns can be interpreted as a distribution of monopoles and dipoles, with the panels associated with the elements of $\mathbf{J}_c^{\text{ext}}$ acting as uniform monopoles (single layers), and the panels associated with Ψ_f acting as uniform dipoles (double layers).

3.2 Difficulties with Multipole Acceleration

Consider using a Krylov-subspace based iterative algorithm, such as GMRES [9], to solve (3.10) at each timestep. The k-th iteration of the GMRES algorithm requires computing the matrix-vector product $\mathbf{H}\mathbf{u}^k$, where $\mathbf{u}^k$ is the k-th GMRES search direction. Since $\mathbf{H}$ is dense, computing $\mathbf{H}\mathbf{u}^k$ directly requires N^2 operations. However, forming $\mathbf{H}\mathbf{u}^k$ is equivalent to computing potentials at N points due to a distribution of N monopoles and dipoles. *Fast-multipole* algorithms [26] [3] [27] can be used to compute approximate values of the N potentials in βN operations, where β is independent of N but dependent on the required accuracy.

If (3.10) is solved by using a fast-multipole algorithm to approximate $\mathbf{H}$ in (3.11), then

$$\tilde{\mathbf{H}}\begin{pmatrix} \widetilde{\Psi_f} \\ \widetilde{\mathbf{J}_c^{\text{ext}}} \end{pmatrix} = \mathbf{b}, \tag{3.12}$$

where $\tilde{\mathbf{H}}$ is the multipole approximation to $\mathbf{H}$, b is the right-hand-side of (3.10), and $\widetilde{\Psi_f}, \widetilde{\mathbf{J}_c^{\text{ext}}}$ are approximations to the true solution $\Psi_f, \mathbf{J}_c^{\text{ext}}$ in (3.10). The relative error in the computed potentials and currents is given by

$$\frac{\left\| \begin{pmatrix} \Psi_f \\ \mathbf{J}_c^{\text{ext}} \end{pmatrix} - \begin{pmatrix} \widetilde{\Psi_f} \\ \widetilde{\mathbf{J}_c^{\text{ext}}} \end{pmatrix} \right\|}{\left\| \begin{pmatrix} \widetilde{\Psi_f} \\ \widetilde{\mathbf{J}_c^{\text{ext}}} \end{pmatrix} \right\|} \leq \mathcal{K}(\mathbf{H}) \frac{\left\| \mathbf{H} - \tilde{\mathbf{H}} \right\|}{\left\| \mathbf{H} \right\|} \tag{3.13}$$

where $\left\| \mathbf{H} - \tilde{\mathbf{H}} \right\|$ is the multipole error and $\mathcal{K}(\mathbf{H})$ is the condition number [12] of $\mathbf{H}$. As clear from (3.13), the error from the multipole algorithm is magnified by the condition number of $\mathbf{H}$. To see the impact of even mild ill-conditioning in $\mathbf{H}$ on multipole algorithm errors, consider the first model problem, a rectangular wire with dimensions $L:1:1$, which is connected to a step voltage source at one end, shown in Figure 3-1. The steady-state voltage at any point on the conductor surface is 1 Volt. Figure 3-2 is a plot of the steady-state voltage at the opposite end of the wire (labeled v1) versus wire length, computed using a multipole-accelerated algorithm.

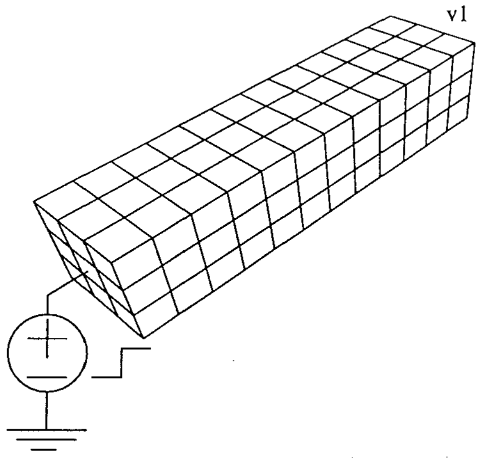


FIGURE 3-1: Single wire (L=4) connected to voltage source at one end.

For the algorithm used (second-order multipole expansions), the multipole approximation errors in the potential calculation is between 0.1-1%, but the steady-state error is much larger because of the magnification due to the ill-conditioning in **H**. As further evidence of this explanation, the condition number of **H** is plotted as a function of wire length in Figure 3-3.

For multi-conductor systems, the condition number of **H** grows as the spacing between conductors is reduced. Figure 3-4 shows a simple two-conductor problem. Each conductor has voltage boundary conditions at one end. Figure 3-5 shows that the condition number for the system increases as the spacing between the conductors is reduced. At very large separations, the two conductors are decoupled, and the condition number approaches that of the single-wire example in Figure 3-1.

We comment here that while higher-order multipole expansions can be used (at a much greater computational expense) to improve the accuracy, it only serves to *delay* the onset of error magnification, and since the condition number is observed to grow *quadratically* with the length of the conductors, we shall pursue other means of resolving this difficulty.

3.3 The Mixed Surface-Volume Formulation

We derive here a mixed surface-volume formulation which can be multipole-accelerated without loss in solution accuracy, although it does not change the condition of the system matrix. Consider the interior Dirichlet-to-Neumann operator $\mathcal{X}$, defined by the linear map between the surface potential ψ and its normal derivative $\frac{\partial}{\partial n}\psi$, where the limit for $\frac{\partial}{\partial n}\psi$ is approached from the interior of the conductor surfaces

$$\mathcal{X}\psi(x) \equiv \frac{\partial \psi}{\partial n}(x), \qquad x \in S. \tag{3.14}$$

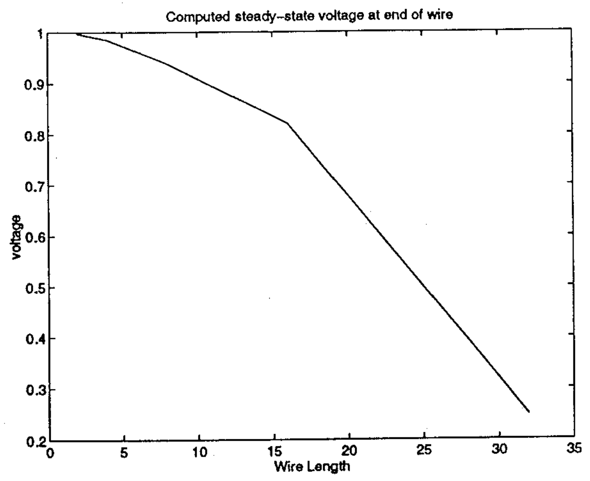


FIGURE 3-2: Steady-state voltage at v1 versus wire length.

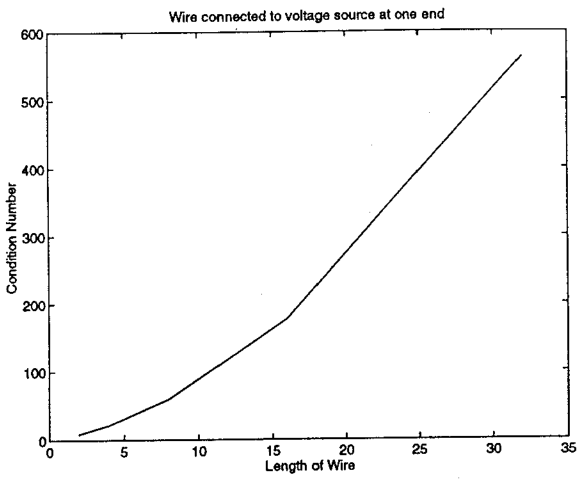


FIGURE 3-3: Condition number grows with wire length.

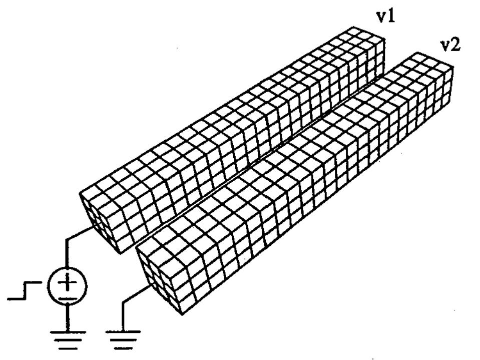


FIGURE 3-4: Parallel wires (L=8), each with voltage contacts at one end.

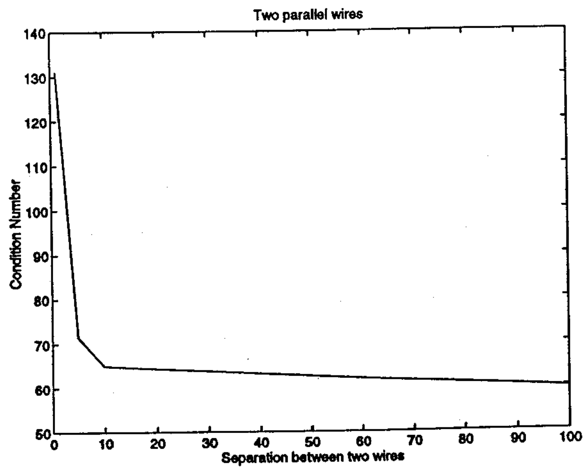


FIGURE 3-5: Condition number increases as wire separation decreases.

This relation allows the surface-integral formulation (3.5) to be written as

$$-4\pi\tau\frac{\partial\psi(x,t)}{\partial t} = \int_S \frac{1}{\|x-x'\|}\mathcal{X}\psi(x',t)da' + \frac{1}{\sigma}\int_S \frac{1}{\|x-x'\|}J^{\text{ external}}(x',t)da', \tag{3.15}$$

We now discretize the conductor surfaces into N panels and assume uniform potentials and currents on each panel as described in Section 3.1. The resulting matrix equation is

$$-4\pi\tau\frac{d}{dt}\Psi(t) = \mathbf{P}(\mathbf{X}\Psi(t) + \frac{1}{\sigma}\mathbf{J}^{\text{ ext}}(t)), \tag{3.16}$$

where $\mathbf{P}$ is as defined in (3.9). The matrix $\mathbf{X} \in \Re^{N\times N}$ approximates the continuous operator $\mathcal{X}$, and is defined by

$$\mathbf{X}\Psi \equiv \Psi_n, \tag{3.17}$$

where $\Psi_n \in \Re^N$ corresponds to $\frac{\partial}{\partial n}\psi$ at the N panels. Given Ψ at the surface nodes of a conductor, Laplace's equation can be solved in the *interior* domain with an interior finite-difference method to yield Ψ_n at each surface node. Hence, applying $\mathbf{X}$ implies solving the interior problems.

As before, a fixed-timestep, backward-Euler method is used to solve the DAE derived from (3.16). The resulting linear system is

$$\mathbf{A}\begin{pmatrix}\Psi_f \\ \mathbf{J}_c^{\text{ext}}\end{pmatrix}_{t=(m+1)h} = \frac{4\pi\tau}{h}\Psi_{t=mh} - \left(\frac{4\pi\tau}{h}\mathbf{I} + \mathbf{PX}\right)\begin{pmatrix}0 \\ \Psi_c\end{pmatrix}_{t=(m+1)h}. \tag{3.18}$$

The new operator $\mathbf{A}$ is defined by the transformation rule

$$\mathbf{A}\begin{pmatrix}\mathbf{v} \\ \mathbf{w}\end{pmatrix} = \frac{4\pi\tau}{h}\begin{pmatrix}\mathbf{v} \\ 0\end{pmatrix} + \mathbf{P}\left\{\mathbf{X}\begin{pmatrix}\mathbf{v} \\ 0\end{pmatrix} + \begin{pmatrix}0 \\ \mathbf{w}\end{pmatrix}\right\}. \tag{3.19}$$

The computation associated with applying $\mathbf{X}$ can be performed efficiently. Since the interior Laplace problem is solved *independently* for each conductor, the action of the $\mathbf{X}$ operator corresponds to solving a ***block-diagonal*** and ***sparse*** linear system. Thus the dominant cost of applying $\mathbf{A}$ in (3.19) comes from applying $\mathbf{P}$, which is a *dense* matrix operation since it couples every panel to all panels on all conductors. But as described in Section 3.2, the application of $\mathbf{P}$ to a vector can be multipole-accelerated. Therefore the combined surface-volume approach can be made very efficient.

The mixed surface-volume method provides an important guarantee on the solution accuracy. This is stated in the theorem below.

Theorem 3.1. If the steady-state solution of (3.15) is such that the surface potential on each conductor is a constant, and none of the conductors is floating, then the steady-state solution computed by the mixed surface-volume method is exact, regardless of multipole approximation error and discretization error.

Proof. Consider first the single conductor problem. From equation (3.16), the steady-state solution satisfies

$$\frac{d}{dt}\Psi = 0 = \mathbf{P}(\mathbf{X}\Psi + \frac{1}{\sigma}\mathbf{J}^{\text{ ext}}). \tag{3.20}$$

From the theory of fractional Sobolev spaces, it can be shown that the potential coefficient matrix $\mathbf{P}$ is non-singular given a sufficiently fine discretization [28]. It then follows that $\mathbf{X}\Psi + (1/\sigma)\mathbf{J}^{\text{ ext}} = 0$ in the steady-state. In the finite-difference implementation of $\mathbf{X}$, this is equivalent to a resistor network connected to external voltage sources [19]. Assuming that all voltage sources are at 1 Volt, the solution satisfies

$$\mathbf{X}\begin{pmatrix} \Psi_f \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{\sigma}\mathbf{J}_c^{\text{ ext}} \end{pmatrix} = 0. \tag{3.21}$$

In the equivalent resistor network picture, N_c of the surface nodes are connected to unit-voltage sources, while the remaining N_f surface nodes are left open-circuited. Network analysis immediately yields $\Psi_f = 1$ and $\mathbf{J}_c^{\text{ext}} = 0$, the exact steady-state solution. For many-conductor problems, the same result holds since each conductor is treated independently by the $\mathbf{X}$ operator.
□

Since (3.6) and (3.15) are both derived from (3.5), the Green's theorem based and the surface-volume based formulations are equivalent in their integral equation form. If we define the integral operators $\mathcal{PX}$ and $\mathcal{D}$ as

$$\mathcal{PX}\psi(x) \equiv \int_S \frac{1}{\|x-x'\|}\frac{\partial}{\partial n'}\psi(x')da' \tag{3.22}$$

$$\mathcal{D}\psi(x) \equiv \int_S \left(\frac{\partial}{\partial n'}\frac{1}{\|x-x'\|}\right)\psi(x')da' \tag{3.23}$$

then formally $\mathcal{PX} = (2\pi I + \mathcal{D})$ by Green's theorem [25], where I is the identity operator. Thus it follows that in the limit as the mesh becomes very fine (*i.e.* $N \to \infty$), the discretized versions of these operators approach each other, $\mathbf{PX} \approx (2\pi\mathbf{I} + \mathbf{D})$. Since $\psi(x) \equiv$ constant implies $\mathcal{X}\psi(x) = \frac{\partial}{\partial n}\psi(x) = 0$, both $\mathbf{PX}$ and $(2\pi\mathbf{I}+\mathbf{D})$ are singular matrices, with the vector $\{1, 1, ..., 1\}$ in the null space. The surface-volume formulation essentially factors the matrix $(2\pi\mathbf{I}+\mathbf{D})$ into the product of a singular $\mathbf{X}$ and a well-conditioned, non-singular $\mathbf{P}$. When the action of $\mathbf{P}$ is multipole-accelerated in the mixed formulation, errors are introduced only in the capacitance matrix of the surface panels, which does not alter the physical character of the system. This error appears only during the transient, and will be shown experimentally to be small and independent of condition number. This is expected since approximations are made only on $\mathbf{P}$, the well-conditioned part. The null space of $\mathbf{PX}$ is preserved. The same is not true for the Green's theorem based, pure boundary formulation, since multipole approximations are made on $\mathbf{D}$, which alters the null space of $(2\pi\mathbf{I}+\mathbf{D})$.

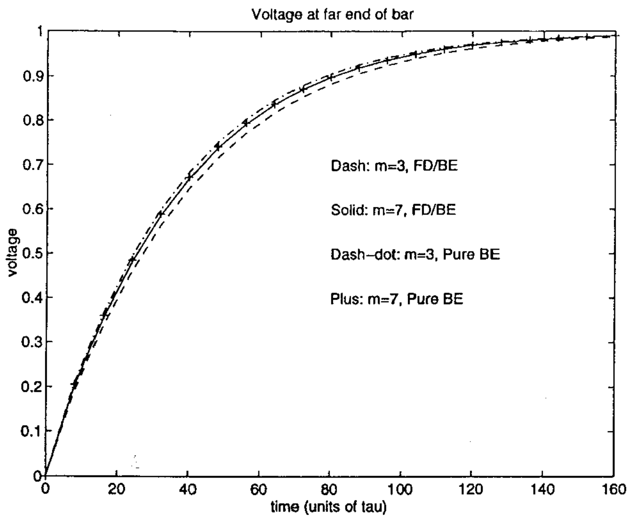


FIGURE 3-6: Without acceleration, both techniques produce correct results.

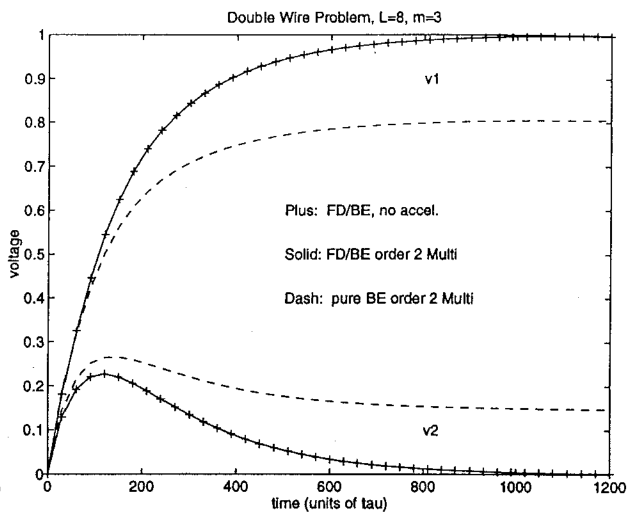


FIGURE 3-7: Multipole errors get magnified only in the pure BE method.

3.4 Computation Results

To show that both the pure boundary-element (BE) formulation and the mixed finite-difference/boundary-element (FD/BE) formulation produce similar results *without* multipole acceleration, we performed simulations on the single-wire conductor in Figure 3-1, using the dimensions L=4, W=1, H=1. Here we introduce a discretization parameter m, which represents the number of sections into which each unit-length is divided. For example, $m = 3$ in Figure 3-1. One end of the conductor is connected to a step voltage source, and the voltage waveform at the other end is shown in Figure 3-6. For the coarse mesh $m = 3$, both methods produce small discretization errors. For the fine mesh $m = 7$, the two methods converge to the same waveform. This confirms the validity of the new mixed formulation.

Multipole-acceleration is performed on both techniques for the double-wire example in Figure 3-4, with actual discretization (m=3) shown. At their near ends, one wire is connected to a step-voltage source, while the other is grounded. Simulated voltage waveforms at their far ends are shown in Figure 3-7. For the mixed finite-difference / boundary-element (FD/BE) formulation, the multipole-accelerated result produces the correct steady-state, and is practically indistinguishable from the non-accelerated, explicit calculations. The multipole-accelerated pure boundary-element (BE) technique is seen to produce obviously erroneous results, as reported in Section 3.2. Experimentally, for the mixed surface-volume formulation, we find that second-order multipole acceleration always produces results matching those of the explicit calculations, independent of the condition number.

A fairly complex three-dimensional interconnect example is presented here to demonstrate that the multipole-accelerated surface-volume method is necessary for large problems. The GMRES [9] iterative method without preconditioning is used to solve the linear systems (3.10) and (3.18). Polysilicon resistivity of $\rho = .02$ Ω·cm is assumed for all conductors, and oxide permitivity of $\epsilon_r = 3.2$ is assumed thoughout space. All computations are performed on a 266 MHz DEC AXP3000/900 workstation, with one gigabyte of physical memory.

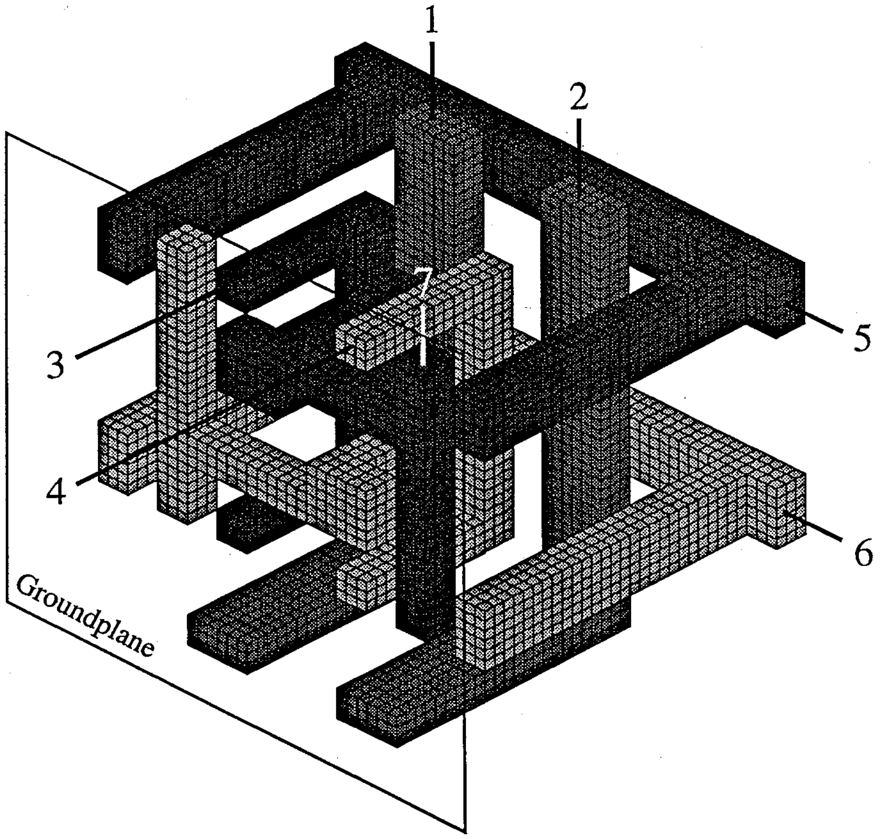


FIGURE 3-8: SRAM cell (m=3 mesh).

m	1	2	3	4
panels	986	3,944	8,874	15,776
FD time	0.3%	1.0%	2.5%	6.7%

Table 3-1: Problem size and FD time for SRAM.

Figure 3-8 displays a model of a six-conductor SRAM cell. The groundplane is shown but not used in this example. Each conductor is connected at a port, labeld 1 through 6. An additional port, labeled 7, is connected to conductor 3. Table 3-1 lists the number of surface unknowns for four successive refinements, with Figure 3-8 corresponding to ($m = 3$). The pair of L-shaped conductors (1 and 2) are the clock lines, while the pair of Π-shaped conductors (5 and 6) are the data lines. A third pair of intertwined, interior conductors (3 and 4) make interconnections between transistors in the cell. Assume that ports 3,4 are grounded, and that ports 5,6,7 are floating during a particular fetch cycle. We simulate the cross-talk noise induced on these lines by a unit-step voltage source on both ports 1 and 2. Such spurious signals must

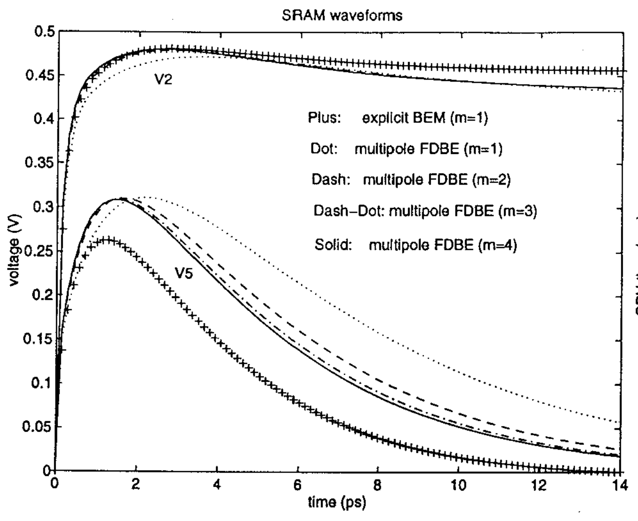


FIGURE 3-9: Waveforms computed using various mesh refinements.

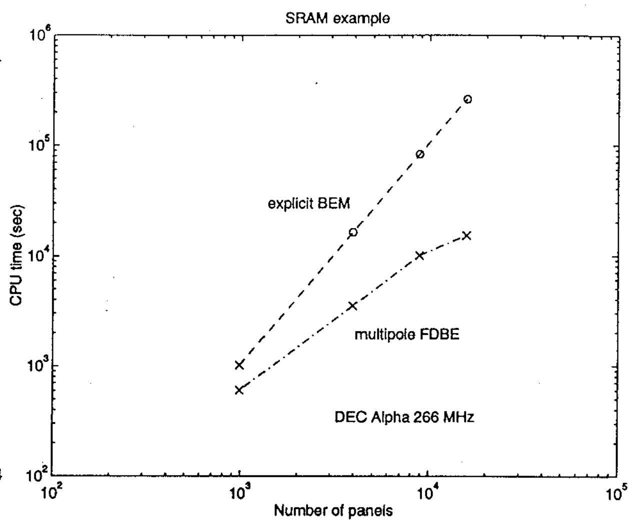


FIGURE 3-10: Comparing CPU times.

be minimized at the design phase to ensure error-free operation.

Results of the time-domain backward-Euler simulation are shown in Figure 3-9. The multipole-FDBE method is applied to four successively finer meshes ($m = 1, 2, 3, 4$), while the explicit-BE method, due to CPU time and memory limitations, is used only for the coarsest mesh ($m = 1$). From the voltage waveforms $v5$ and $v7$, it is seen that the ($m = 1$) mesh results in significant discretization error. The finer meshes generate large numbers of unknowns, and hence necessitates using the multipole-accelerated FDBE method. CPU times for the multipole-FDBE method are plotted in Figure 3-10, which can be seen to exhibit linear, or $O(N)$, growth. CPU times for the explicit-BEM approach are extrapolated from the ($m = 1$) mesh computation, and grows as $O(N^2)$ since it is a dense-matrix method. For the ($m = 4$) mesh, with 15,776 panels, the multipole-accelerated method is seventeen times faster than the dense-matrix approach.

We make the additional note here that the CPU time consumed by the interior finite-difference computation as a percentage of the total CPU time grows with increasing mesh refinement but remains small, as shown in Table 3-1. This confirms the earlier assertion that the cost of the boundary-element calculation is dominant.

4

Model-Order Reduction and Preconditioning

4.1 The Arnoldi Algorithm

In order to fully evaluate the effects of interconnect on overall circuit performance, it is necessary to perform a *coupled* circuit-interconnect simulation at the SPICE level. It is impractical to incorporate the large, dense matrices associated with the 3D interconnect directly into the circuit simulator. Instead, reduced-order models, which use small matrices to capture the current-voltage relations at the terminal ports of the interconnect, can be extracted from the full model and then used in the coupled simulation. Techniques such as Asymptotic Waveform Evaluation (AWE) [21] and the Pade-via-Lanczos (PVL) algorithm [22] have been used successfully for this purpose. In this section, we summarize previous work on the similar Arnoldi [29] algorithm, a numerically robust, orthogonal-projection based scheme which generates guaranteed stable reduced-order models [30].

Consider the single-input-single-output (SISO), linear, time-invariant system described by a system of first-order ordinary differential equations of the form

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t), \\ y(t) &= \mathbf{c}^T\mathbf{x}(t), \end{aligned} \tag{4.1}$$

where the N-vector $\mathbf{x}$ represents the circuit variables or the detailed internal voltages of the interconnect, and the $N \times N$ matrix $\mathbf{A}$ represents the detailed interactions among internal elements; $\mathbf{b} \in \Re^N$ is the excitation vector corresponding to the input terminal, and $\mathbf{c} \in \Re^N$ is the observation vector corresponding to the output terminal. The scalar quantities $u(t)$ and $y(t)$ are the input and output terminal-port variables, through which the linear system "interfaces" with external circuitry. The state-space representation of (4.1) is

$$s\mathbf{X}(s) = \mathbf{A}\mathbf{X}(s) + \mathbf{b}U(s),$$

$$Y(s) \;=\; \mathbf{c}^T\mathbf{X}(s), \tag{4.2}$$

where $\mathbf{X}, U$, and Y denote the Laplace transforms of $\mathbf{x}, u$, and y, respectively. The transfer function $F(s) \equiv Y(s)/U(s)$ can be written

$$F(s) = \mathbf{c}^T \cdot (\mathbf{I} - s\mathbf{A}^{-1}) \cdot \mathbf{p} = \Sigma_{k=1}^{N} \frac{\nu_k}{s - \lambda_k} \tag{4.3}$$

where $\mathbf{p} = -(\mathbf{A}^{-1}) \cdot \mathbf{b}$.

Since N can be of the order of tens of thousands, it is desirable to *reduce* the large and dense matrix $\mathbf{A}$ or $\mathbf{A}^{-1}$ in a manner that captures the low-frequency behaviour of the transfer function. This is done by matching Taylor series terms at $s = 0$. It has been shown in [29] that an Arnoldi-based orthogonalization process can be used to construct an orthonormal basis for the Krylov subspace

$$\mathcal{K}_q(\mathbf{A}^{-1}, \mathbf{p}) = span\{\mathbf{p}, \mathbf{A}^{-1}\mathbf{p}, \mathbf{A}^{-2}\mathbf{p}, \cdots, \mathbf{A}^{-(q-1)}\mathbf{p}\}. \tag{4.4}$$

After q steps, the Arnoldi algorithm returns a set of q orthonormal vectors, as the columns of the matrix $\mathbf{V}_q \in \Re^{N\times q}$, where N is the size of $\mathbf{A}$, and typically $q \ll N$. The reduced-order transfer function can then be constructed as

$$\begin{aligned} \tilde{F}(s) &= \tilde{\mathbf{c}}^T \cdot (\mathbf{I} - s\mathbf{H}_q)^{-1} \cdot \tilde{\mathbf{p}}, \\ \mathbf{H}_q &= \mathbf{V_q}^T(\mathbf{A}^{-1})\mathbf{V}_q, \\ \tilde{\mathbf{p}} &= \mathbf{V_q}^T\mathbf{p} = \|\mathbf{p}\|\mathbf{e}_1, \\ \tilde{\mathbf{c}}^T &= \mathbf{c}^T\mathbf{V}_q, \end{aligned} \tag{4.5}$$

where $\mathbf{H}_q$ is a $q \times q$ upper Hessenberg matrix. The transfer function $\tilde{F}(s)$ of the reduced q-th order system (4.5) has been shown in [29] to match $(q-2)$ derivatives, or moments, of the exact transfer function in (4.3) at $s = 0$, the low-frequency limit. The triplet $[\mathbf{H}_q, \tilde{\mathbf{p}}, \tilde{\mathbf{c}}]$ is said to be the *reduced-order model* of the triplet $[\mathbf{A}^{-1}, \mathbf{p}, \mathbf{c}]$.

It is possible to extend the present work to the multiple-input-multiple-output (MIMO) case using block algorithms similar to those described in [31] and [23].

4.2 Preconditioned Model-Order Reduction

Both the AWE and PVL algorithms have been successfully applied to reduce circuit networks for the lumped-element model of the interconnect since the associated large, sparse matrices can be factored to solve for the low-frequency moments of the transfer function [32]. The difficulty with applying the AWE, PVL, or Arnoldi algorithm to reduce three-dimensional interconnect models is that the associated large, dense matrices are too expensive to store and factor. Matrix-implicit iterative solution can also be expensive since many matrix-vector product computations are required for ill-conditioned problems. We recall here that the matrix ill-conditioning results from the wide range of time constants associated with typical interconnect and is thus independent of the problem formulation. In section 4.2.1, we show that straight-forward iterative solution converges slowly, and in section 4.2.2, we reformulate the mixed surface-volume approach slightly and derive an effective preconditioner, which allows for rapid convergence of the iterative solution. Section 4.3 describes how to include ideal ground-planes in the problem, and Section 4.4 presents the computational results.

4.2.1 Application of Arnoldi

To simplify notation in equation (3.16), section 3.3, let $\hat{\mathbf{P}} = \left(\frac{-1}{4\pi\epsilon}\right)\mathbf{P}$ and $\hat{\mathbf{D}} = \left(\frac{-1}{4\pi\tau}\right)\mathbf{PX}$, which results in

$$\frac{d}{dt}\Psi(t) = \hat{\mathbf{D}}\Psi(t) + \hat{\mathbf{P}}\mathbf{J}^{\text{ext}}(t). \tag{4.6}$$

Since $\hat{\mathbf{D}}$ is singular, the steady-state voltage $\Psi(t)$ is not uniquely determined by the external current $\mathbf{J}^{\text{ext}}(t)$. Thus we recast (4.6) as a differential-algebraic (DAE) system. This is done by using voltage sources instead of current sources, and then computing the resulting n-port frequency-dependent admittance matrix, which is then well-behaved near zero frequency. Recall that the N unknowns are the first N_f entries in the potential vector Ψ corresponding to the free potentials Ψ_f plus the N_c non-zero externally supplied currents $\mathbf{J}_c^{\text{ext}}$. The last N_c entries Ψ_c in Ψ are given *a priori* and correspond to external voltage sources. In frequency domain, the result is a system of equations

$$\begin{bmatrix} s\mathbf{I} - \hat{\mathbf{D}}_{ff} & -\hat{\mathbf{P}}_{fc} \\ -\hat{\mathbf{D}}_{cf} & -\hat{\mathbf{P}}_{cc} \end{bmatrix} \begin{bmatrix} \Psi_f(s) \\ \mathbf{J}_c^{\text{ext}}(s) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{D}}_{fc}\Psi_c(s) \\ (\hat{\mathbf{D}}_{cc} - s\mathbf{I})\Psi_c(s) \end{bmatrix} \tag{4.7}$$

where $\hat{\mathbf{D}}_{ff} \in \Re^{N_f \times N_f}, \hat{\mathbf{D}}_{fc} \in \Re^{N_f \times N_c}, \hat{\mathbf{D}}_{cf} \in \Re^{N_c \times N_f}, \hat{\mathbf{D}}_{cc} \in \Re^{N_c \times N_c}$ are partitions of the $\hat{\mathbf{D}}$ matrix

$$\hat{\mathbf{D}} = \begin{bmatrix} \hat{\mathbf{D}}_{ff} & \hat{\mathbf{D}}_{fc} \\ \hat{\mathbf{D}}_{cf} & \hat{\mathbf{D}}_{cc} \end{bmatrix}. \tag{4.8}$$

Similarly, $\hat{\mathbf{P}}_{ff} \in \Re^{N_f \times N_f}, \hat{\mathbf{P}}_{fc} \in \Re^{N_f \times N_c}, \hat{\mathbf{P}}_{cf} \in \Re^{N_c \times N_f}, \hat{\mathbf{P}}_{cc} \in \Re^{N_c \times N_c}$ are partitions of the $\hat{\mathbf{P}}$ matrix. The subscript f denotes the free-floating panels and the subscript c denotes panels in contact with voltage sources. Since the contacts are typically at the ends of long conductors, the number of contact panels is typically much smaller than that of floating panels. Therefore $\hat{\mathbf{P}}_{cc}$ is a small matrix and can be inexpensively inverted. Using $\hat{\mathbf{P}}_{cc}^{-1}$ allows (4.7) to be recast in the standard form for reduced-order modeling. Let $\mathbf{v} \in \Re^{N_c}, \mathbf{w} \in \Re^{N_c}$ be vectors of ones and zeros which selects the input voltage and output current panels, respectively. Then $\hat{\Psi}_c(s) = \mathbf{v}u(s)$ and $y(s) = \mathbf{w}^T \cdot \hat{\mathbf{J}}_c^{\text{ext}}(s)$, where $u(s)$ is the scalar voltage input and $y(s)$ is the scalar current output. After some amount of algebra, the admittance transfer function $g(s) = y(s)/u(s)$ is

$$g(s) = (k_0 + k_1 s) + \mathbf{c}^T \cdot (s\mathbf{I} - \mathbf{A})^{-1} \cdot \mathbf{b}, \tag{4.9}$$

where

$$\mathbf{A} = \hat{\mathbf{D}}_{ff} - \hat{\mathbf{P}}_{fc}\hat{\mathbf{P}}_{cc}^{-1}\hat{\mathbf{D}}_{cf}, \tag{4.10}$$

$\mathbf{b} = (\hat{\mathbf{D}}_{fc} - \hat{\mathbf{P}}_{fc}\hat{\mathbf{P}}_{cc}^{-1}\hat{\mathbf{D}}_{cc} + \mathbf{A}\hat{\mathbf{P}}_{fc}\hat{\mathbf{P}}_{cc}^{-1}) \cdot \mathbf{v}$, $\mathbf{c}^T = -\mathbf{w}^T \cdot \hat{\mathbf{P}}_{cc}^{-1}\hat{\mathbf{D}}_{cf}$,and k_0, k_1 are scalar constants.

Rewriting the second term in (4.9) as $f(s) = \mathbf{c}^T \cdot (\mathbf{I} - s\mathbf{A}^{-1}) \cdot \mathbf{p}$, with $\mathbf{p} = -(\mathbf{A}^{-1}) \cdot \mathbf{b}$, we apply the Arnoldi method to reduce the triplet $[\mathbf{A}^{-1}, \mathbf{b}, \mathbf{p}]$ by matching low-frequency moments. For each additional order in the model, a new vector in the Krylov subspace $\mathcal{K}_q(\mathbf{A}^{-1}, \mathbf{p})$ in (4.4) is generated by appling GMRES to perform an iterative solution of the system

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{RHS} \tag{4.11}$$

using only multipole-accelerated matrix-vector multiplies as described in Chapter 3.

As a numerical experiment, we directly apply the above Arnoldi algorithm to the simple interconnect in Figure 3-1, a single wire, with the supply voltage as input variable $u(s)$ and the supply current as the output vairable $y(s)$. The same calculation is performed for wires of varying lengths, keeping the other two dimensions fixed. Our numerical results, summarized in Table 4-1, show that the number of iterations, or matrix-vector product calculations, required for GMRES convergence in solving (4.11) grows quickly as the wire length, or aspect ratio, is increased. This is caused by the the system matrix $\mathbf{A}$ becoming more ill-conditioned as the range of time constants, or eigenvalues, grows with the wire length. It is well-known that the rate of convergence for Krylov-subspace style algorithms deteriorates with growing matrix condition number [14].

Wire Aspect Ratio	16	32	64	128
Mat-Vecs (Direct apply)	24	37	60	102
Mat-Vecs (Preconditioned)	4	5	5	6

Table 4-1: Matrix-vector multiplies required per order vs. length of wire

4.2.2 Preconditioned Formulation

We derive here a slightly modified version of equation (3.16), which can be easily preconditioned to accelerate convergence of the iterative method used to compute the Krylov subspace vectors. In this formulation, we assume that the panels in contact with voltage sources store no charge, or equivalently, that the contact capacitances have been removed. This model may also be supported based on physical arguments: terminal ports of interconnects are not exposed surfaced when the connections to transistors or other curcuitry have been made. The contact panels in practice exist *inside* conducting material, where Laplace's equation holds, and hence cannot store charge.

We start from the original integral formulation of (3.5). By writing the surface integrals over S as a direct sum of integrals over the contact and non-contact surfaces S_{contact} and S_{free}, (3.5) becomes

$$-4\pi\tau\frac{\partial\psi(x,t)}{\partial t} = \int_{S_{\text{free}}} \frac{1}{\|x-x'\|}\left[\frac{\partial\psi}{\partial n'}(x',t)+\frac{1}{\sigma}J^{\text{external}}(x',t)\right]da' + \int_{S_{\text{contact}}} \frac{1}{\|x-x'\|}\left[\frac{\partial\psi}{\partial n'}(x',t)+\frac{1}{\sigma}J^{\text{external}}(x',t)\right]da'. \tag{4.12}$$

The assumption that the charge density ρ_s is zero at contact surfaces S_{contact}, combined with the continuity condition (3.3) and the consitutive relation (3.4), implies

$$\frac{\partial\psi}{\partial n'}(x',t)+\frac{1}{\sigma}J^{\text{external}}(x',t)=0, \qquad x'\in S_{\text{contact}}. \tag{4.13}$$

Thus the second surface integral in (4.12)vanishes. In addition, since there are no external supply currents at non-contact surfaces, $J^{\text{external}}(x',t)=0\ \forall x'\in S_{\text{free}}$. The unknown potentials on S_{free} then satisfy

$$\begin{aligned}-4\pi\tau\frac{\partial\psi(x,t)}{\partial t} &= \int_{S_{\text{free}}}\frac{1}{\|x-x'\|}\frac{\partial\psi}{\partial n'}(x',t)da' \\ &= -\int_{S_{\text{free}}}\frac{1}{\|x-x'\|}\frac{1}{\sigma}J^{\text{internal}}(x',t)da', \qquad x\in S_{\text{free}}\end{aligned} \tag{4.14}$$

where (3.4) has been used in the second equality.

As before , we discretize S_f into N_f elements and S_c into N_c elements using the collocation scheme. The interior Dirichlet-to-Neumann operator defined in (3.17), Section 3.3 can be rewritten as

$$\mathbf{X}\begin{bmatrix}\mathbf{\Psi}_f\\ \mathbf{\Psi}_c\end{bmatrix} \equiv \begin{bmatrix}\mathbf{X}_{ff} & \mathbf{X}_{fc}\\ \mathbf{X}_{cf} & \mathbf{X}_{cc}\end{bmatrix}\begin{bmatrix}\mathbf{\Psi}_f\\ \mathbf{\Psi}_c\end{bmatrix} = -\frac{1}{\sigma}\begin{bmatrix}\mathbf{J}_f^{\text{int}}\\ \mathbf{J}_c^{\text{int}}\end{bmatrix}, \tag{4.15}$$

where $\mathbf{J}_f^{\text{int}}\in\Re^{N_f}$ and $\mathbf{J}_c^{\text{int}}\in\Re^{N_c}$ correspond to normal current densities *just inside* the N_f non-contact panels and the N_c contact panels, respectively. Discretization of (4.14) yields the $N_f\times N_f$ system

$$\frac{d}{dt}\mathbf{\Psi}_f(t) = -\hat{\mathbf{P}}_{ff}\mathbf{J}_f^{\text{int}}(t). \tag{4.16}$$

where $\hat{\mathbf{P}}_{ff}\in\Re^{N_f\times N_f}$ has been defined previously. Combining (4.15) and (4.16) and again letting $\mathbf{\Psi}_c(s)=\mathbf{v}u(s)$ and $y(s)=\mathbf{w}^T\cdot\mathbf{J}_c^{\text{int}}(s)$, we have the state-space form

$$s\mathbf{\Psi}_f(s) = \mathbf{A}\mathbf{\Psi}_f(s)+\mathbf{b}u(s), \tag{4.17}$$

$$y(s) = \mathbf{c}^T\cdot\mathbf{\Psi}_f(s)+d\cdot u(s), \tag{4.18}$$

where

$$\mathbf{A}=\sigma\hat{\mathbf{P}}_{ff}\mathbf{X}_{ff}\equiv\left(\frac{1}{4\pi\tau}\right)\mathbf{P}_{ff}\mathbf{X}_{ff} \tag{4.19}$$

$\mathbf{b}=(\frac{1}{4\pi\tau})\mathbf{P}_{ff}\mathbf{X}_{fc}\mathbf{v}$, $\mathbf{c}^T=\mathbf{w}^T\mathbf{X}_{cf}$, and $d=\mathbf{w}^T\mathbf{X}_{cc}\mathbf{v}$. The new expression for $\mathbf{A}$ in (4.19) is to be compared with that in (4.10). Notice that $\hat{\mathbf{D}}_{ff}=(\frac{1}{4\pi\tau}\mathbf{P}\mathbf{X})_{ff}\neq(\frac{1}{4\pi\tau})\mathbf{P}_{ff}\mathbf{X}_{ff}$. Time-domain solutions show that for reasonably long wires, the two formulations yield the same results since

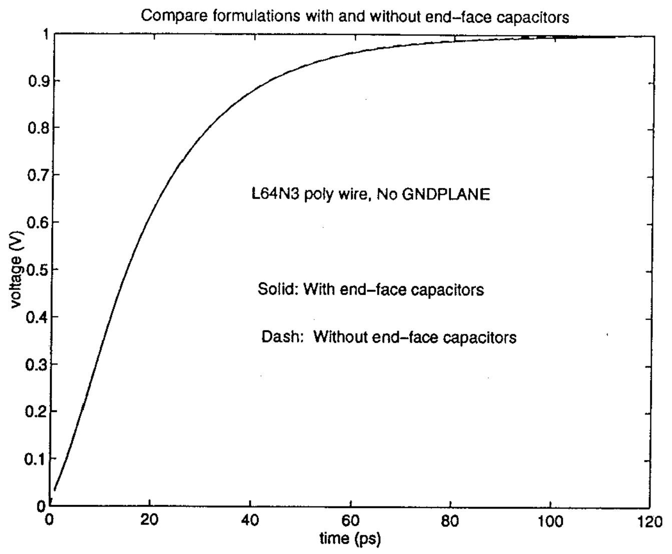


FIGURE 4-1: Compare formulations with and without contact-port capacitances

the capacitances associated with the contact ports are comparatively small. See Figure 4-1 for the far-end voltage waveforms computed for a length=64 wire, in the absence of a ground-plane, excited by a unit-step voltage source at the near-end.

Proceeding with the Arnoldi algorithm as in Section 4.2.1, the central task is to solve linear systems of the form $\mathbf{A} \cdot \mathbf{x} = \mathbf{RHS}$ for arbitrary right-hand-sides. Since the operator $\mathbf{A}$ has now a product form, it is easy to reduce its condition number by making the substitution

$$\mathbf{x} = \mathbf{X}_{ff}^{-1} \mathbf{\Pi}_{ff}^{-1} \cdot \mathbf{y}, \tag{4.20}$$

where $\mathbf{\Pi}_{ff}^{-1}$ is a *sparse* matrix approximation to the inverse of the *dense* matrix $\mathbf{P}_{ff}$, and is constructed by explicitly inverting local, overlapping blocks of $\mathbf{P}_{ff}$. For details on this computation, which fits naturally in the fast-multipole algorithm as demonstrated in the capacitance extraction program Fastcap, see [3]. The operator $\mathbf{X}_{ff}^{-1}$ is the exact inverse of $\mathbf{X}_{ff}$, and its action is effected by solving the interior Laplace problem with mixed boundary conditions: Dirichlet on the contact panels ($\Psi_c = 0$) and Neumann on the free panels ($\mathbf{J}_f^{\text{int}}$ arbitrary). As in the pure Dirichlet-to-Neumann problem $\mathbf{X}$, the mixed problem is solved by LU-factoring the associated sparse matrix generated from finite-differences. The free-panel potentials Ψ_f are computed, along with $\mathbf{J}_f^{\text{int}}$ as a by-product.

Using the *preconditioner* $\mathbf{X}_{ff}^{-1}\mathbf{\Pi}_{ff}^{-1}$, we apply GMRES to solve for $\mathbf{y}$ in

$$\frac{1}{4\pi\tau}(\mathbf{P}_{ff}\mathbf{\Pi}_{ff}^{-1}) \cdot \mathbf{y} = \mathbf{RHS}, \tag{4.21}$$

and then compute the final solution $\mathbf{x}$ by applying (4.20). This preconditioned Arnoldi algorithm is applied to the single-wire interconnect test case in Section 4.2.1. Table 4-1 displays the number of iterations, or matrix-vector product calculations, required for the iterative solution of (4.21). The rapid convergence shows that the condition number of the operator $\mathbf{P}_{ff}\Pi_{ff}^{-1}$ is much smaller than that of $\mathbf{A}$, and nearly independent of conductor length. The ill-conditioning caused by the wide range of time-constants has been removed by explicit solution of the interior problem $\mathbf{X}_{ff}^{-1}$, and the ill-conditioning caused by the proximity of conductors is removed by the overlapping preconditioner Π_{ff}^{-1}.

We make a note here that the starting Arnoldi vector $\mathbf{p}$ in (4.4) is computed by $\mathbf{p} = -\mathbf{X}_{ff}^{-1}\mathbf{X}_{fc} \cdot \mathbf{v}$ rather than an iterative solve involving $\mathbf{A}$. Hence, a q-order reduced model requires q GMRES iterative solutions rather than $(q+1)$ solutions.

4.3 Ground-plane Implementation

The potential variation of the grounded silicon substrate is typically of the order of tens of milivolts due to the many local, grounded body-plugs. Since this is small compared with the 3-volt or 5-volt power supply, we will assume an *ideal ground-plane* in this work. To include the ground-plane in the preconditioned formulation of the previous section, the only modification to make is the charge-to-potential operator $\hat{\mathbf{P}}_{ff}$. The interior Dirichlet-to-Neumann operator $\mathbf{X}$ remains unaffected. Let the ground-plane be approximated by a finite sheet, and assume it is explicitly discretized into N_g panels. Let $\Psi_g \in \Re^{N_g}$ be the vector of ground panel potentials, and let $\mathbf{J}_g \in \Re^{N_g}$ be the vector of corresponding panel currents. To include the ground-plane, additional terms are introduced into (4.16)

$$\begin{bmatrix} \hat{\mathbf{P}}_{ff} & \hat{\mathbf{P}}_{fg} \\ \hat{\mathbf{P}}_{gf} & \hat{\mathbf{P}}_{gg} \end{bmatrix} \begin{bmatrix} \mathbf{J}_f^{\text{int}} \\ \mathbf{J}_g \end{bmatrix} = -\frac{d}{dt} \begin{bmatrix} \Psi_f \\ \Psi_g \end{bmatrix}, \tag{4.22}$$

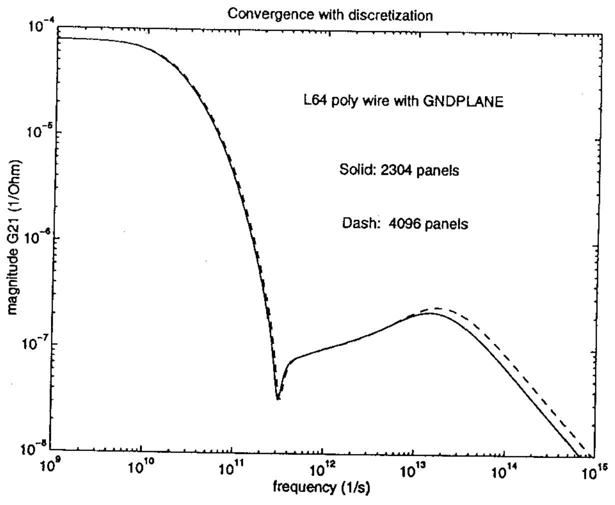
where $\hat{\mathbf{P}}_{fg} \in \Re^{N_f \times N_g}, \hat{\mathbf{P}}_{gf} \in \Re^{N_g \times N_f}, \hat{\mathbf{P}}_{gg} \in \Re^{N_g \times N_g}$ describe capacitive interactions among conductor surfaces and the ground-plane, and are similarly defined as $\hat{\mathbf{P}}_{ff}$. The condition $\frac{d}{dt}\Psi_g(t) = 0$ in the dynamic equation (4.22) implies that

$$\frac{d}{dt}\Psi_f = -\hat{\mathbf{P}}_{ff}^{\text{G}} J_f, \tag{4.23}$$

$$\hat{\mathbf{P}}_{ff}^{\text{G}} = \hat{\mathbf{P}}_{ff} - \hat{\mathbf{P}}_{fg}\hat{\mathbf{P}}_{gg}^{-1}\hat{\mathbf{P}}_{gf}, \tag{4.24}$$

where $\hat{\mathbf{P}}_{ff}^{\text{G}} \in \Re^{N_f \times N_f}$ is the new charge-to-potential operator in the presence of a ground-plane. Since N_g may be large, it is impractical to factor $\mathbf{P}_{gg}$, and since $\hat{\mathbf{P}}_{ff}^{\text{G}}$ is applied multiple times in an iterative solve, it is impractical to apply $\mathbf{P}_{gg}^{-1}$ via an inner-loop iterative solve. Hence we will use the *method-of-images* [25] to apply the operator $\hat{\mathbf{P}}_{ff}^{\text{G}}$ Fictitious image charge panels are created by reflecting real charge panels across the ground-plane, and are always assigned the opposite charge. A similar procedure applies to the overlapping preconditioning operation. Since the $O(N)$ fast-multipole algorithm is used, the net cost is twice that of the problem without the ground-plane.

It would also be possible to use precorrected-FFT methods with a modified Greeen's function to include the ground-plane [33].



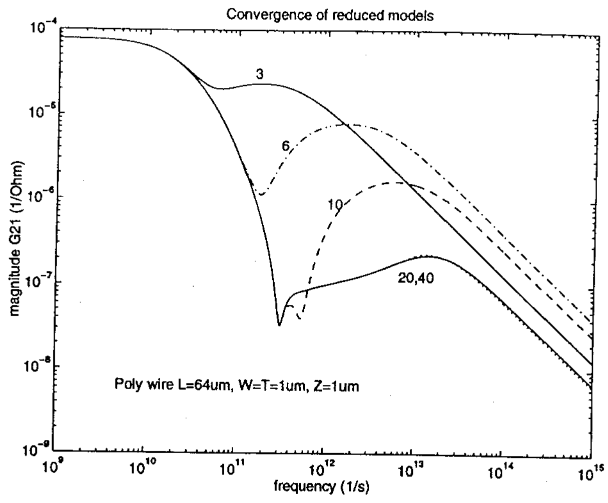


FIGURE 4-2: Solution converges with discretization

FIGURE 4-3: Reduced-order models converge in frequency domain

4.4 Model-Order Reduction Results

In this section, we present numerical results of our multipole-accelerated, preconditioned model-order reduction algorithm and demonstrate its accuracy and efficiency. Throughout, polysilicon conductivity and oxide permittivity will be assumed unless otherwise noted. The groundplane is also included in all following examples. Figure 4-2 shows the frequency response for a 2-port, computed from the full model, for a rectangular conductor with aspect ratios $1\mu m \times 1\mu m \times 64\mu m$, sitting one micron above the groundplane. Two discretizations are used: the coarser one divides each unit square into 9 equal panels, and the finer one divides each unit square into 16 panels. It is seen that up to a frequency of 10 Terahertz (10^{13}) the results for the two discretizations are nearly identical. Henceforth we shall use the coarser discretization.

Figure 4-3 is a plot of the frequency response of the reduced-order models for the same conductor and shows that a twentieth-order model produces virtually identical results as the full model, of order 2,304. Time-domain data generated by the full-order models and the reduced-order models are also given for comparison. Figure 4-4 displays the short-circuit current in port 2 (held at ground), and Figure 4-5 displays the open-circuit voltage at port 2, for various reduced models; the excitation in both is a unit-step voltage source at port 1. We see from Figures 4-3 and 4-4 that third-order models are accurate enough if there are no signals in the system faster than 10 $\sim$ 30 picoseconds. Figure 4-5 shows that the third-order model captures most of the essential features of the true response, while the first-order model, which is equivalent to a single-lumped RC model, fails miserably.

Next, we perform two-conductor coupling experiments using the same configuration as in Figure 3-4, with the driven conductor connected to a voltage source and the "victim" conductor grounded at the near ends. We are interested in the voltage noise $v2$. Both wires have

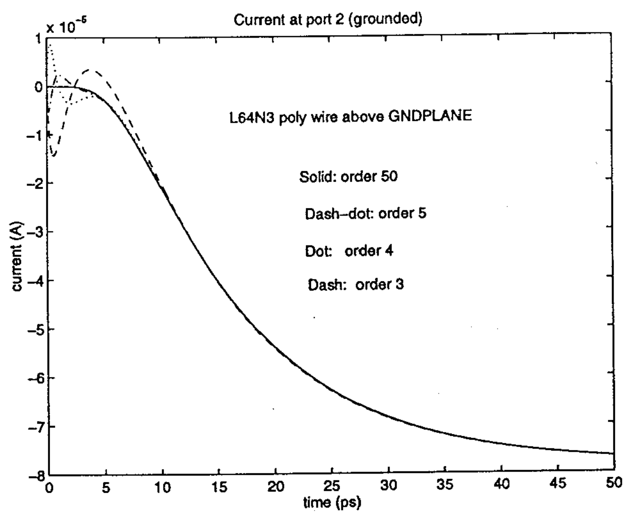


FIGURE 4-4: Port 2 current vs. time computed with reduced models

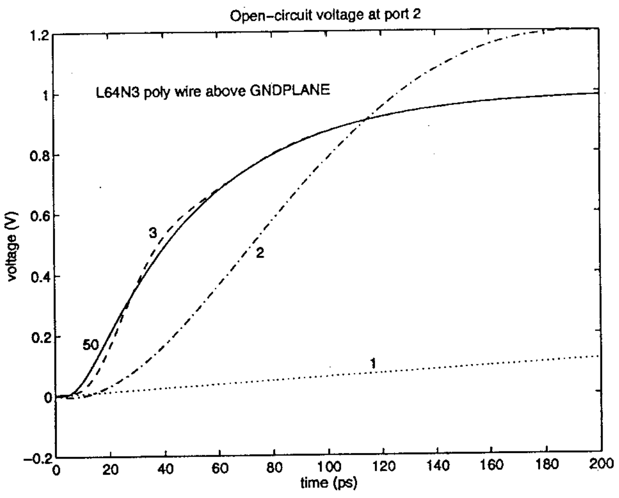


FIGURE 4-5: Port 2 voltage vs. time computed with reduced models

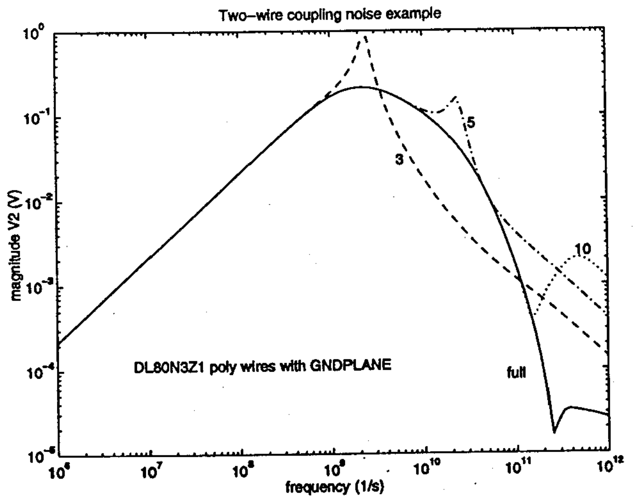


FIGURE 4-6: Poly-to-poly coupling voltage noise in frequency domain

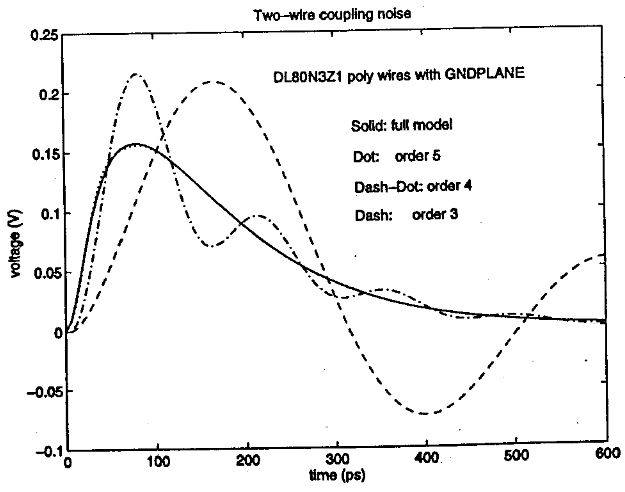


FIGURE 4-7: Poly-to-poly coupling voltage noise in time domain

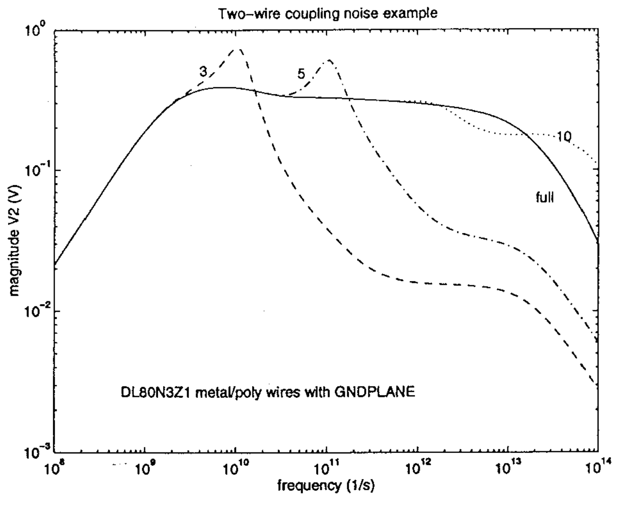


FIGURE 4-8: Metal-to-poly coupling voltage noise in frequency domain

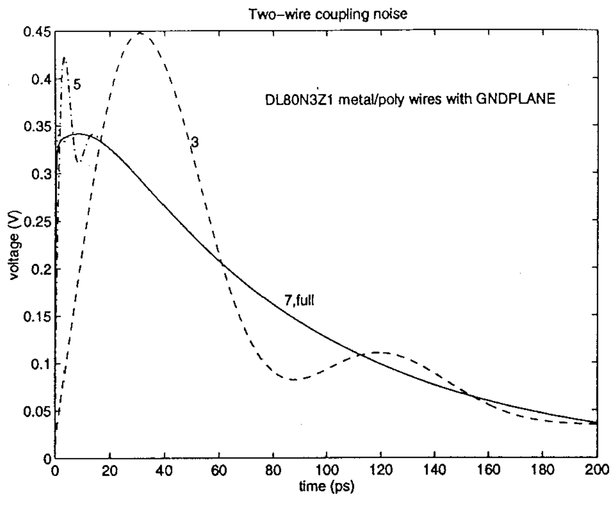


FIGURE 4-9: Metal-to-poly coupling voltage noise in time domain

dimensions $1\mu m \times 1\mu m \times 80\mu m$, and sit 1 micron above the groundplane. Figure 4-6 shows the magnitude of $v2$ in the frequency domain, generated from several reduced models. It is seen that a fifth-order model is necessary to capture the full model up to 10 GHz. Figure 4-7 shows the time-domain response $v2$ to a unit-step voltage source. The fifth-order model is nearly indistinguishable from the full model. Similar experiments were performed with the driven polysilicon line replaced by aluminum, and the victim line material unchanged. Results are shown in Figures 4-8 and 4-9. The metal line introduces a much smaller timescale due to its high conductivity, and as a result the coupling noise remains significant up to a much higher frequency, 10 THz. The time-response also shows a much faster risetime. For purposes of SPICE-level simulation in which the excitation is bandwidth-limited to say, below 10 GHz, a fifth-order model is sufficient.

Next, we apply our algorithm to two large interconnect examples. The first is the six-conductor SRAM structure shown in Figure 3-8. The structure is treated as a six-port problem, with the excitation ports labeled 1-6 in the figure. The simulation is now run with the groundplane with its approximate position shown in the figure. Refer to Section 3.3 for the discretization scheme and labeling. Total panel counts, including real and image panels, are shown in Table 4-2. Figure 4-10 shows the frequency response of the conductance G_{61} computed using various discretizations. It is seen that up to 10^{13} Hz, the results are nearly identical for the mesh refinements $m = 3$ and $m = 4$; refer to Section 3.4 for the definition of the mesh parameter m. The coaser meshes, $m = 1$ and $m = 2$ may be used for quick estimates. The model-order reduction results for $m = 4$ is plotted in Figure 4-11, which shows that a sixth-order model is necessary to capture the first *knee* in the frequency response at $\sim$ 100 GHz. We make a note here that the straight-line section of Figure 4-11 corresponds to the low-frequency

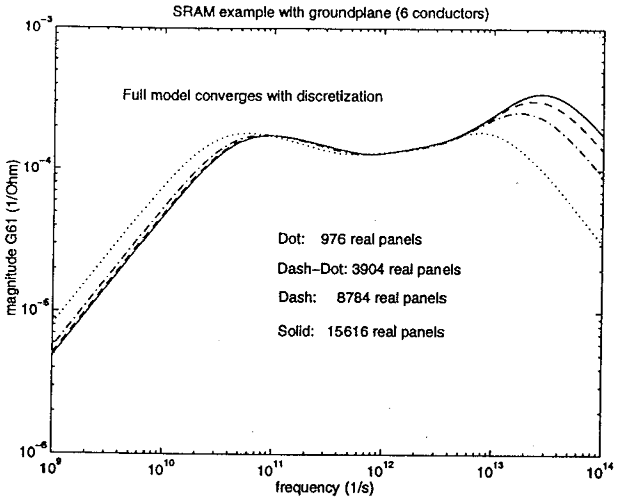


FIGURE 4-10: Convergence with discretization for SRAM

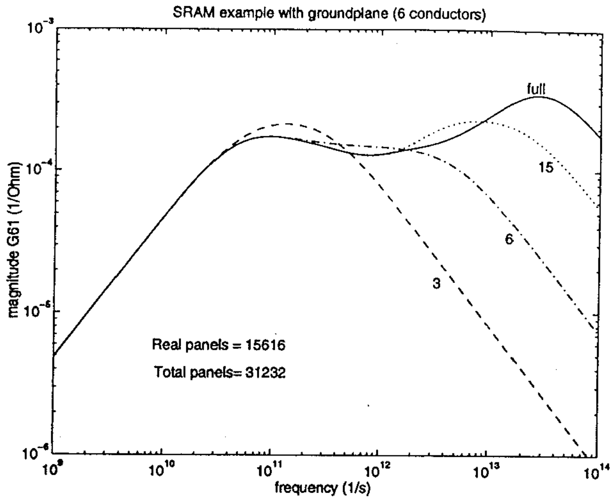


FIGURE 4-11: Reduced-order models for SRAM

approximation

$$\mathbf{Y} = j\omega\mathbf{C}, \tag{4.25}$$

where $\mathbf{Y}, \mathbf{C}$ are the admittance and capacitance matrices, respectively. Resistance plays no part in the interconnect conductance until the higher-frequency components are excited.

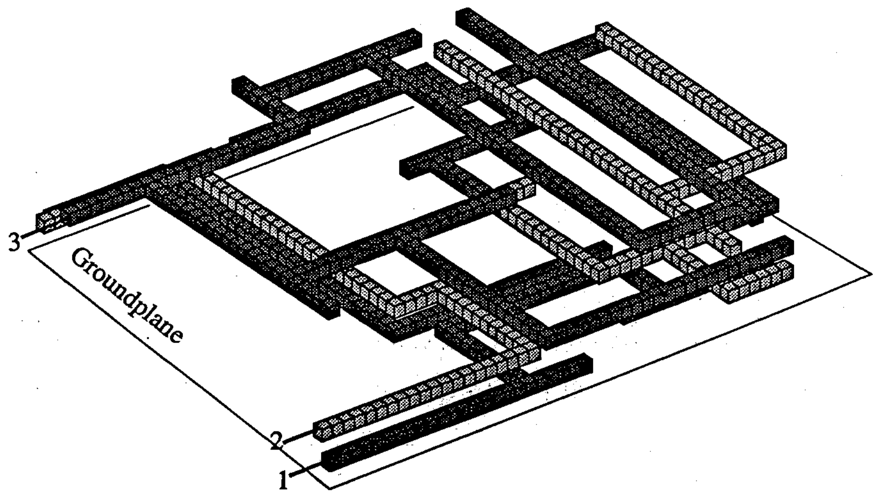


FIGURE 4-12: Three-level interconnect (m=1 mesh).

m	1	2	3	4
SRAM	1,952	7,808	17,568	31,232
3-level	5,078	20,312	45,702	NA

Table 4-2: Total (real+image) panel count.

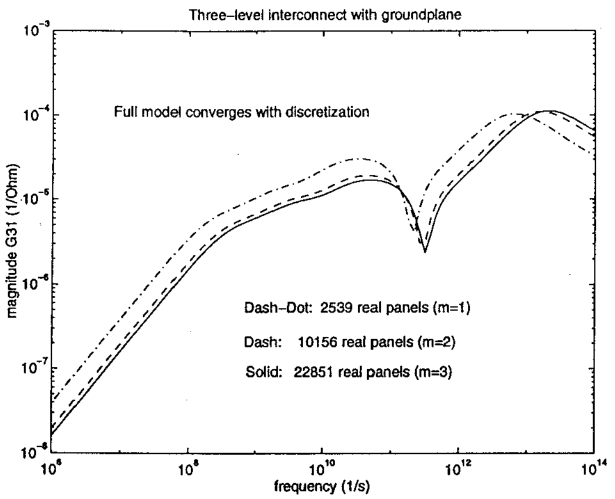


FIGURE 4-13: Convergence with discretization for three-level interconnect

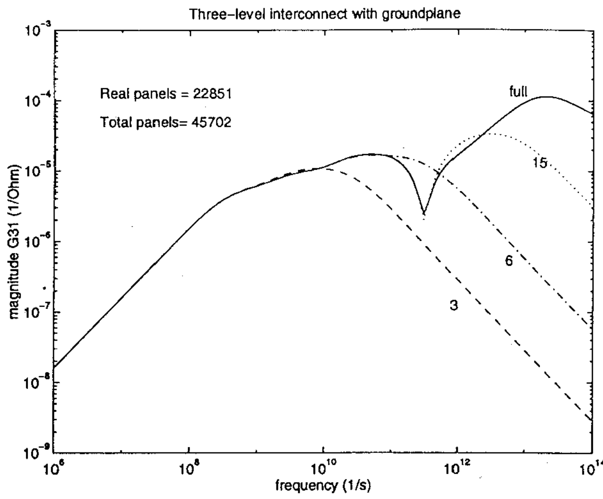


FIGURE 4-14: Reduced-order models for three-level interconnect

A three-level interconnect structure with a coarse discretization ($m = 1$) is shown in Figure 4-12, in which each unit square is $1\mu m \times 1\mu m$, and the groundplane is one micron below the bottom-level interconnect. Polysilicon is used for the bottom level, and aluminum for the top two. Each interconnect layer is excited at a single port, as shown in the figure. In this example, we compute the first column of the admittance matrix by connecting port 1 to a voltage source and grounding ports 2 and 3. Three discretizations were used ($m = 1, 2, 3$), and the total panel count is shown in Table 4-2.

Figure 4-13 shows the convergence with discretization of the frequency response for G_{31}, and Figure 4-14 shows results of reduced-order modeling. Although the frequency dependence is complicated, a third-order model is accurate up to 1 GHz, and a sixth-order model accurate up to 10 GHz.

To demonstrate that the entire multipole-accelerated, preconditioned model-order reduction algorithm has order N complexity, we plot CPU time and memory used versus the total number of panels (real and image) in computing a single-input-multiple-output (SIMO), sixth-order model for the SRAM and the three-level interconnect examples. See Figures 4-15 and 4-16.

Since a SIMO reduced-order model corresponds to one column of the frequency-dependent admittance matrix, we compare this cost to that of computing one column of the capacitance matrix using FASTCAP [3], a multipole-accelerated capacitance extraction program. Table 4-3 displays the ratio of the CPU times. While a sixth-order model essentially solves the capacitance problem six times, the actual CPU time overhead is seen to be only a factor of two to three. This is because the significant set-up time associated with the multipole algorithm, common to both procedures, is better amortized in the reduced-order model computation. Memory requirements are nearly identical in both cases.

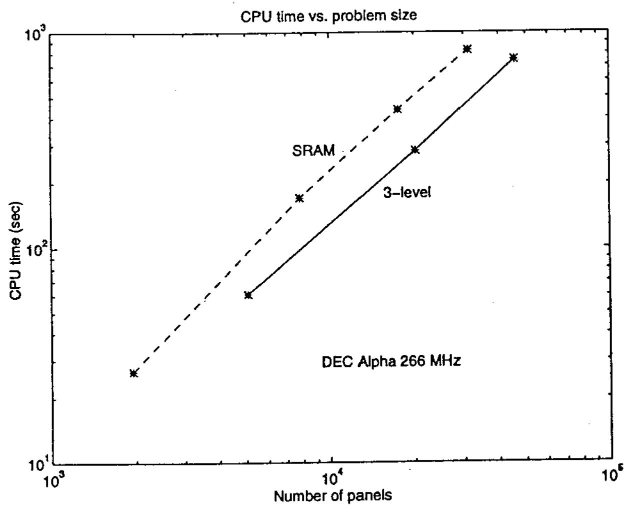


FIGURE 4-15: CPU time grows linearly with problem size

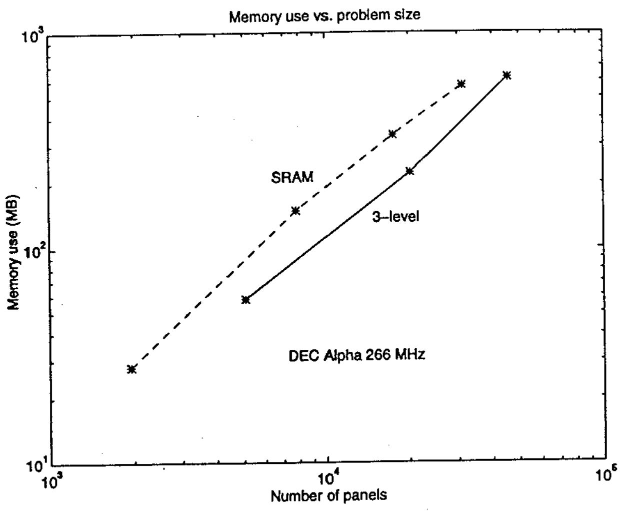


FIGURE 4-16: Memory use grows linearly with problem size

	m=1	m=2	m=3	m=4
3-level	1.6	1.6	2.4	NA
SRAM	1.8	2.2	3.0	3.1

Table 4-3: Ratio of reduced-model to capacitance-extraction CPU times.

5

Comparing Diffusion and 3D Models

For analyzing two-dimensional interconnect problems, such as a single, long wire or a collection of parallel wires, the *diffusion equation* equation is often used in the electro-quasistatic approximation, or RC regime. In the context of interconnect analysis, the basic, single-conductor diffusion equation can be written as

$$\mathcal{R}\mathcal{C}\,\frac{\partial}{\partial t}\Phi(x,t) = \frac{\partial^2}{\partial x^2}\Phi(x,t), \tag{5.1}$$

where $\mathcal{R}, \mathcal{C}$ are the resistance and capacitance, respectively, per unit length, and $\Phi(x,t)$ is the electric potential along the wire as a function of position and time. Equation (5.1) can be easily derived by taking the continuous limit of the discrete RC ladder circuit shown in Figure 5-1. Similarly, numerical solutions which are accurate up to a given excitation frequency can be obtained from a circuit solution of the discrete RC ladder network if a large enough number of sections, or lumps, are used.

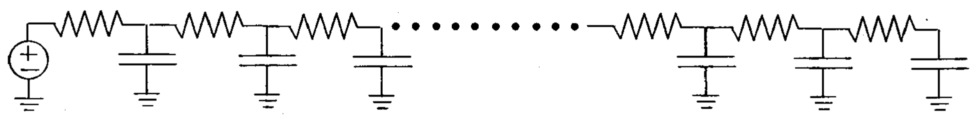


FIGURE 5-1: RC ladder circuit

The diffusion model differs from the full three-dimensional model in several ways. First, the diffusion equation assumes capacitive coupling only between each node and the groundplane, and not among the nodes themselves, whereas the three-dimensional picture models capacitive coupling among all panels as well as the groundplane. Figure 5-2 displays the capacitive interaction between one particular node and all other nodes. Secondly, the diffusion picture models only current flow parallel to the wire, whereas the three-dimensional picture models current flow in all three directions in the conductor, produced by possible potential differences in the

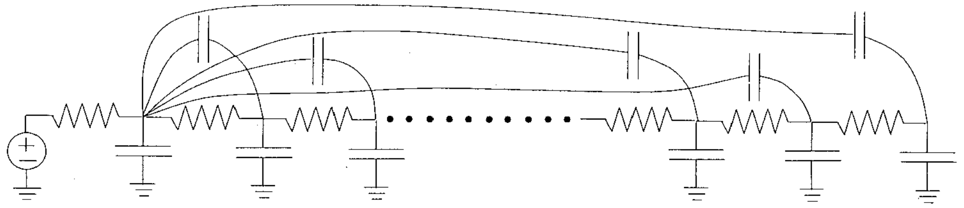


FIGURE 5-2: Global capacitive coupling model

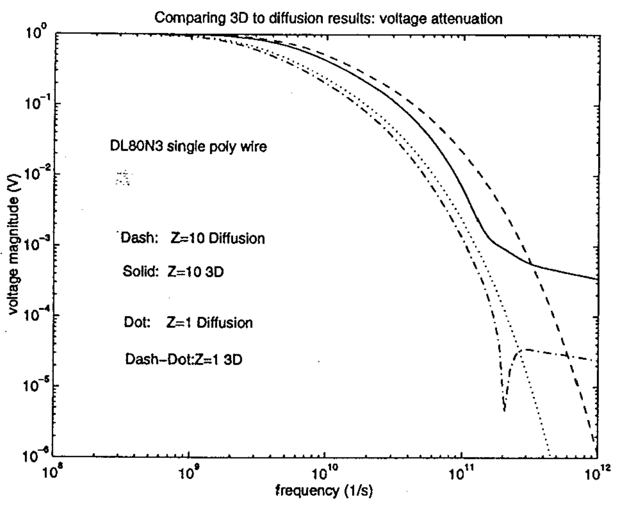


FIGURE 5-3: Voltage attenuation: diffusion model vs. 3D

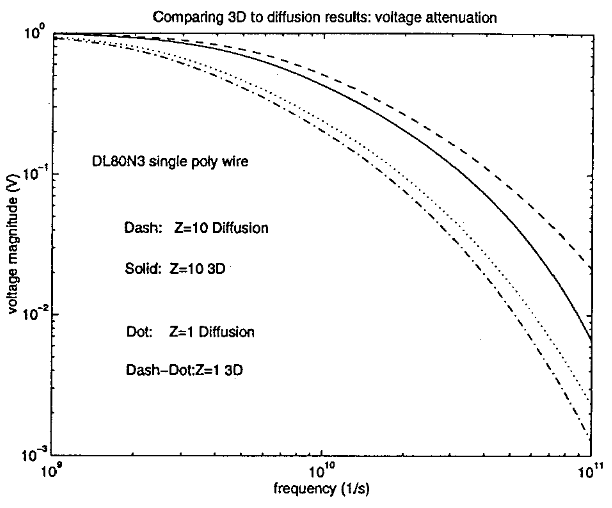


FIGURE 5-4: Diffusion model less accurate for large Z

transverse directions. Results from the two models approach each other as the conductor approaches the groundplane and as the excitation frequency is lowered, since the former effectively reduces the relative strength of panel-to-panel interactions, and the latter makes the conductor potential more uniform in both transverse and longitudinal directions.

We present results from numerical experiments which compare the diffusion and the three-dimensional models. A three-dimensional capacitance extraction is performed on a single, rectangular wire over a groundplane, using the mesh in Figure 3-1. A long enough wire is used to ensure that the capacitance per unit length is within one percent of the long-wire limit. This capacitance value, along with the wire resistance computed from its cross sectional area, is used in the diffusion model. For all experiments in this section, the conductor dimensions are $80\mu m \times 1\mu m \times 1\mu m$, with a distance above the groundplane at $1\mu m (Z = 1)$ or $10\mu m (Z = 10)$. First, we perform the single-wire experiment, in which the near-end of the conductor is excited by a voltage source, and the resulting far-end voltage is measured as a function of frequency. The set-up is similar to that in Figure 3-1. The results from the diffusion and 3D model are shown in Figure 5-3. A close-up view is shown in Figure 5-4. It is seen that up to a frequency of 100 GHz (10^{11}Hz), the diffusion results give a fair approximation to the 3D results. Also, the approximation becomes worse as the distance between the wire and groundplane is increased.

Next, we perform the two-conductor coupling experiment in a set-up similar to that of Figure 3-4. The parameters used for the diffusion model are extracted from the two-conductor capacitance matrix, computed with the groundplane included, and the coupled diffusion equation is solved numerically with the coupled RC ladders shown in Figure 5-5. The noise voltage $v2$ is plotted as a function of excitation frequency for the case $Z = 1$ in Figure 5-6, and a magnified view is shown in Figure 5-7 for both the $Z = 1$ and $Z = 10$ cases. The same observations can be made here as in the single-wire experiments. The low-frequency, straight-line section in the figures correspond to the capacitive limit described by (4.25), where the wire resistance plays no role.

We conclude from the above experiments that the diffusion model and the 3D model yield similar results when the conductors are in close proximity to the groundplane, in which case the relative importance of global capacitive coupling is minimized.

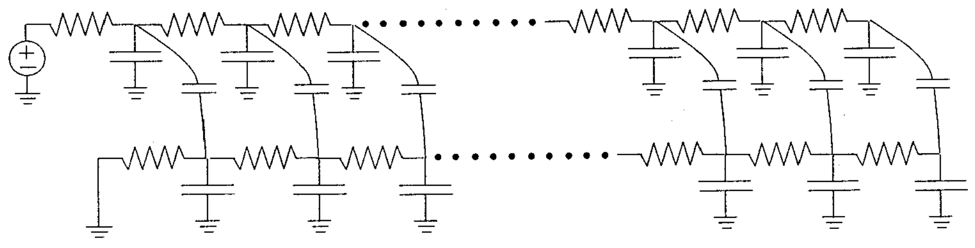


FIGURE 5-5: Coupled RC ladders

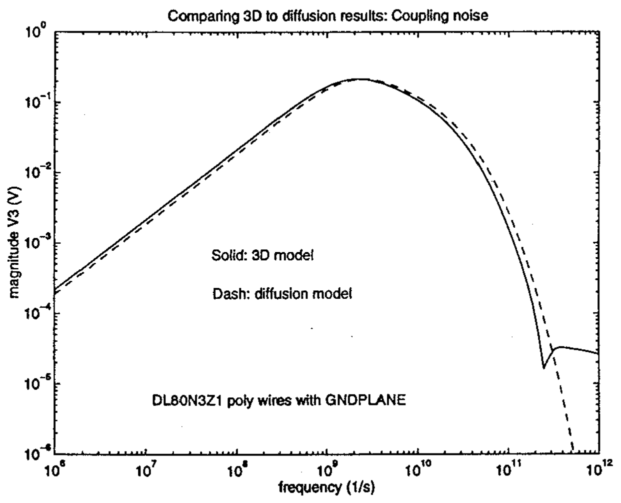


FIGURE 5-6: Coupling noise(Z=1): diffusion model vs. 3D

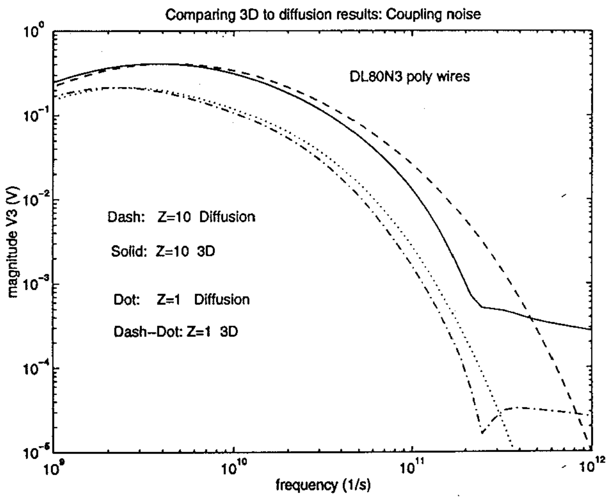


FIGURE 5-7: Diffusion model less accurate for large Z

Part II

The Substrate Coupling Problem

6

Overview of the Substrate Coupling Problem

The design of single chip mixed-signal systems which combine both analog and digital functional blocks on a common substrate is now an active area of research, driven by the relentless quest for high-level integration and cost reduction. A major challenge for mixed-signal design tools is the accurate modeling of the parasitic noise coupling through the common substrate between the high-speed digital and high-precision analog components [34, 35, 36]. Fast switching logic components inject current into the substrate, causing voltage fluctuation which can affect the operation of sensitive analog circuitry through the body-effect, since the transistor threshold is a strong function of substrate bias. This coupling mechanism is illustrated in Figure 6-1, in which a switching digital node injects current J via the p-n junction into the bulk, causing the local substrate potential V_b to vary at an analog node. This interaction is also illustrated in Figure 6-2 from the circuit point of view.

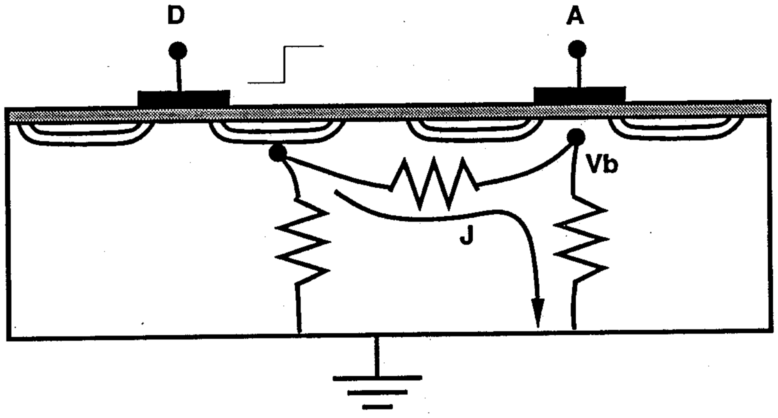


FIGURE 6-1: Substrate coupling mechanism.

For the accurate modeling of substrate-coupled noise, several numerical schemes currently

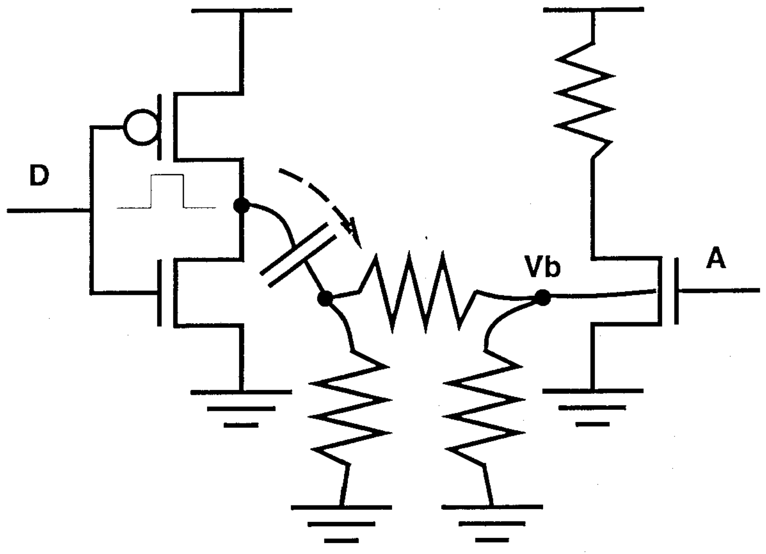


FIGURE 6-2: Substrate coupling from circuit point of view.

exist, but each has its limitations. Since it has been understood that the bulk substrate behaves resistively up to a frequency of a few gigahertz [37, 18], it is sufficient to solve Laplace's equation inside the substrate with proper boundary and interface conditions. Examples of this approach [35, 34, 38, 39, 40] includes Finite Element (FEM) and Finite Difference (FD) methods. These techniques perform a full domain discretization on the large but bounded substrate and can easily handle irregular substrates (wells, doping profiles, etc). Although the resulting linear systems are sparse, such methods are impractical for complex layouts because the number of unknowns resulting from three-dimensional volume-meshing of the entire substrate is too large.

Integral equation (IE) based techniques have been applied with some success to the modeling of substrate coupling [41, 42, 17]. By requiring only the discretization of the individual, two-dimensional substrate contacts, IE methods dramatically reduce the number of unknowns and hence the size of the linear system to be solved. The primary drawback to the integral formulation is that the resulting matrices are *dense*, which makes direct factorization impractical for problems with more than a few hundred unknowns. To address this difficulty, similar heuristic partitioning schemes were described in [43, 41] as an attempt to sparsify the matrix *inverse* by setting direct admittances to contacts outside a user-defined region to zero. While this approximation makes larger problems tractible, it requires too much user intervention and, more importantly, results in errors that are difficult to control and quantify.

Iterative algorithms form an attractive alternative to direct matrix factorization for large or dense linear systems. GMRES [9], a Krylov-subspace based iterative method similar to the well-known conjugate gradient technique, was used in [17] to solve the IE system in a matrix-free manner. Since only matrix-vector products are required to generate new search directions, a multipole-accelerated, "black-box" algorithm was formulated to perform the matrix-vector multiplication without having to explicitly compute or store the dense matrix. This allows all direct and indirect substrate contact-to-contact interactions to be included. However, the major difficulty with conjugate gradient style iterative methods is slow convergence when applied to large IE systems, which tend to be ill-conditioned [28, 14, 5]. Hundreds of matrix-vector products may be required per solution for large problems.

Multigrid methods, or more generally, multilevel methods, are well-developed and known to be the most efficient iterative techniques in the solution of elliptic partial differential equations (PDE's) [44, 45, 46] due to their fast convergence. More specifically, such methods can yield convergence rates independent of problem size and matrix condition [47, 45]. However, multilevel methods are not well-developed for the solution of first-kind integral equations [5] defined over complicated surfaces, as is the case here for the integral formulation of the substrate coupling problem. In this thesis, we address this void by developing the many algorithmic components necessary for a sparsified, multigrid iterative solution of such IE systems. We then demonstrate that the resulting convergence rate is independent of problem size and similar to those for PDE's.

Our multigrid development for the substrate coupling problem is organized as follows. Chapter 7 summarizes the integral equation formulation for substrate coupling resistance extraction. Chapter 8 reviews some basic ideas from function analysis and the theory of integral equations, which are then used to motivate our multiresolution, or multigrid analysis. A sparsification method based on eigenanalysis is presented in Chapter 9. For solving first-kind integral equations defined over regular domains, a novel multigrid method is developed in Chapter 10. Chapter 11 then extends this multigrid algorithm to solving equations defined over complicated geometries, such as a typical IC layout. Computational results are given in Chapter 12, where comparisons to conjugate gradient style methods are also made.

7

Background and Previous Work

This chapter reviews the integral equation (IE) formulation for the mixed-signal substrate coupling problem. The eigenfunctions and Green's functions for the integral equation are also introduced. The notation and convention introduced here will be used throughout the rest of the thesis.

7.1 Integral Formulation

For typical mixed-signal circuits operating at frequencies below a few gigahertz, the substrate behaves resistively [37, 17]. Assuming this electrostatic approximation, the substrate is therefore modeled as a stratified medium composed of several homogeneous layers characterized by their conductivities, as shown in Figure 7-1. Three substrate contacts are shown in gray. For this work, the substrate backplane is assumed to be grounded electrically. The governing equation in the electrostatic case is Poisson's equation

$$-\nabla \cdot (\sigma_i \nabla \phi(\mathbf{r})) = \rho(\mathbf{r}) \tag{7.1}$$

where ϕ is the electrostatic potential, $\mathbf{r}$ is the position vector, σ_i is the conductivity associated with the i-th layer, and ρ is the current flux density $\rho = \nabla \cdot J$.

Since current is injected only from the substrate contacts on the top surface, Laplace's equation, where the right-hand side of (7.1) is zero, holds in the interior of the substrate. Thus, if the Green's function $G(\mathbf{r};\mathbf{r}')$ satisfying

$$-\nabla \cdot (\sigma_i \nabla G(\mathbf{r};\mathbf{r}')) = \delta(\mathbf{r}') \tag{7.2}$$

and appropriate boundary and interface conditions can be efficiently computed, an integral equation defined over S, the collection of two-dimensional contact surfaces, can be written

$$\phi(\mathbf{r}) = \int_S \rho_S(\mathbf{r}')G(\mathbf{r};\mathbf{r}')da', \qquad \mathbf{r} \in S, \tag{7.3}$$

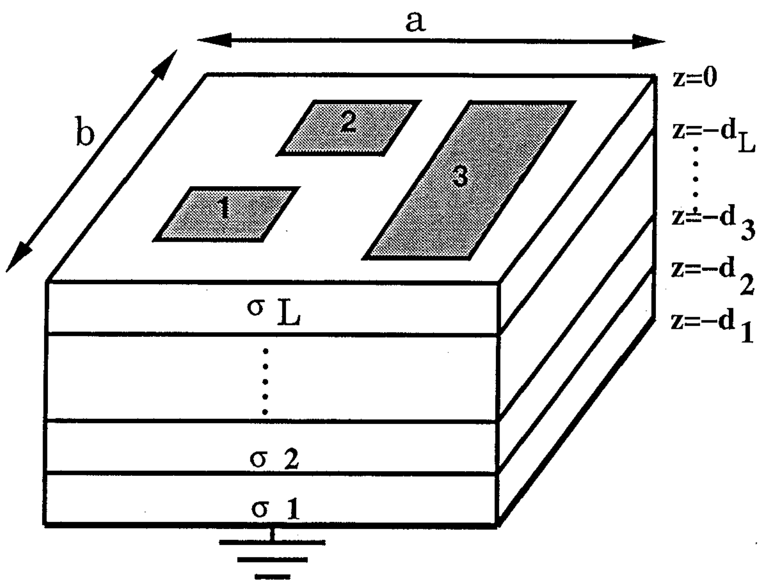


FIGURE 7-1: 3D substrate profile.

where $\mathbf{r}, \mathbf{r}'$ are now points on S, and ρ_S is the current density on S. This is a first-kind integral equation [5] which forms the basis for the numercial techniques used in [41, 43, 17]. Figure 7-2 illustrates a situation where the Green's function $G(\mathbf{r}; \mathbf{r}')$ is to be evaluated for all pairs $(\mathbf{r}; \mathbf{r}')$ on the two substrate contacts, colored in gray.

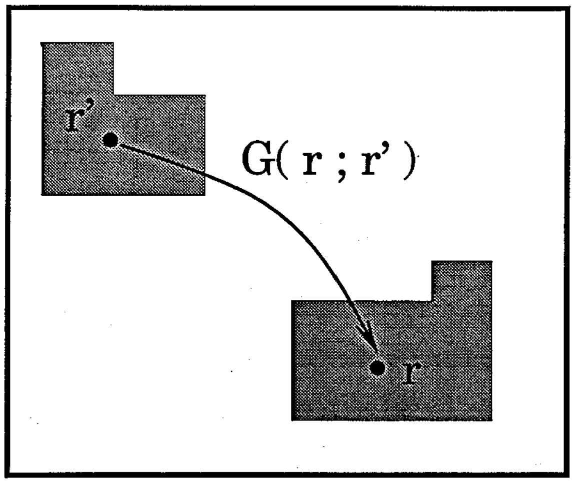


FIGURE 7-2: Green's function to be evaluated.

To numerically solve (7.3), the domain S is broken up, or discretized, into a collection of N disjoint, rectangular *panels* $\{p_i\}$ such that $S = \bigcup_{i=1}^{N} p_i$. An example of panel discretization for a three-contact layout is given in Figure 7-3. In the piece-wise constant Galerkin scheme [6], it is assumed that the current density $\rho_S(\mathbf{r})$ on each panel p_i is uniform and equal to ρ_i. Then N linear equations are constructed by evaluating the *average* of the potential $\phi(\mathbf{r})$ over each panel

p_i. The Galerkin method yields a discretized version of (7.3)

$$Pq = v \tag{7.4}$$

where q and v are length-N vectors with q_i denoting the total current on panel i and v_j denoting the average potential on panel j. P is an $N \times N$ matrix, with elements P_{ij} given by

$$P_{ij} = \frac{1}{a_i a_j} \int_{p_i} \int_{p_j} G(\mathbf{r}; \mathbf{r}') da da' \tag{7.5}$$

where a_i and a_j are the surface areas of panels i and j respectively. P is often called the *coefficient-of-potential* matrix. We note here that P is *dense* since current injected into panel i will produce a non-zero potential at panel j.

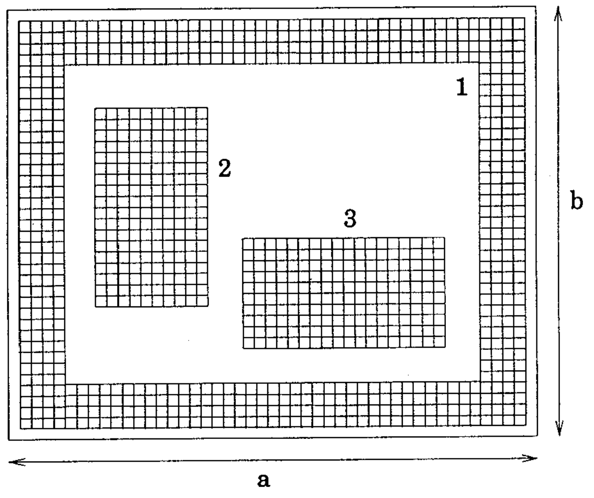


FIGURE 7-3: Example of contact discretization.

Let n be the number of contacts, or nodes, in the substrate layout. Typically $n << N$. The aim of substrate coupling extraction is to derive a *macromodel* in the form of a *conductance matrix* which models the substrate current flow completely from the point of view of the n nodes or ports. For example, the three-contact problems shown in Figure 7-1 and 7-3 can be modeled with six conductances as shown in Figure 7-4. The resulting macromodel is

$$Q = GV, \tag{7.6}$$

where $G \in \Re^{n \times n}$ is the conductance matrix, $Q \in \Re^n$ specifies the total current Q_i for each node i, and $V \in \Re^n$ specifies the voltage V_j for each node j. The matrix

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1n} \\ g_{21} & g_{22} & \cdots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nn} \end{bmatrix}. \tag{7.7}$$

is symmetric and diagonally dominant. If a direct connection exists between each node and ground (*i.e.* if the backplane is grounded), then G is also strictly diagonally dominant. To extract the i-th column of G, the $N \times N$ linear system (7.4) is solved for the specific right-hand side, or detailed potential distribution v, in which all panels belonging to the i-th contact are set to one volt, while all other panels are set to zero volt. After the detailed current distribution q is obtained, the element g_{ki} can be computed by summing the panel currents in q corresponding to the k-th contact. Thus, to derive the $n \times n$ macromodel G, (7.4) is to be solved n times for q given v. This is analogous to the capacitance extraction problem [4, 3].

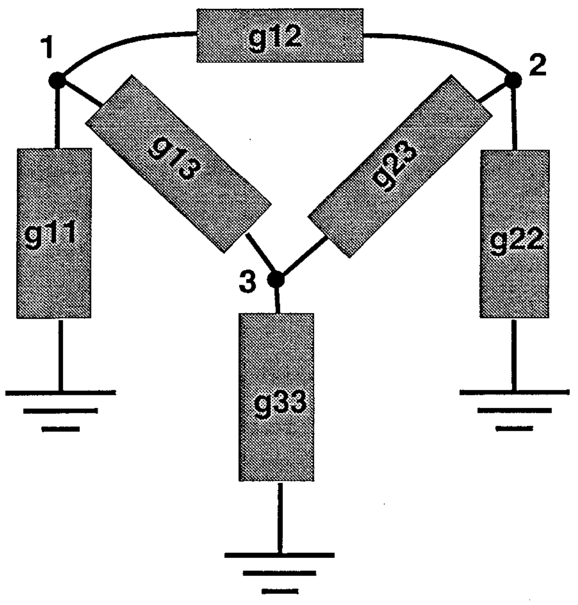


FIGURE 7-4: Three-node macromodel for substrate.

We make the special note here that a purely *resistive* macromodel of the substrate is sufficient in the case where the dielectric relaxation time constant $\tau = \epsilon/\sigma$ [48] is much smaller than the typical time scales of the circuit. This is true for most substrates operating below a few Gigahertz, as shown in [37, 17]. However, the RC time constants resulting from substrate interactions with junction and gate capacitances are certainly not negligible. These capacitances, typically nonlinear, are much more easily handled by a circuit simulator such as SPICE. Hence, the nodes in Figure 7-4 correspond to *bottom* plates of junction or gate capacitances, as suggested in Figure 6-1. Reduced-order models which include the effects of junction capacitances have been proposed in [40]. These macromodels are more complicated than the conductance matrix G describes above, and may be less accurate when nonlinear effects of the junctions become significant.

7.2 Green's Function based Framework

The Green's function $G(\mathbf{r};\mathbf{r}')$ specific to the substrate coupling problem must satisfy Poisson's equation

$$-\nabla\cdot(\sigma(\mathbf{r})\nabla G(\mathbf{r};\mathbf{r}')) = \delta(\mathbf{r}'). \tag{7.8}$$

In addition, the appropriate interface conditions

$$\begin{aligned} \sigma_{i-1}\cdot\frac{\partial G(\mathbf{r};\mathbf{r}')}{\partial z}\Big|_{z\to -d_i^-} &= \sigma_i\cdot\frac{\partial G(\mathbf{r};\mathbf{r}')}{\partial z}\Big|_{z\to -d_i^+} \\ G(\mathbf{r};\mathbf{r}')\Big|_{z\to -d_i^-} &= G(\mathbf{r};\mathbf{r}')\Big|_{z\to -d_i^+} \end{aligned} \tag{7.9}$$

must be satisfied at each interfaces $z = -d_i$ between layers with conductivities σ_{i-1} and σ_i. This ensures that both potential and normal current flow are continuous across layer boundaries. See Figure 7-1. Finally, the boundary conditions

$$\begin{aligned} \frac{\partial G(\mathbf{r};\mathbf{r}')}{\partial n} &= 0, \quad r\in \text{top face or side faces} \\ G(\mathbf{r};\mathbf{r}') &= 0, \quad r\in \text{ bottom face} \end{aligned} \tag{7.10}$$

where n is the unit outward normal at the boundaries, must be satisfied by G. This enfores that the normal current flow is zero (Neumann B.C.) at the top and side faces of the substrate, and that the substrate bottom contact remains at ground (Dirichlet B.C.). For the case in which the backplane is *floating*, a zero-Neumann B.C. is applied to the bottom face as well as the side faces, leading to a *modified* Green's function [41, 49]. In this thesis, we shall focus on the case with the backplane grounded.

Since the Green's function $G(\mathbf{r};\mathbf{r}')$ is to be evaluated for target $\mathbf{r}$ and source $\mathbf{r}'$ points on the two-dimensional substrate contacts on the top surface (as shown in Figure 7-2), we make the substitution $\mathbf{r}\to(x,y)$ and $\mathbf{r}'\to(x',y')$. The problem is reduced to two-dimensions, where $z = z' = 0$ is implied below. The Green's function satisfying (7.8),(7.9), and (7.10) is shown in [37, 43] to be an infinite series

$$G(x,y;x',y') = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} C_{mn}\, f_{mn}\, \cos\left(\frac{m\pi x}{a}\right)\cos\left(\frac{m\pi x'}{a}\right)\cos\left(\frac{n\pi y}{b}\right)\cos\left(\frac{n\pi y'}{b}\right) \tag{7.11}$$

where a, b are the lateral substrate dimensions, and x, y are the cartesian coordinates. From (7.11), it is clear that $G(\mathbf{r};\mathbf{r}')$ does not have *translational invariance*. This means that in general, $G(\mathbf{r};\mathbf{r}')\neq G(\mathbf{r}+\Delta;\mathbf{r}'+\Delta)$. The scaling constants C_{mn} are defined by

$$C_{mn} = \begin{cases} 1/(ab) & m=0,\ n=0 \\ 2/(ab) & m=0,\ n>0 \ \text{ or } \ m>0,\ n=0 \\ 4/(ab) & m>0,\ n>0 \end{cases} \tag{7.12}$$

The coefficients f_{mn} take into account the vertical dimensions and the layer conductivities, and can be computed with the help of recursion formulas. For the sake of completeness, we summarize the procedure derived in [37] for computing f_{mn}. First, define

$$\gamma_{mn} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}. \tag{7.13}$$

Assume there are a total of L layers, as shown in Figure 7-1, and let $d \equiv d_1$ be the substrate thickness. For m, n not both zero, we then compute the quantities $\beta_{mn}^{(L)}$ and $\Gamma_{mn}^{(L)}$ recursively from the relation

$$\begin{bmatrix} \beta_{mn}^{(k)} \\ \Gamma_{mn}^{(k)} \end{bmatrix} = \begin{bmatrix} \frac{\sigma_{k-1}}{\sigma_k}\cosh^2(\Theta_{mn}^{(k)}) - \sinh^2(\Theta_{mn}^{(k)}) & , \left(\frac{\sigma_{k-1}}{\sigma_k} - 1\right)\cosh(\Theta_{mn}^{(k)})\sinh(\Theta_{mn}^{(k)}) \\ \left(1 - \frac{\sigma_{k-1}}{\sigma_k}\right)\cosh(\Theta_{mn}^{(k)})\sinh(\Theta_{mn}^{(k)}) & , \cosh^2(\Theta_{mn}^{(k)}) - \frac{\sigma_{k-1}}{\sigma_k}\sinh^2(\Theta_{mn}^{(k)}) \end{bmatrix} \cdot \begin{bmatrix} \beta_{mn}^{(k-1)} \\ \Gamma_{mn}^{(k-1)} \end{bmatrix} \tag{7.14}$$

where $2 \le k \le L$ and $\Theta_{mn}^{(k)} = \gamma_{mn}(d - d_k)$. The recursion starts with $\beta_{mn}^{(1)} = 1$ and $\Gamma_{mn}^{(1)} = 0$. The coefficients f_{mn} are then given by

$$f_{mn} = \frac{1}{\sigma_L \cdot \gamma_{mn}} \cdot \frac{\beta_{mn}^{(L)}\tanh(\gamma_{mn}d) + \Gamma_{mn}^{(L)}}{\beta_{mn}^{(L)} + \Gamma_{mn}^{(L)}\tanh(\gamma_{mn}d)} \quad , \quad m > 0 \text{ or } n > 0. \tag{7.15}$$

For the case $m = n = 0$, the quantities $\beta_{00}^{(L)}$ and $\Gamma_{00}^{(L)}$ are computed from the recursion

$$\begin{bmatrix} \beta_{00}^{(k)} \\ \Gamma_{00}^{(k)} \end{bmatrix} = \begin{bmatrix} \frac{\sigma_{k-1}}{\sigma_k} & 0 \\ \left(\frac{\sigma_{k-1}}{\sigma_k} - 1\right)d_k & 1 \end{bmatrix} \cdot \begin{bmatrix} \beta_{00}^{(k-1)} \\ \Gamma_{00}^{(k-1)} \end{bmatrix} \tag{7.16}$$

starting with $\beta_{00}^{(1)} = 1$ and $\Gamma_{00}^{(1)} = d$. The coefficient f_{00} is given by

$$f_{00} = \frac{1}{\sigma_L} \cdot \frac{\Gamma_{00}^{(L)}}{\beta_{00}^{(L)}}. \tag{7.17}$$

For the case of the uniform substrate ($L = 1$) with conductivity σ, the coefficients f_{mn} are given by the simpler equation

$$f_{mn} = \begin{cases} [\tanh(\gamma_{mn}d)]/(\sigma \cdot \gamma_{mn}) & m > 0 \text{ or } n > 0, \\ (d/\sigma) & m = 0,\ n = 0 \end{cases} \tag{7.18}$$

For a given substrate profile, the cost associated with computing an $M \times M$ array $\{f_{mn}\}$, $0 \le m, n \le M - 1$, is $O(M^2)$.

Suppose that the substrate surface is represented by a regular, $M \times M$ computational grid, and that each panel aligns to a *cell* on this grid. This situation is depicted in Figure 7-5 in which panels are shown in gray. It was shown in [43] that by truncating the Green's function (7.11) to a finite M by M series and substituting this into (7.5), it is possible to construct each entry P_{ij} of the coefficient-of-potential matrix from linear combinations of appropriate terms from the two-dimensional $(M + 1) \times (M + 1)$ array $\{F_{ij}\}$ defined by

$$F_{ij} = \sum_{m=0}^{M-1}{}' \sum_{n=0}^{N-1}{}' f_{mn} \cos\left(i\pi\frac{m}{M}\right)\cos\left(j\pi\frac{n}{M}\right), \quad 0 \le i, j \le M \tag{7.19}$$

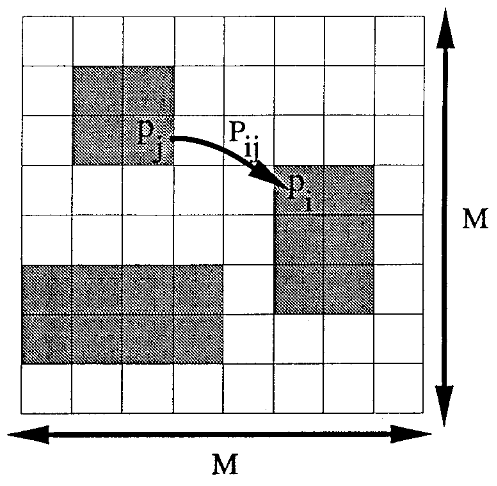


FIGURE 7-5: Each panel aligns to a cell.

where the primed summation indicates that the first term in each sum is to be multiplied by (1/2). The array $\{F_{ij}\}$ can thus be regarded as a *lookup table.* If M is a power of two, then $\{F_{ij}\}$ is the two-dimensional, Type-1 inverse Discrete Cosine Transform [13] (IDCT) of the array $\{f_{mn}\}$, which can be computed efficiently with the Fast Fourier Transform (FFT). This is an efficient way of evaluating and integrating the Green's function $G(\mathbf{r};\mathbf{r}')$ without summing the series in (7.11) directly. However, it is possible to show that for reasonable accuracy, at least $2M \times 2M$ terms in the series (7.11) are required for a *physical* grid of size $M \times M$. Hence we leave to Chapter 9 the details of this table lookup approach, along with our development of an efficient algorithm which achieves higher accuracy by incorporating more terms in the Green's function expansion.

Although the construction of the lookup table $\{F_{ij}\}$ via DCT allows the individual entries of P to be computed, the solution of (7.4) still requires $\mathcal{O}(N^3)$ CPU time to factor and $\mathcal{O}(N^2)$ memory to store the dense matrix P. This limits the size of the problem to a few hundred panels. However, the table lookup approach will be used in the multigrid algorithm when *direct* panel-to-panel interactions are required.

7.3 Eigenfunction based Framework

In contrast to the Green's function based approach derived in [37, 43] and outlined above, we derive here an alternative framework based on *eigenfunctions*, or *eigendecomposition.* This framework offers two crucial advantages over the Green's function based approach. The first advantage is that analytic properties of our particular first-kind integral equation 7.3 become immediately accessible, offering critical insights which motivate the development of a multiresolution analysis. The second advantage is that the eigendecomposition picture leads to a *sparsification* algorithm for the dense matrix-vector product required in solving the discretized integral equation (7.4). In this section, we describe the eigenfunction based approach and its connection with the Green's function based approach.

Let $\Omega \equiv [0,a] \times [0,b]$ represent the entire substrate surface, and let $\varphi : \Omega \longrightarrow \Re$ be a function satisfying

$$\int_\Omega G(x,y;x',y') \cdot \varphi(x',y') da' = \lambda \cdot \varphi(x,y) \quad , \qquad (x,y) \in \Omega. \tag{7.20}$$

Hence, $\varphi(x,y)$ is an *eigenfunction* of the integral equation *defined over the entire substrate surface* Ω. The scalar λ is the *eigenvalue* corresponding to $\varphi(x,y)$. Care must be taken to differentiate φ and λ from the eigenfunctions and eigenvalues of (7.3), which is defined over S, the collection of substrate *contacts.* To find the eigenfunctions $\varphi(x,y)$, it is convenient to look for solutions of the partial differential equation corresponding to (7.20)

$$-\nabla \cdot (\sigma \nabla \Psi(x,y,z)) = \varphi(x,y) \cdot \delta(z) \tag{7.21}$$

such that

$$\Psi(x,y,0) = \lambda \cdot \varphi(x,y). \tag{7.22}$$

Note that the Dirac delta funciton $\delta(z)$ was used to specify current sources only at the substrate surface.

It can be shown [50] that the normalized eigenfunctions are

$$\varphi_{mn}(x,y) = \alpha_{mn} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right), \qquad m,n \in \text{ integers} \tag{7.23}$$

where the normalization constants α_{mn} are

$$\alpha_{mn} = \begin{cases} \sqrt{1/(ab)} & m=0,\ n=0 \\ \sqrt{2/(ab)} & m=0,\ n>0 \text{ or } m>0,\ n=0 \\ \sqrt{4/(ab)} & m>0,\ n>0 \end{cases} \tag{7.24}$$

Notice the appearance of these eigenfunctions in the infinite series expansion (7.11) of the Green's function. Hence defined, the set $\{\varphi_{nm}\}$ is *orthonormal*

$$\int_\Omega \varphi_{ij}(x,y)\varphi_{mn}(x,y)\, dx\, dy = \begin{cases} 1 & (i,j)=(m,n) \\ 0 & \text{otherwise} \end{cases} . \tag{7.25}$$

The coefficients $\{f_{mn}\}$ used to compute the Green's function (7.11) in Section 7.2 have been slightly modified from the form given in [37] so that they are exactly equal to the *eigenvalues* $\{\lambda_{mn}\}$ in this section

$$\lambda_{mn} = f_{mn} \quad , \tag{7.26}$$

where f_{mn} is given in (7.15),(7.17), and (7.18). Also notice the similarity between α_{mn} defined in (7.24) and C_{mn} defined in (7.12). This intimate connection between the Green's function $G(x,y;x',y')$ and the eigenfunctions $\{\varphi_{nm}\}$ is explored below.

To facilitate our exposition, we first introduce some basic notation from the field of quantum mechanics. A function $\psi : \Omega \to \Re$, labeled as a *ket* $|\psi\rangle$, is now considered to be an element of a *function space* X, *i.e.* $|\psi\rangle \in X$. Its *adjoint* $|\psi\rangle^\dagger$ is labeled as a *bra* $\langle\psi|$. We define $|\psi\rangle^\dagger \equiv \langle\psi|$ and $\langle\psi|^\dagger \equiv |\psi\rangle$. Hence the ket is analogous to a *column* vector, while the bra is analogous to a *row* vector. The adjoint operation is similar to taking the transpose conjugate of a vector. The inner product $\langle\phi|\psi\rangle \in \Re$ between two functions $\phi, \psi \in X$ is defined as

$$\langle\phi|\psi\rangle = \int_\Omega \phi^*(x', y')\ \psi(x', y')\ dx'\ dy', \tag{7.27}$$

where $\phi^*(x, y)$ is the complex conjugate of $\phi(x, y)$. In addition, we have the property

$$(\ \langle\phi|\psi\rangle\ )^* = (\ \langle\phi|\psi\rangle\ )^\dagger = \langle\psi|\phi\rangle \tag{7.28}$$

A more powerful way of interpreting (7.27) is to see it as a *vector projection* of the element $|\psi\rangle$ onto the element $|\phi\rangle$. In particular, let us define the element $|x, y\rangle$, where $(x, y) \in \Omega$, to represent the two-dimensional Dirac delta function $\delta(x' - x, y' - y)$ centered at (x, y). Letting $|\phi\rangle = |x, y\rangle$ in (7.27) yields

$$\langle x, y|\psi\rangle = \psi(x, y). \tag{7.29}$$

Equation (7.29) gives the expansion of $|\psi\rangle$ in the basis set $|x, y\rangle$, with the coefficients of expansion given by $\psi(x, y)$.

A linear operator H mapping from a normed function space X to a normed function space Y, $H : X \to Y$, can also be defined. In operator notation, (7.20) can be written simply as

$$H|\varphi\rangle = \lambda|\varphi\rangle. \tag{7.30}$$

To simplify notation, let $|m, n\rangle$ represent the *eigenvector* corresponding to the *normalized* eigenfunction $\varphi_{mn}(x, y)$, that is,

$$\langle x, y|m, n\rangle = \varphi_{mn}(x, y)\ . \tag{7.31}$$

The orthonormality condition in (7.25) then becomes

$$\langle i, j|m, n\rangle = \begin{cases} 1 & (i, j) = (m, n) \\ 0 & \text{otherwise} \end{cases}\ . \tag{7.32}$$

Since the eigen-elements $\{|m, n\rangle\}$ are orthonormal and complete, the operator H can be expanded as an outer product in this *basis*

$$H = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} |m, n\rangle\ \lambda_{mn}\ \langle m, n|\ . \tag{7.33}$$

The Green's function $G(x, y; x', y')$ can be evaluated with (7.33) by computing the potential due to the source $|x', y'\rangle$ and then projecting onto the target $|x, y\rangle$

$$G(x, y; x', y') = \langle x, y|H|x', y'\rangle = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \langle x, y|m, n\rangle\ \lambda_{mn}\ \langle m, n|x', y'\rangle\ . \tag{7.34}$$

Use of (7.31) and (7.28) in (7.34) immediately yields the infinite series expansion (7.11). The first equality in (7.34) suggests that $G(x,y;x',y')$ can be considered a *matrix element* of the operator H in the basis $\{|x,y\rangle\}$. Since in general $G(x,y;x',y') \neq 0$, this is called a *dense* representation of the operator H. However, consider expanding H in the basis $\{|m,n\rangle\}$

$$\begin{aligned} F(m,n;m',n') &= \langle m,n|H|m',n'\rangle \\ &= \sum_{i=0}^{\infty}\sum_{j=0}^{\infty}\langle m,n|i,j\rangle\ \lambda_{ij}\ \langle i,j|m',n'\rangle \\ &= \delta_{mm'}\delta_{nn'}\lambda_{mn} \quad . \end{aligned} \tag{7.35}$$

The operator H has a diagonal matrix representation in the basis formed by the set of eigenfunctions $\varphi_{mn}(x,y)$. This is the motivation for our sparsification algorithm based on eigendecomposition. The connection between the Green's function based framework and the eigenfunction based framework can thus be viewed as a *coordinate transformation*, or *change of basis.*

8

Motivation for Multigrid Analysis

Although it has been shown in the previous chapter that the matrix elements P_{ij} can be constructed efficiently, it is still too expensive to store or factor the *dense* $N \times N$ matrix P directly when N, the number of panels, becomes large. To reduce the computational complexity for solving the discretized integral equation

$$Pq = v \tag{8.1}$$

several approaches have been previously proposed, each with its own difficulties. Heuristic schemes were described in [43, 37] and [41], which attempt to sparsify Y, the matrix *inverse* of P, by zeroing direct admittances Y_{ij} whenever panels i and j are "well-separated". However, accuracy is compromised since these heuristic schemes cannot give reliable error bounds or a simple means of error control. In [17, 18], a Krylov-subspace based iterative method, GMRES [9], was used in combination with multipole sparsification to solve (8.1). The multipole-accelerated computation of $(P \cdot \hat{q})$ takes all panel-to-panel interactions into account, and is much faster than direct matrix-vector multiplication. However, because multipole approximations require translational invariance in the Green's function, this approach cannot handle edge effects of the substrate, and it was necessary in [17, 18] to approximate the Green's function with a polynomial in $(1/r)$. A more fundamental difficulty with the approach in [17, 18] is that Krylov-subspace based iterative methods converge slowly for *ill-conditioned* linear systems [14, 15]. Such is the case for discretized first-kind integral equations with weakly singular kernels [28, 5]. Hundreds or more iterations may be required for convergence in a large problem, wiping out the benefits of an iterative approach.

Multigrid, or more generally, *multilevel*, methods offer enticing prospects as an iterative solver for the integral equation (8.1). The efficiency of multigrid iterative methods for solving elliptic partial differential equations (PDEs) discretized using finite-differences (FD) or finite-elements (FEM) is well known [44, 45, 46], and is a direct result of the fact that the convergence rate is independent of discretization, and hence problem size. This is to be contrasted with

Krylov-subspace based (conjugate-gradient style) iterative methods, whose convergence rates deteriorate with increasing FD or FEM mesh refinement, or equivalently, worse matrix conditioning.

In this chapter, we use the analytic properties of the underlying integral operator, described in Chapter 7, to motivate subsequent development of a multigrid solver. Section 8.1 explores similarities between the first-kind integral equation and the standard second-order elliptic PDE. In both cases, either the operator itself or its inverse is *unbounded* in the L^2 sense. Hence both problems are often considered *ill-posed.* It will be shown in Section 8.2 that in the framework of Sobolev spaces and Sobolev norms, both problems are bounded and boundedly-invertible. In this setting, the nature of the integral operator as a *pseudo-differential operator* is apparent. Finally, in Section 8.3 we describe how iterative solvers in general suffer from ill-conditioning in the linear system, and then describe the source of ill-conditioning as a result of the coexistence of eigenmodes with distinct characteristic length scales. The promise of multigrid methods is then clear, since they attempt to remove the ill-conditioning by analyzing the problem at each length scale independently.

8.1 Connection with Elliptic PDEs

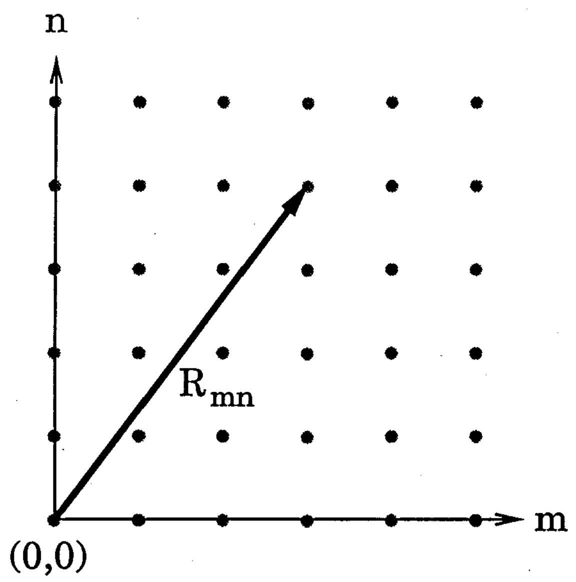


FIGURE 8-1: Discrete (m, n) space.

We begin by assuming that the integral equation is defined over the entire substrate surface $\Omega \equiv [0,a] \times [0,b]$

$$\int_\Omega G(x,y;x',y')\rho(x',y')da' = \phi(x,y), \qquad (x,y) \in \Omega \ , \tag{8.2}$$

or in the operator notation

$$\mathcal{K}\ \rho = \phi \ . \tag{8.3}$$

Recall from Chapter 7 the eigenfunctions

$$\varphi_{mn}(x,y) = \alpha_{mn} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right), \qquad m,n \in \text{ integers } . \tag{8.4}$$

If we further assume that the substrate has uniform conductivity σ, then the eigenvalues are

$$\lambda_{mn} = \begin{cases} [\tanh(\gamma_{mn} d)]/(\sigma \cdot \gamma_{mn}) & m > 0 \text{ or } n > 0, \\ (d/\sigma) & m = 0,\ n = 0 \end{cases} \tag{8.5}$$

Also recall the definition of γ_{mn}

$$\gamma_{mn} = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \ . \tag{8.6}$$

As m or n (or both) becomes large, the eigenvalue behaves as

$$\lambda_{mn} \longrightarrow \frac{1}{\sigma\sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2}} \qquad m,n \longrightarrow \infty. \tag{8.7}$$

Suppose that the domain Ω is a square $a = b$, and define

$$R_{m,n} \equiv \sqrt{m^2 + n^2} \tag{8.8}$$

to be the *radius* in the discrete (m, n) space shown in Figure 8-1, then

$$\lambda_{mn} \sim 1/R_{m,n} \qquad R_{m,n} \longrightarrow \infty. \tag{8.9}$$

If we further let $d = \sigma = (\pi/a) = 1$, then $\gamma_{mn} = R_{m,n}$, and the eigenvalue $\lambda_{mn} = \tanh(R_{mn})/R_{mn}$. The behavior of $\tanh(x)/x$ is plotted in Figure 8-2.

Consider an *elliptic PDE* in two dimensions

$$\begin{aligned} \nabla^2\psi(x,y) &= \varrho(x,y) \ \text{ on } \ \Omega = [0,a] \times [0,b] \\ \frac{\partial\psi}{\partial n} &= 0 \qquad \text{on } \partial\Omega \end{aligned} \tag{8.10}$$

where n is the unit outward normal at the boundary $\partial\Omega$. In operator notation, (8.10) becomes

$$\mathcal{L}\psi = \varrho \tag{8.11}$$

where the boundary conditions are implied. The eigenfunctions of the Laplacian operator

$$\nabla^2\varphi(x,y) = \lambda\varphi(x,y) \ \text{ on } \ \Omega \tag{8.12}$$

with homogeneous Neumann boundary conditions ($\partial\varphi/\partial n = 0$ on $\partial\Omega$) can be easily shown to be the same as those of the integral equation (8.4). The *eigenvalues*, however, are given by

$$\lambda_{mn} = \left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 \ . \tag{8.13}$$

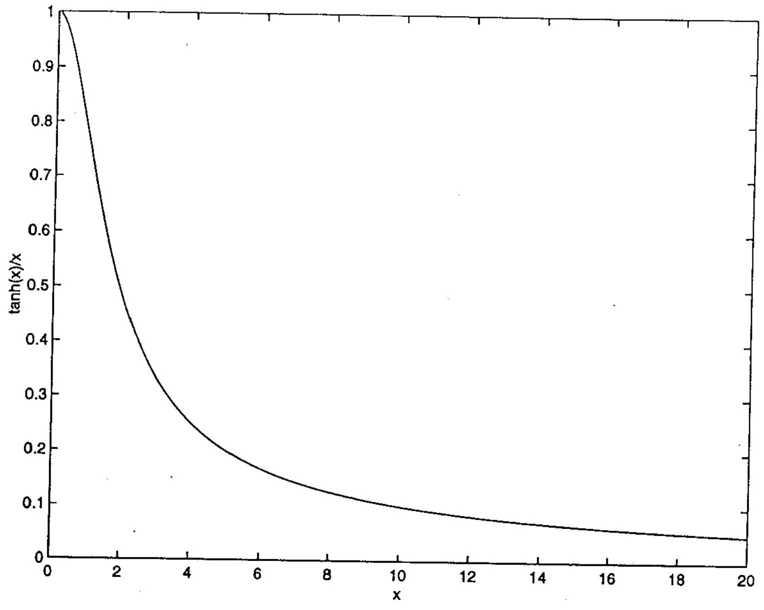


FIGURE 8-2: Behavior of $\tanh(x)/x$.

Again, letting $a = b$ and using the definition (8.8), we have for the elliptic PDE

$$\lambda_{mn} \sim R_{m,n}^2 \qquad R_{m,n} \to \infty. \tag{8.14}$$

Since $\lambda_{00} = 0$, a discretized version of (8.10) is *singular*. Let us *de-singularize* (8.10) by removing the $(m, n) = (0, 0)$ mode from the problem. This requires that

$$\int_\Omega \varphi_{0,0}(x,y)\ \varrho(x,y)\ dx\ dy = \sqrt{1/(ab)} \cdot \int_\Omega \varrho(x,y)\ dx\ dy = 0. \tag{8.15}$$

The lowest eigenvalue is now $\lambda_{01} = \lambda_{10} \neq 0$.

As $R_{mn} = \sqrt{m^2 + n^2}$ increases, the eigenfunction φ_{mn} becomes more oscillatory, or *non-smooth.* For the second-order PDE in (8.10), the eigenvalue λ_{mn} grows like $R_{m,n}^2$, whereas for the integral equation (8.2), the λ_{mn} *shrinks* like $1/R_{m,n}$. For this reason, the linear operator $\mathcal{K}$ associated with the first-kind integral equation at hand is called a *pseudo-differential operator of order* -1. This implies that the substrate Green's function $G(x, y; x', y')$ in (8.2) corresponds to a *weakly singular* kernel. The singularity in the Green's function is not strong enough to overcome the *smoothing* property of the two-dimensional integration over Ω. Since the weakly singular kernel $1/\|\mathbf{r} - \mathbf{r}'\|$ used for the ordinary capacitance extraction problem in three dimensions (3-D)

$$\phi(\mathbf{r}) = \int_{\partial\Omega} \frac{\sigma(\mathbf{r}')}{4\pi\epsilon\|\mathbf{r} - \mathbf{r}'\|} da' \tag{8.16}$$

also leads to a pseudo-differential operator of order -1 [28] over smooth boundaries $\partial\Omega$ of a 3-D volume Ω, there is much in common between the (8.2) and (8.16), both integral equations of the first-kind. Although it is possible to show, using the method of images [25], that the substrate

Green's function $G(\mathbf{r};\mathbf{r}') \sim 1/\|\mathbf{r}-\mathbf{r}'\|$ as $\mathbf{r} \to \mathbf{r}'$, it is not as useful as the eigenfunctions $\{\phi_{mn}\}$ in establishing analytic properties of (8.2).

If the functions $\phi, \rho, \psi, \varrho$ in (8.3) and (8.11) are *square integrable*, they can be considered elements of the function space $L^2(\Omega)$ with implied boundary conditions. The usual L^2 inner product and L^2 norm (also called the *mean square* norm) are defined as

$$\langle \varphi|\psi\rangle_{L^2} \equiv \int_\Omega \varphi^*(x,y)\ \psi(x,y)\ dx\ dy \quad , \quad \varphi, \psi \in L^2(\Omega) \tag{8.17}$$

and

$$\|\varphi\|_{L^2} \equiv \sqrt{\langle\varphi|\varphi\rangle_{L^2}}\ . \tag{8.18}$$

Let us consider the integral operator $\mathcal{K}$ and the differential operator $\mathcal{L}$ as mappings from $L^2(\Omega)$ into itself, *i.e.* $\mathcal{K} : L^2(\Omega) \to L^2(\Omega)$ and $\mathcal{L} : L^2(\Omega) \to L^2(\Omega)$. A linear operator $\mathcal{A} : L^2(\Omega) \to L^2(\Omega)$ is *bounded* if the operator norm defined by

$$\|\mathcal{A}\|_{L^2(\Omega)} \equiv \sup_{\varphi\in L^2(\Omega)} \frac{\|\mathcal{A}\varphi\|_{L^2(\Omega)}}{\|\varphi\|_{L^2(\Omega)}} \tag{8.19}$$

is finite. For the first-kind integral equation (8.2) in our substrate coupling problem, it can be seen from the eigenvalues (8.5) that the integral operator $\mathcal{K}$ is bounded, but since $\lambda_{mn} \to 0$ as $m,n \to \infty$, $\mathcal{K}$ does not possess a bounded *inverse*. On the other hand, for the elliptic PDE in (8.10), the operator $\mathcal{L}$ itself is *unbounded* since $\lambda_{mn} \to \infty$ as $m,n \to \infty$, as seen in (8.13). Recall that since the zero eigenvalue λ_{00} has been removed in the de-singularized version, $\mathcal{L}$ has a bounded inverse. The phenomenon of *ill-conditioning* in the numerical solution of discretized first-kind integral equations or elliptic PDEs arises fundamentally from the unboundedness of either the underlying continuous operator *or* its inverse. We shall see in the next section how it is possible to interpret both $\mathcal{K}$ and $\mathcal{L}$ as *bounded*, and *boundedly-invertible*, linear operators in the framework of Sobolev spaces and Sobolev norms.

8.2 Sobolev Spaces

Perhaps more accessible than the theory of pseudo-differential operators is the concept of *Sobolev spaces* [5]. We first introduce here the definition of Sobolev spaces for periodic functions over $[0,2\pi]$, following [5]. Then we extend this treatment to functions defined on the rectangle $\Omega = [0,a] \times [0,b]$ and show how the integral operators $\mathcal{K}$ and $\mathcal{L}$ defined in Section 8.1 maps between these spaces of functions.

Let $\phi : [0,2\pi] \to \Re$ be a 2π-periodic and *square integrable* function, *i.e.* $\phi \in L^2[0,2\pi]$. Then

$$\sum_{m=-\infty}^{\infty} a_m\ e^{imt} \tag{8.20}$$

where

$$a_m \equiv \frac{1}{2\pi}\int_0^{2\pi} e^{-imt}\ \phi(t)\ dt \tag{8.21}$$

is called the *Fourier series* of $\phi(t)$, and a_m are called the *Fourier coefficients.* Let $f_m(t) \equiv e^{imt}$ represent the Fourier basis functions. The set $\{e^{imt}\}$ is orthonormal and complete, and the series (8.20) converges in the mean square norm [5].

Now we define subspace $H^p[0, 2\pi]$ of $L^2[0, 2\pi]$ by requiring that the Fourier coefficients a_m decays at a certain rate as $|m| \to \infty$. Let $0 \leq p < \infty$. Then by $H^p[0, 2\pi]$ we denote the space of all functions $\phi \in L^2[0, 2\pi]$ with the property

$$\sum_{m=-\infty}^{\infty} (1 + m^2)^p |a_m|^2 < \infty \tag{8.22}$$

for the Fourier coefficients a_m of ϕ. The space $H^p[0, 2\pi]$ is called a Sobolev space. Notice that $H^0[0, 2\pi]$ coincides with $L^2[0, 2\pi]$. The higher the p value, the smoother are the functions contained in the space $H^p[0, 2\pi]$. Notice that non-integer values of p are allowed in the definition.

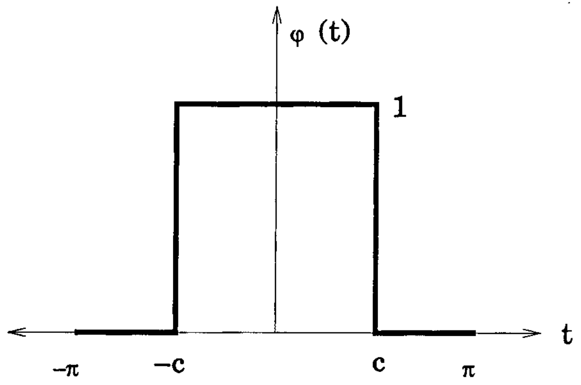


FIGURE 8-3: Step function in one dimension.

As a simple but important example, let us define the step function $\phi(t)$ shown in Figure 8-3

$$\phi(t) = \begin{cases} 1 & -c < t < c \\ 0 & \text{otherwise} \end{cases} \qquad t \in [-\pi, \pi]. \tag{8.23}$$

This function can be extended to a periodic function on $[0, 2\pi]$. The Fourier coefficients are given by

$$a_m = \frac{\sin(m\pi c)}{m\pi}. \tag{8.24}$$

Since $|\sin(m\pi c)| \leq 1$, let us plug in $|a_m| = (1/m)$ into (8.22) for $p = 1/2 - \varepsilon$, resulting in the series

$$\sum_{m=-\infty}^{\infty} (1 + m^2)^{\frac{1}{2}-\varepsilon} \left(\frac{1}{m}\right)^2 \approx \sum_{m=-\infty}^{\infty} \frac{1}{m^{1+2\varepsilon}}. \tag{8.25}$$

Clearly, (8.25) diverges for $\varepsilon = 0$ but converges for any $\varepsilon > 0$. Hence, we say that the step function $\phi(t)$ is *almost* smooth enough to be in $H^{1/2}[0, 2\pi]$, and that $\phi \in H^{1/2-\varepsilon}[0, 2\pi]$ for any $\varepsilon > 0$.

Assume that the series (8.22) converges for functions φ and ψ for some positive p. Then it is possible to define a more *stringent* Sobolev norm based on the Sobolev inner product

$$\langle \varphi | \psi \rangle_p \equiv \sum_{m=-\infty}^{\infty} (1+m^2)^p \; a_m^* \; b_m \tag{8.26}$$

where $\varphi, \psi \in H^p[0, 2\pi]$ with Fourier coefficients a_m and b_m, respectively. The Sobolev norm $\| \cdot \|_p$ is then

$$\|\varphi\|_p = \left\{ \sum_{m=-\infty}^{\infty} (1+m^2)^p \; |a_m|^2 \right\}^{1/2} \quad . \tag{8.27}$$

It is easy to see from (8.27) that $\|\varphi\|_p < \|\varphi\|_q$ if $p < q$.

For negative orders, the Sobolev space $H^{-p}[0, 2\pi]$ where $0 \leq p < \infty$ is defined as the *dual space* of $H^p[0, 2\pi]$, that is, the space of *bounded linear functionals* on $H^p[0, 2\pi]$. A linear functional $G : H^p[0, 2\pi] \rightarrow \Re$ maps every function $\varphi \in H^p[0, 2\pi]$ to a real number $G(\varphi)$. A linear functional G is *bounded* if its *norm*, defined as

$$\|G\|_p \equiv \sup_{\varphi \in H^p[0,2\pi]} \frac{|G(\varphi)|}{\|\varphi\|_p} \quad , \tag{8.28}$$

is finite. If we define the linear functional G by its action on each function $f_m(t) \equiv e^{imt}$

$$G(f_m) = c_m \quad , \tag{8.29}$$

then it can be shown [5] that

$$\|G\|_p = \left\{ \sum_{m=-\infty}^{\infty} (1+m^2)^{-p} \cdot |c_m|^2 \right\}^{1/2} \quad . \tag{8.30}$$

Each function $g \in L^2[0, 2\pi]$ is associated with a linear functional G by the duality pairing

$$G(\varphi) \equiv \frac{1}{2\pi} \int_0^{2\pi} g^*(t) \; \varphi(t) \; dt, \quad \varphi \in H^p[0, 2\pi]. \tag{8.31}$$

Since this defines a linear functional $G \in H^{-p}[0, 2\pi]$, we can regard $L^2[0, 2\pi]$ as a *subspace* of $H^{-p}[0, 2\pi]$. In this sense, the Sobolev space $H^{-p}[0, 2\pi]$ for $p \geq 0$ is a space of generalized functions or distributions $g(t)$, which may not be square integrable, and whose Fourier coefficients c_m are given by the corresponding linear functional. The Sobolev norm of this generalized function g is equal to the norm of its dual (the associated linear functional G), given by (8.30), *i.e.* $\|g\|_{-p} \equiv \|G\|_p$. For example, the linear functional defined by

$$G(\varphi) \equiv \frac{1}{2\pi} \varphi(\tau) \tag{8.32}$$

corresponds to the Dirac delta function, $\delta(t - \tau)$, which is an element of $H^{-1/2-\varepsilon}[0, 2\pi]$ for any positive ε. It is *almost* smooth enough to be in $H^{-1/2}[0, 2\pi]$.

We now show how Sobolev spaces $H^p(\Omega)$ can be defined for functions $\varphi : \Omega \to \Re$, where $\Omega = [0,a] \times [0,b]$. First, define $L^2(\Omega)$ as the space of functions $\varphi : \Omega \to \Re$ which are square integrable over Ω and which satisfy the boundary condition $\partial\varphi/\partial n = 0$ on $\partial\Omega$. Then, similar to (8.20) and (8.21), we define a *Fourier cosine series* (or just Fourier series) associated with φ

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} a_{mn} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right) \tag{8.33}$$

with Fourier coefficients a_{mn} are given be

$$a_{mn} \equiv C_{mn} \cdot \int_{\Omega} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right) \varphi(x,y)\, dx\, dy \quad , \tag{8.34}$$

where C_{mn} are the normalization constants given in (7.12). The normalized Fourier basis functions are

$$f_{mn} = \sqrt{C_{mn}} \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right) \quad . \tag{8.35}$$

Now, for $0 \le p < \infty$, define the Sobolev space $H^p(\Omega)$ as a subspace of $L^2(\Omega)$ by requiring that the Fourier coefficients a_{mn} decay at a certain rate. Then by $H^p(\Omega)$ we denote the space of all functions $\varphi \in L^2(\Omega)$ with the property

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^p\, |a_{mn}|^2 < \infty \tag{8.36}$$

for the Fourier coefficients a_{mn} of φ. The Sobolev inner product $\langle \cdot \mid \cdot \rangle_p$ is defined as

$$\langle\varphi|\psi\rangle_p \equiv \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^p\, a_{mn}^*\, b_{mn} \tag{8.37}$$

where $\varphi, \psi \in H^p(\Omega)$ with Fourier coefficients a_{mn} and b_{mn}, respectively. The Sobolev norm $\|\cdot\|_p$ is then

$$\|\varphi\|_p = \left\{\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^p\, |a_{mn}|^2\right\}^{1/2} \quad . \tag{8.38}$$

The Sobolev spaces $H^{-p}(\Omega)$, for $0 \le p < \infty$, is defined as the space of bounded linear functionals on $H^p(\Omega)$, as done previously in the 1-D case. If a linear functional $G \in H^{-p}(\Omega)$ is defined by

$$G(f_{mn}) = c_{mn} \quad , \tag{8.39}$$

then its norm is given by

$$\|G\|_p = \left\{\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^{-p} \cdot |c_{mn}|^2\right\}^{1/2} \quad . \tag{8.40}$$

A linear functional $G \in H^{-p}(\Omega)$ is associated with a generalized function $g(x,y)$ in a duality pairing similar to (8.31). The Fourier coefficients of $g \in H^{-p}(\Omega)$ is given by the c_{mn} in (8.39), and the Sobolev norm $\|g\|_{-p} \equiv \|G\|_p$ given by (8.40).

Now consider the differential operator $\mathcal{L}$ defined in Section 8.1, as a mapping between Sobolev spaces $\mathcal{L} : H^p(\Omega) \longrightarrow H^{p-2}(\Omega)$. The operator norm of $\mathcal{L}$ defined using the appropriate Sobolev norms, is

$$\|\mathcal{L}\|_{p\to p-2} \equiv \sup_{\varphi\in H^p(\Omega)} \frac{\|\mathcal{L}\varphi\|_{p-2}}{\|\varphi\|_p} \quad . \tag{8.41}$$

Let a_{mn} be the Fourier coefficients of φ. Since the eigenfunctions of $\mathcal{L}$ coincide with the Sobolev basis functions f_{mn}, the Fourier coefficients of $\mathcal{L}\varphi$ are given by $\lambda_{mn}a_{mn} = (m^2+n^2)a_{mn}$, as given by (8.13), where we have let $\pi/a = \pi/b = 1$. It then follows that

$$\begin{aligned}
\|\mathcal{L}\varphi\|_{p-2} &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^{p-2}|(m^2+n^2)a_{mn}|^2 \\
&\leq \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^{p}|a_{mn}|^2 \\
&= \|\varphi\|_p \ ,
\end{aligned} \tag{8.42}$$

which implies that $\|\mathcal{L}\|_{p\to p-2} \leq 1$. Hence $\mathcal{L} : H^p(\Omega) \longrightarrow H^{p-2}(\Omega)$ is now a *bounded* linear operator, whereas $\mathcal{L} : L^2(\Omega) \longrightarrow L^2(\Omega)$ is *unbounded* as shown in Section 8.1.

We now consider the integral operator $\mathcal{K}$ in (8.3) as a mapping $\mathcal{K} : H^p(\Omega) \longrightarrow H^{p+1}(\Omega)$, and show that under the appropriate Sobolev norms, $\mathcal{K}$ is both bounded and boundedly-invertible (*i.e.* has a bounded inverse). Similar to (8.41), the operator norm of $\mathcal{K}$ is

$$\|\mathcal{K}\|_{p\to p+1} \equiv \sup_{\varphi\in H^p(\Omega)} \frac{\|\mathcal{K}\varphi\|_{p+1}}{\|\varphi\|_p} \quad . \tag{8.43}$$

Again, let a_{mn} be the Fourier coefficients of φ. Since the eigenfunctions of $\mathcal{K}$ coincide with the Sobolev basis functions f_{mn}, the Fourier coefficients of $\mathcal{K}\varphi$ are given by $\lambda_{mn}a_{mn}$, with λ_{mn} given by (8.5). Again, letting $\pi/a = \pi/b = 1$ and $\sigma = d = 1$ to simplify things, we have

$$\lambda_{mn} = \begin{cases} 1 & m=0,\ n=0 \ , \\ \tanh(\gamma_{mn})/\gamma_{mn} & m>0 \text{ or } n>0 \end{cases} \ , \tag{8.44}$$

where $\gamma_{mn} = \sqrt{m^2+n^2}$. Making use of the fact that $\lambda_{00} = 1$ and that $\tanh(x) \leq 1$ for real $x > 0$, we have

$$\begin{aligned}
\|\mathcal{K}\varphi\|_{p+1} &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^{p+1}|\lambda_{mn}a_{mn}|^2 \\
&\leq |a_{00}|^2 + \sum_{m,n\neq 0,0}\sum(1+m^2+n^2)^{p+1}\left(\frac{1}{m^2+n^2}\right)|a_{mn}|^2 \\
&\leq |a_{00}|^2 + 2\sum_{m,n\neq 0,0}\sum(1+m^2+n^2)^{p}|a_{mn}|^2 \\
&\leq 2\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^{p}|a_{mn}|^2 \\
&= 2\|\varphi\|_p \ ,
\end{aligned} \tag{8.45}$$

resulting in $\|\mathcal{K}\|_{p\to p+1} \leq 2$.

For the inverse operator $\mathcal{K}^{-1} : H^{p+1}(\Omega) \to H^p(\Omega)$, we have the norm

$$\|\mathcal{K}^{-1}\|_{p+1\to p} \equiv \sup_{\varphi \in H^{p+1}(\Omega)} \frac{\|\mathcal{K}^{-1}\varphi\|_p}{\|\varphi\|_{p+1}} \ . \tag{8.46}$$

The Fourier coefficients of $\mathcal{K}^{-1}\varphi$ are now given by $\lambda_{mn}^{-1} \cdot a_{mn}$, given by

$$\lambda_{mn}^{-1} = \begin{cases} 1 & m = 0,\ n = 0 \ , \\ \gamma_{mn}/\tanh(\gamma_{mn}) & m > 0 \text{ or } n > 0 \end{cases} , \tag{8.47}$$

and it is easy to show that

$$\lambda_{mn}^{-1} < (1.32)\sqrt{m^2+n^2} \ \ , \ \ m > 0 \text{ or } n > 0 \tag{8.48}$$

Using (8.48) and $\lambda_{00}^{-1} = 1$, we get

$$\begin{aligned} \|\mathcal{K}^{-1}\varphi\|_p &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^p|\lambda_{mn}^{-1}\cdot a_{mn}|^2 \\ &\leq |a_{00}|^2 + \sum_{m,n\,\neq 0,0}\sum(1+m^2+n^2)^p \cdot |(1.32)\sqrt{m^2+n^2}\cdot a_{mn}|^2 \\ &\leq |a_{00}|^2 + (1.75)\sum_{m,n\,\neq 0,0}\sum(1+m^2+n^2)^{p+1}|a_{mn}|^2 \\ &\leq (1.75)\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}(1+m^2+n^2)^{p+1}|a_{mn}|^2 \\ &= (1.75)\|\varphi\|_{p+1} \ , \end{aligned} \tag{8.49}$$

from which we have $\|\mathcal{K}^{-1}\|_{p+1\to p} \leq 1.75$. Hence we have just proved the following theorem:

Theorem 8.1. The linear operator $\mathcal{K}$ associated with the first-kind integral equation (8.2) with the substrate kernel $G(x,y;x',y')$ given by (7.11) is a bounded *and* boundedly-invertible *mapping from the Sobolev space $H^p(\Omega)$ to the Sobolev space $H^{p+1}(\Omega)$.*

Since $\mathcal{K} : H^p(\Omega) \to H^{p+1}(\Omega)$, is bounded and boundedly-invertibe, and since $H^{p+1}(\Omega)$ is a *smooth* subset of functions in $H^p(\Omega)$, we call $\mathcal{K}$ a *smoothing* operator. The relation between the Sobolev spaces $H^p(\Omega)$ and $H^{p+1}(\Omega)$ is roughly depicted in Figure 8-4. This is not to be taken literally, since $H^{p+1}(\Omega)$ actually forms a *dense* set in $H^p(\Omega)$.

The idea that Sobolev norms can lead to linear operators which are bounded and boundedly-invertible can be used to formulate numerical methods [51]. The desired result will be linear systems with bounded condition numbers. However, the numerical evaluation of the appropriate *Sobolev norms* of fractional order is notoriously difficult, especially over complicated regions, as in our substrate coupling problem. The basic idea behind Sobolev norms is a *rescaling* of the Fourier components such that the high-frequency components are either emphasized or de-emphasized relative to the low-frequency components. This is called *renormalization* in the physics literature. Hence, rather than computing Sobolev norms directly, it is possible to

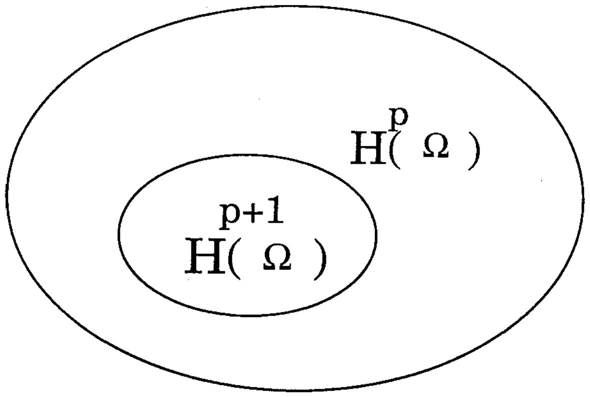


FIGURE 8-4: Relation between Sobolev spaces.

achieve the same results by analyzing the problem at various length scales, or resolutions. This effectively breaks up the various Fourier components of the problem, analyzing each component *individually.* This idea is explored in the next section.

8.3 Analysis at Multiple Length Scales

We first turn to the solution of linear systems using "black-box" iterative algorithms, including classical methods and Krylov-subspace based methods, and show how they both suffer from slow convergence for *ill-conditioned* linear systems, such as those generated by elliptic PDEs or first-kind integral equations. We then suggest how this difficulty might be overcome by exploiting the underlying analytic properties of the integral or differential operator. This motivates our development of fast-converging iterative schemes based on multigrid, or multiresolution, analysis.

Consider solving an $N \times N$ linear system

$$Ax = b \tag{8.50}$$

for x given b with an *iterative* algorithm. For ease of analysis, let us assume that the matrix $A \in \Re^{N \times N}$ is symmetric positive definite (SPD), *i.e.*

$$x^t A x > 0 \text{ if } \|x\| > 0 \ . \tag{8.51}$$

This implies that A is non-singular, and that all eigenvalues of A are real and positive. Let us further assume that A has unit *norm*, *i.e.* $\|A\| = 1$. Hence, the maximum eigenvalue of A is $\lambda_{max} = 1$, and the minimum eigenvalue is $\lambda_{min} \equiv \delta > 0$. The eigenvalue distribution of A is shown in Figure 8-5.

Consider the *operator splitting*

$$A = I - M \ , \tag{8.52}$$

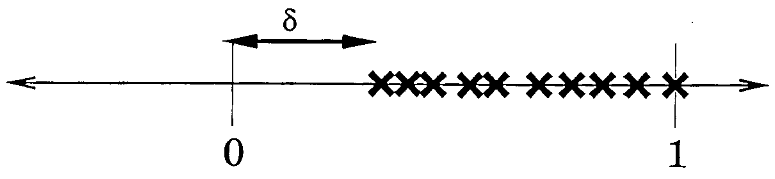


FIGURE 8-5: Eigenvalues of the matrix A.

where I is the $N \times N$ identity matrix. It is not difficult to show that $\|M\| = (1-\delta) < 1$. This implies that the *Neumann series* [52] for A^{-1} converges, that is

$$\begin{aligned} A^{-1} \equiv (I-M)^{-1} &= I + M + M^2 + M^3 + \cdots \\ &= \sum_{k=0}^{\infty} M^k \ . \end{aligned} \tag{8.53}$$

The solution x to the linear system (8.50) can be written

$$x = A^{-1}b = b + Mb + M^2b + M^3b + \cdots \ . \tag{8.54}$$

This motivates an *iterative* scheme for computing the solution x based on the operator splitting (8.52), described in Algorithm 2.

Algorithm 2 (Iterative Algorithm for Solving $(I-M)x = b$ *).*

```
Set $k \equiv 0$, initial guess $x^{(0)} \equiv b$.
Repeat {
    Compute $x^{(k+1)} = Mx^{(k)} + b$ .
    Set $k = k + 1$.
} Until converged.
```

This algorithm is basically a *preconditioned* version of the classical *Gauss-Jacobi* iteration [12]. The error at the k-th iteration

$$e^{(k)} \equiv x^{(k)} - x \tag{8.55}$$

can be easily shown to be reduced by the relation

$$\begin{aligned} \frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} &\leq \|M\| \ , \\ \frac{\|e^{(k)}\|}{\|e^{(0)}\|} &\leq \|M\|^k \ . \end{aligned} \tag{8.56}$$

Since $\|M\| = (1-\delta) < 1$, Algorithm 2 *converges* given the assumptions we have made about A. The *condition number* $\kappa(A)$ of a square matrix A is defined as

$$\kappa(A) \equiv \|A\| \cdot \|A^{-1}\| \tag{8.57}$$

where $\|\cdot\|$ denotes the 2-norm. This is equivalent to the definition

$$\kappa(A) \equiv \frac{\sigma_{max}}{\sigma_{min}} \tag{8.58}$$

where σ_{max} and σ_{min} are the largest and smallest *singular values* [15], respectively, of A. For a symmetric positive definite matrix A, we also have $\kappa(A) = \lambda_{max}(A)/\lambda_{min}(A)$. From Figure 8-5, it is clear that $\kappa = (1/\delta)$. Hence the error norm is reduced at the rate

$$\frac{\|e^{(k+1)}\|}{\|e^{(k)}\|} \leq \rho_e \equiv 1 - \frac{1}{\kappa} \quad . \tag{8.59}$$

Algorithm 2 essentially expresses the k-th iterate for the solution as a k-th order *polynomial* $p_k(M)$ of the iteration matrix M

$$x^{(k)} = b + Mb + M^2b + \cdots + M^kb \equiv p_k(M) \cdot b \quad . \tag{8.60}$$

Since $A = I - M$, $p_k(M)$ is also a k-th order polynomial in A

$$x^{(k)} = \hat{p}_k(A) \cdot b \quad . \tag{8.61}$$

In classical iterative algorithms such as the Jacobi iteration, the coefficients of the approximating polynomial $p_k(M)$, and hence those of $\hat{p}_k(A)$, are *independent* of A and the right-hand side b *Krylov-subspace* based iterative algorithms, on the other hand, taylors the matrix polynomial $\tilde{p}_k(A)$ specifically to the matrix A and each right-hand side b in an attempt to minimize some error metric at each iteration. Examples of Krylov-subspace methods are the Conjugate Gradient (CG), Generalized Minimum RESidual (GMRES), and Quasi Minimal Residual (QMR) algorithms [53]. Recall the definition of the residual $r^{(k)}$ for the k-th iterate $x^{(k)}$

$$r^{(k)} = b - Ax^{(k)} \quad . \tag{8.62}$$

Define the Krylov subspace $\mathcal{K}_k(A,b)$ as the k-dimensional linear subspace spanned by vectors generated from the right-hand side b

$$\mathcal{K}_k(A,b) = \text{span}\{b, Ab, A^2b, \ldots, A^{k-1}b\} \quad . \tag{8.63}$$

The popular GMRES algorithm, summarized in Algorithm 3, searches for a vector $x^{(k)}$ in $\mathcal{K}_k(A,b)$ which minimizes the *residual norm, i.e.*

$$r^{(k)} \equiv \|b - Ax^{(k)}\| = \min_{x^* \in \mathcal{K}_k(A,b)} \|b - Ax^*\| \quad , \tag{8.64}$$

where it has been assumed that the initial guess $x^{(0)} = 0$, and $r^{(0)} = b$. The condition $x^{(k)} \in \mathcal{K}_k(A, b)$ is equivalent to

$$x^{(k)} = \mathcal{Z}_{k-1}(A) \cdot b \quad , \tag{8.65}$$

where $\mathcal{Z}_{k-1}$ is a polynomial of order $k - 1$. It then follows that

$$\begin{aligned} r^{(k)} = b - Ax^{(k)} &= b - A \cdot \mathcal{Z}_{k-1}(A) \cdot b \\ &= p_k(A) \cdot b, \quad p_k(0) = 1 \quad , \end{aligned} \tag{8.66}$$

where p_k is a k-th order polynomial with the *constraint* $p_k(0) = 1$. Thus, (8.64) can be restated as

$$\|r^{(k)}\| = \min_{p_k \in \mathcal{P}_k, p_k(0)=1} \|p_k(A) \cdot r^{(0)}\| \quad , \tag{8.67}$$

where $\mathcal{P}_k$ denotes the space of k-th order polynomials. If A is SPD, then the upper bound derived using Chebyshev polynomials gives [14]

$$\begin{aligned} \frac{\|r^{(k+1)}\|}{\|r^{(k)}\|} \le \rho_r &\equiv \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \\ &= 1 - \frac{2}{\sqrt{\kappa}} + \mathcal{O}\left(\frac{1}{\kappa}\right) \quad , \end{aligned} \tag{8.68}$$

where the $\mathcal{O}(1/\kappa)$ term can be ignored for $\kappa >> 1$.

The number of iterations k required to reduce the error or residual norm by a factor of 10^{-m} is given by

$$\rho^k \le 10^{-m} \quad \text{, or} \quad k \ge \frac{m}{-\log_{10} \rho} \tag{8.69}$$

where $\rho = \rho_e$ or ρ_r. The *rate of convergence* is defined as

$$R = -\log_{10}(\rho) \quad . \tag{8.70}$$

From (8.59), we see that the classical Jacobi iteration gives

$$R_e \sim 1/\kappa \quad , \tag{8.71}$$

whereas from (8.68), the GMRES iterative method yields

$$R_r \sim 1/\sqrt{\kappa} \quad . \tag{8.72}$$

Hence, as the condition number κ grows, the number of iterations required of classical iterative methods also grows as κ, whereas that of Krylov-subspace methods grows only as the *square root* of κ. Nevertheless, the required number of GMRES iterations still grows *without bound* as matrix condition worsens.

```
Algorithm 3 ( GMRES iterative method for solving Ax=b ).

1. Start:
   Set $k = 0, x_0 = 0, r_0 = b$ and $\beta = \|b\|, v_1 = b/\beta$
   Define the $(m+1) \times m$ matrix $H_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$ and initialize it to zero.
2. Arnoldi loop:
   (a) For $j = 1, 2, \ldots, m$ do {
         Compute $w_j = Av_j$
         For $i = 1, \ldots, j$ do {
            $h_{ij} = (v_i, w_j)$
            $w_j = w_j - h_{ij} v_i$
         }
   }
   (b) Compute $h_{j+1,j} = \|w_j\|$. If $h_{j+1,j} = 0$ go to 3.
   (c) Compute $v_{j+1} = w_j / h_{j+1,j}$.
3. Form approximate solution:
   Compute $y_m$, the minimizer (over $y$) of $\|\beta e_1 - H_m y\|$ and $x_m = V_m y_m$ where
   $V_m = [v_1, v_2, \ldots, v_m]$.
```

Recall from section 8.1 the eigenvalues and eigenfunctions of the integral operator $\mathcal{K}$ on Ω

$$\varphi_{mn}(x,y) \sim \cos\left(\frac{m\pi x}{a}\right) \cos\left(\frac{n\pi y}{b}\right), \qquad m, n \in \text{integers} \ . \tag{8.73}$$

and

$$\lambda_{mn} \sim \frac{1}{\sqrt{m^2+n^2}}, \qquad m, n \to \infty. \tag{8.74}$$

Hence the operator $\mathcal{K}$ has a infinite number of real and positive eigenvalues, with an accumulation point at zero as $m, n \to \infty$, as depicted in Figure 8-6. Since the $N \times N$ matrix P in

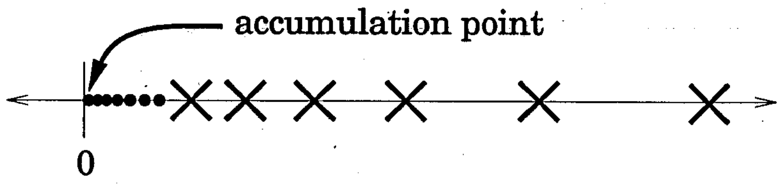


FIGURE 8-6: Eigenvalues accumulate at zero.

(7.4) is a discrete representation of $\mathcal{K}$, it is capable of approximating only N of the largest eigenvalues which correspond to the "lowest-frequency" eigenfunctions of $\mathcal{K}$ (*i.e.* the cosine modes with lowest m, n). As the mesh is refined, or as N increases, it becomes possible to represent the "higher-frequency" eigenfunctions and their associated eigenvalues, which shrink at the rate given in (8.74). This is the mechanism through which P becomes more *ill-conditioned* with mesh refinement. For a given discretization, the large eigenvalues correspond to the low-frequency cosine modes, and the small eigenvalues to the high-frequency cosine modes. This is shown in Figure 8-7. It is now clear that the ill-conditioning in the linear system is caused

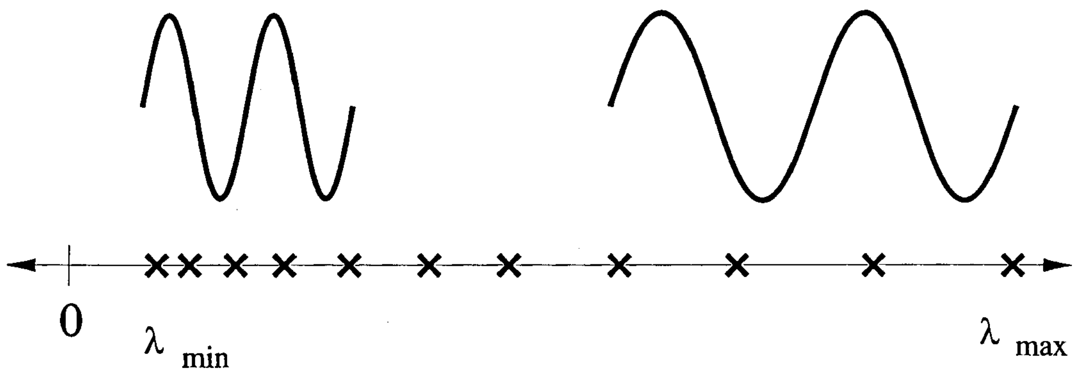


FIGURE 8-7: Mixing of low and high frequency modes.

by the simultaneous presence of eigenmodes with very distinct *characteristic length scales*, or spatial variations. We shall term this phenomenon "mode-mixing". Since iterative methods generally suffer from ill-conditioning, this analysis suggests a way of resolving this difficulty: If the problem can be analyzed *separately* at several length scales, then each of the *sub-problems* may be "better-conditioned" than the original problem in the sense that the ratio of eigenvalues associated with eigenmodes *within* each length scale is now much closer to one. The critical insight is that for first-kind integral equations, the characteristic length scale of an eigenmode is monotonically related to the *magnitude* of its associated eigenvalue. The same argument can also be made for equations defined over complicated geometries with only minor modifications. Once the linear system is "transformed" into a well-conditioned one, both classical linear relaxation or Krylov-subspace methods are expected to converge rapidly. The former approach is the stand-alone multigrid algorithm, which is a linear relaxation scheme. The latter approach leads to a multigrid-preconditioned Krylov-subspace algorithm. We later demonstrate that even with the optimal search strategy, the preconditioned Krylov-subspace approach produces almost no improvement in convergence rate over the stand-alone multigrid method. This is the justification for our claim that the multigrid scheme should be considered the principal "solver" rather than a "preconditioner".

9

Sparsification via Eigendecomposition

We show in this chapter how to compute the matrix-vector product $P \cdot q$ efficiently on a regular $M_x \times M_y$ substrate grid, where each of the N panel aligns to a *cell* on this grid, as shown previously in Figure 7-5. This is achieved by exanding the *global* current density function $J(x, y)$ as a sum of *eigenfunctions* of the integral operator $\mathcal{K}$ on Ω, given in (8.4). The DCT is then used to compute the average potentials on the $M_x \times M_y$ substrate grid, from which the individual panel potentials can be easily extracted. Our eigendecomposition approach differs from the multipole-accelerated approach used in [17] in one critical respect: For the multipole approximation used in [17], it was necessary to assume a substrate Green's function which has translational invariance and which can be fitted to a sum of polynomials in $(1/r)$, where $r = \|r - r'\|$. In contrast, the eigendecomposition approach accounts for the substrate edge effects explicitly and does not require translational symmetry in $G(r; r')$.

Although the eigenfunctions in (7.23) also appear in [43, 37], they were used indirectly to construct the panel-to-panel interaction coefficients P_{ij}. Let $M = M_x = M_y$, and and let N be the total number of panels. Assume that all panels are minimum sized cells on the $M \times M$ substrate grid, and that 10% of the cells are occupied by actual panels (*i.e.* $N = (0.1)M^2$). Then the dense matrix-vector multiplication $P \cdot q$ in [43] would require $\mathcal{O}(N^2) \sim \mathcal{O}(0.01 \times M^4)$ operations. In contrast, we use the eigenfunctions directly to expand the global current density $J(x, y)$ and show that the $P \cdot q$ product can be computed in $\mathcal{O}(2 \cdot M^2 \cdot \log_2(M))$ operations using the DCT. At $M = 128$, eigendecomposition is already an order of magnitude faster than direct multiplication. For the case of non-uniform panel sizing, the eigendecomposition approach may be adapted to achieve similar computational savings, but this is not pursued in this thesis.

We define *normalized* prototype characteristic functions centered about the origin in one dimension

$$\Theta_a(x) = \begin{cases} M/a & \text{if } |x| \le a/2M \\ 0 & \text{otherwise} \end{cases}, \qquad \Theta_b(y) = \begin{cases} M/b & \text{if } |y| \le b/2M \\ 0 & \text{otherwise} \end{cases}. \tag{9.1}$$

This is shown in Figure 9-1. We shall call $\Theta_a(x)$ and $\Theta_b(y)$ square-bump functions. It is then clear that the panel characteristic functions $\mathcal{X}_i(x, y)$ can be obtained by combining and shifting $\Theta_a(x)$ and $\Theta_b(y)$. The piecewise constant Galerkin discretization implies that the global current density is of the form

$$J(x,y) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} f(i,j) \cdot \Theta_a\left(x - \frac{(i+1/2)\,a}{M}\right) \cdot \Theta_b\left(y - \frac{(j+1/2)\,b}{M}\right). \tag{9.2}$$

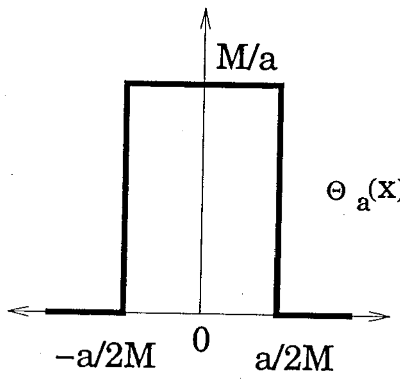


FIGURE 9-1: Prototype characteristic function.

If we can expand $J(x, y)$ in (9.2) in terms of the eigenfunctions $\{\varphi_{ij}\}$

$$J(x,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a(i,j)\, \varphi_{ij}(x,y), \tag{9.3}$$

then (7.20) immediately leads to

$$\Phi(x,y) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \lambda(i,j)\, a(i,j)\, \varphi_{ij}(x,y). \tag{9.4}$$

This motivates our development of a sparsification algorithm based on eigendecomposition. To make the computation feasible, the infinite series in (9.3) and (9.4) is truncated to $K \times K$ terms, where $K \ge 2M$ for reasonable accuracy. We will show in this chapter that if $\eta(p, q)$ is the $M \times M$ 2-D Type-2 DCT of $f(i, j)$, then the average cell potentials $\overline{\Phi}(p, q)$ are given by the $M \times M$ 2-D Type-2 inverse DCT of the array $\eta(p, q)\widehat{\lambda}(p, q)$, where $\widehat{\lambda}(p, q)$ is a *modified* eigenvalue to be defined later. The use of modified eigenvalues keeps the size of the DCT and inverse DCT to $M \times M$ regardless of the number of terms ($K \times K$) used to expand $J(x, y)$.

To compute panel potentials from panel currents using eigendecomposition, we will make use of the two-dimensional (2-D) Type-2 discrete cosine transform (DCT) and its inverse transform

(IDCT). The Type-2 DCT array is of size $N_x \times N_y$, where N_x, N_y are both powers of two. The indexes run from zero to $N_x - 1$ or $N_y - 1$. The DCT of an $N_x \times N_y$ array $\{f(j_x, j_y)\}$ is defined as

$$\eta(k_x, k_y) = \sum_{j_x=0}^{N_x-1} \sum_{j_y=0}^{N_y-1} f(j_x, j_y) \cdot \cos\left(\frac{\pi k_x (j_x + 1/2)}{N_x}\right) \cdot \cos\left(\frac{\pi k_y (j_y + 1/2)}{N_y}\right) \quad . \tag{9.5}$$

Its inverse transform (IDCT) is defined as

$$f(j_x, j_y) = \left(\frac{4}{N_x N_y}\right) {\sum_{k_x=0}^{N_x-1}}' {\sum_{k_y=0}^{N_y-1}}' \eta(k_x, k_y) \cdot \cos\left(\frac{\pi k_x (j_x + 1/2)}{N_x}\right) \cdot \cos\left(\frac{\pi k_y (j_y + 1/2)}{N_y}\right) \tag{9.6}$$

where the primed summation indicates that the first term in each sum is to be multiplied by (1/2). Fast algorithms for the implementation of the 2-D Type-2 DCT can be found in [54, 55, 56, 57, 58]. The transforms in (9.5) or (9.6) can be performed in $\mathcal{O}(N_x N_y \log_2(N_x N_y))$ operations.

In the multigrid algorithm, it will also be necessary to compute nearby panel-to-panel interaction coefficients P_{ij}. This is done using the table look-up approach previously developed by [37, 43]. Let $\{\lambda(k_x, k_y)\}$ represent an $N_x \times N_y$ array of eigenvalues, then we will need its Type-1 *inverse* discrete cosine transform (IDCT) $\{\overline{\lambda}(j_x, j_y)\}$ as the look-up table. This computation

$$\overline{\lambda}(j_x, j_y) = \left(\frac{4}{N_x N_y}\right) {\sum_{k_x=0}^{N_x-1}}' {\sum_{k_y=0}^{N_y-1}}' \lambda(k_x, k_y) \cdot \cos\left(\frac{\pi k_x j_x}{N_x}\right) \cdot \cos\left(\frac{\pi k_y j_y}{N_y}\right) \tag{9.7}$$

is performed only *once* during the initial set-up phase. For convenience, we state here the standard definition of the 1-D Type-1 DCT and IDCT, which is a transform on $(N + 1)$ data points, where N is a power of two. The 1-D DCT is defined as

$$F(k) = \frac{1}{2}[f(0) + (-1)^k f(N)] + \sum_{j=1}^{N-1} f(j) \cdot \cos\left(\frac{\pi j k}{N}\right) \quad , \quad k = 0 : N \tag{9.8}$$

and the 1-D IDCT as

$$f(j) = \left(\frac{2}{N}\right) \left\{\frac{1}{2}[F(0) + (-1)^j F(N)] + \sum_{k=1}^{N-1} F(k) \cdot \cos\left(\frac{\pi j k}{N}\right)\right\} \quad , \quad j = 0 : N \ . \tag{9.9}$$

Standard fast DCT algorithms [13] exist for evaluating (9.8) and (9.9). The 2-D inverse transform (9.7) can then be performed by successively applying (9.9) in each direction, where $F(N)$ is always first set to zero.

To derive the eigenvalue folding algorithm, we first tackle the one dimensional (1-D) problem. Suppose that the current density function $J(x)$ is expressed as a sum of 1-D characteristic functions

$$J(x) = \sum_{i=0}^{M-1} f(i) \cdot \Theta_a\left(x - \frac{(i + 1/2)\,a}{M}\right) \tag{9.10}$$

where $f(i)$ denotes the total current in the 1-D panel p_i. Let

$$\begin{aligned} \varphi_k(x) &= \alpha_k \cos\left(\frac{\pi k x}{a}\right) \\ \alpha_k &= \begin{cases} \sqrt{1/a} & k = 0 \\ \sqrt{2/a} & k > 0 \end{cases} \end{aligned} \tag{9.11}$$

be the normalized eigenfunctions in the interval $[0, a]$. The current density $J(x)$ in (9.10) can be expanded as an infinite series in the eigenfunctions $\varphi_k(x)$. Truncating this series to K terms leads to

$$\begin{aligned} J(x) &= \sum_{k=0}^{K-1} B(k)\varphi_k(x) \\ &= \sum_{k=0}^{K-1} B(k)\alpha_k \cos\left(\frac{\pi k x}{a}\right) \end{aligned} \tag{9.12}$$

where K must be reasonably large for an accurate approximation. It can be shown that the expansion of "step" functions with cosines is fairly accurate if we choose $K = 2M$ or $K = 4M$. The expansion coefficients $B(k)$ are computed as

$$\begin{aligned} B(k) &\equiv \int_0^a J(x) \cdot \varphi_k(x)\, dx \\ &= \alpha_k \sum_{i=0}^{M-1} f(i) \cdot \left[\int_0^a \Theta_a\left(x - \frac{(i+1/2)\,a}{M}\right) \cdot \cos\left(\frac{\pi k x}{a}\right)\, dx\right] \end{aligned} \tag{9.13}$$

where k is an integer and $k = 0 : K-1$. The terms in square brackets can be evaluated to yield

$$B(k) = \alpha_k \cdot U_M(k) \sum_{i=0}^{M-1} f(i) \cdot \cos\left(\frac{\pi k(i+1/2)}{M}\right) \quad , \qquad k = 0 : K-1 \tag{9.14}$$

where the function $U_M(k)$ is defined as

$$U_M(k) \equiv \begin{cases} 1 & k = 0 \\ \left(\frac{2M}{\pi k}\right) \cdot \sin\left(\frac{\pi k}{2M}\right) & k > 0 \end{cases} . \tag{9.15}$$

For convenience we define

$$\widetilde{B}(k) \equiv \sum_{i=0}^{M-1} f(i) \cdot \cos\left(\frac{\pi k(i+1/2)}{M}\right) \quad , \quad k = 0 : K-1 \tag{9.16}$$

which leads to

$$B(k) = \alpha_k U_M(k) \cdot \widetilde{B}(k) \ . \tag{9.17}$$

Assuming that the expansion coefficients $B(k)$ have been computed for $k = 0 : K-1$, we can express the potential $\Phi(x)$ as

$$\Phi(x) = \sum_{k=0}^{K-1} \lambda(k) B(k) \alpha_k \cos\left(\frac{\pi k x}{a}\right) \tag{9.18}$$

where $\lambda(k)$ is the 1-D eigenvalue for φ_k. Given (9.18), the *average* potential over panel p_i is computed as

$$\begin{aligned}
\Phi(i) &\equiv \int_0^a \Phi(x) \cdot \Theta_a \left(x - \frac{(i+1/2)\,a}{M} \right) \, dx \\
&= \sum_{k=0}^{K-1} \alpha_k \lambda(k) B(k) \cdot \left[\int_0^a \Theta_a \left(x - \frac{(i+1/2)\,a}{M} \right) \cdot \cos\left(\frac{\pi k x}{a} \right) \, dx \right] \\
&= \sum_{k=0}^{K-1} \alpha_k \lambda(k) B(k) \cdot \left[U_M(k) \cdot \cos\left(\frac{\pi k (i+1/2)}{M} \right) \right] \quad . \qquad (9.19)
\end{aligned}$$

Making use of (9.17) in (9.19), we get

$$\Phi(i) = \sum_{k=0}^{K-1} \alpha_k^2 \cdot \widetilde{B}(k) \cdot \lambda(k) \cdot U_M^2(k) \cdot \cos\left(\frac{\pi k (i+1/2)}{M} \right) \quad . \qquad (9.20)$$

Using the defintion for the normalization constants α_k in (9.11), we get

$$\Phi(i) = \left(\frac{2}{a} \right) \sum_{k=0}^{K-1}{}' \, \widetilde{B}(k) \cdot \lambda(k) \cdot U_M^2(k) \cdot \cos\left(\frac{\pi k (i+1/2)}{M} \right) \quad , \qquad (9.21)$$

where the prime in the summation indicates that the first term in the sum is to be multiplied by (1/2).

Although there are a total of K cosine terms used in the expansion, we show that using an *eigenvalue folding* technique, it is possible to reduce the computation to a 1-D Type-2 DCT of size M. Thus, an arbitrary number of cosine modes can be included in the series expansion (9.12) without increasing the cost of the DCT calculation. For the case $K = 2M$, the mode index k ranges from zero to $(2M-1)$. For the modes in the upper half (*i.e.* $k = M : (2M-1)$), we can write $k = M + q$, and use the following relation

$$\begin{aligned}
\cos\left(\frac{\pi k (i+1/2)}{M} \right) &= -\sin\left(\pi(i+1/2)\right) \cdot \sin\left(\frac{\pi q (i+1/2)}{M} \right) \\
&= (-1)^{i+1} \sin\left(\frac{\pi q (i+1/2)}{M} \right) \quad , \quad q = 0 : M-1. \qquad (9.22)
\end{aligned}$$

Using the identity $\sin(\theta) = \cos(\pi/2 - \theta)$, we derive

$$\sin\left(\frac{\pi q (i+1/2)}{M} \right) = (-1)^i \cdot \cos\left(\frac{\pi q' (i+1/2)}{M} \right) \qquad (9.23)$$

where the index q' is defined as

$$q' \equiv M - q \quad , \qquad q = 0 : M-1. \qquad (9.24)$$

Combining (9.22) and (9.23) yields the useful relation

$$\cos\left(\frac{\pi (M+q)(i+1/2)}{M} \right) = -\cos\left(\frac{\pi q' (i+1/2)}{M} \right) \quad . \qquad (9.25)$$

Now we break up the length-2M sequence $\widetilde{B}(k)$ into two sequences, each of legnth M. The lower sequence $\eta(q)$ corresponds to $\widetilde{B}(k)$ for $k = 0 : M-1$ and is defined as

$$\eta(q) \equiv \sum_{i=0}^{M-1} f(i) \cdot \cos\left(\frac{\pi q(i+1/2)}{M}\right) \quad , \quad q = 0 : M-1. \tag{9.26}$$

Notice that $\{\eta(q)\}$ is simply the 1-D Type-2 DCT of $\{f(i)\}$. The upper sequence $\zeta(q)$ corresponds to $\widetilde{B}(k)$ for $k = M : 2M-1$ and is defined as

$$\begin{aligned} \zeta(q) &\equiv \sum_{i=0}^{M-1} f(i) \cdot \cos\left(\frac{\pi(M+q)(i+1/2)}{M}\right) \\ &= -\sum_{i=0}^{M-1} f(i) \cdot \cos\left(\frac{\pi q'(i+1/2)}{M}\right) \quad , \quad q = 0 : M-1 \quad , \quad q' = M-q, \end{aligned} \tag{9.27}$$

where (9.25) has been used in the second equality. The sequence $\zeta(q)$ follows directly from $\eta(q)$ since

$$\zeta(q) = -\eta(q') = -\eta(M-q) \quad . \tag{9.28}$$

The first term $\zeta(0)$ is zero as seen directly from (9.27). Thus we have the relation

$$\zeta(q) = \begin{cases} 0 & q = 0 \\ -\eta(M-q) & q = 1 : M-1 \end{cases} \tag{9.29}$$

which essentially states that $\{\zeta(q)\}$ is a *mirror image* of $\{\eta(q)\}$.

With the help of (9.25) and the definitions (9.26), (9.27), we can rewrite (9.21) for the $K = 2M$ case as

$$\begin{aligned} \Phi(j) &= \left(\frac{2}{a}\right) \sum_{k=0}^{M-1}{}' \ \eta(k) \cdot [\lambda(k) \cdot U_M^2(k)] \cdot \cos\left(\frac{\pi k(j+1/2)}{M}\right) \\ &\quad - \left(\frac{2}{a}\right) \sum_{q=0}^{M-1} \zeta(q) \cdot [\lambda(M+q) \cdot U_M^2(M+q)] \cdot \cos\left(\frac{\pi q'(j+1/2)}{M}\right) \end{aligned} \tag{9.30}$$

Making use of (9.29) and (9.24), we rewrite the second term on the right-hand side of (9.30) as

$$\begin{aligned} &- \left(\frac{2}{a}\right) \sum_{q'=1}^{M-1} \zeta(M-q') \cdot [\lambda(2M-q') \cdot U_M^2(2M-q')] \cdot \cos\left(\frac{\pi q'(j+1/2)}{M}\right) \\ &= \left(\frac{2}{a}\right) \sum_{q'=1}^{M-1} \eta(q') \cdot [\lambda(2M-q') \cdot U_M^2(2M-q')] \cdot \cos\left(\frac{\pi q'(j+1/2)}{M}\right) \quad . \end{aligned} \tag{9.31}$$

This allows (9.30) to be simplified as

$$\Phi(j) = \left(\frac{2}{a}\right) \sum_{q=0}^{M-1}{}' \ \eta(q) \cdot \hat{\lambda}(q) \cdot \cos\left(\frac{\pi q(j+1/2)}{M}\right) \tag{9.32}$$

where $\widehat{\lambda}(q)$ are the *modified* eigenvalues defined by the *eigenvalue folding* scheme

$$\widehat{\lambda}(q) \equiv \begin{cases} \lambda(0)U_M^2(0) & q=0 \\ \lambda(q)U_M^2(q) + \lambda(2M-q)U_M^2(2M-q) & q=1:M-1 \end{cases} \tag{9.33}$$

A graphical illustration for the folding scheme is given in Figure 9-2 for the case $M = 8$ and $K = 16$. Assuming that the modified eigenvalues $\{\widehat{\lambda}(q)\}$ have been computed and stored, we now have a fast algorithm to compute potentials from currents on a 1-D cell array of size M. Given the M cell currents $\{f(i)\}$, we first calculate $\{\eta(q)\}$ defined in (9.26) using the 1-D Type-2 DCT. Then $\{\eta(q)\}$ is multiplied by the modified eigenvalues $\{\widehat{\lambda}(q)\}$ term-by-term. Finally, a 1-D Type-2 inverse DCT (IDCT) is performed on the resulting array $\{\eta(q)\widehat{\lambda}(q)\}$ to give the average cell potentials $\Phi(j)$ as prescribed by (9.32).

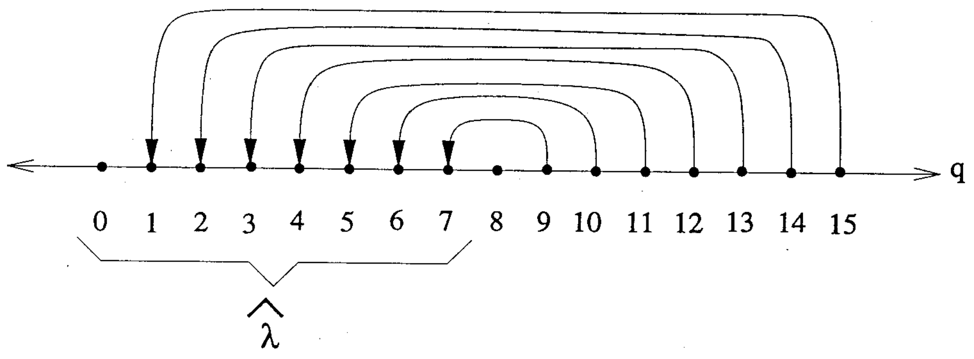


FIGURE 9-2: Eigenvalue folding for $M = 8$ and $K = 16$.

It is easy to include more cosine terms in the expansion (9.12) without increasing the size of the DCT computation. For the case $K = 4M$, in addition to (9.25), we have the following identities

$$\begin{aligned} \cos\left(\frac{\pi(2M+q)(i+1/2)}{M}\right) &= -\cos\left(\frac{\pi q(i+1/2)}{M}\right) \\ \cos\left(\frac{\pi(3M+q)(i+1/2)}{M}\right) &= \cos\left(\frac{\pi q'(i+1/2)}{M}\right) \end{aligned} \tag{9.34}$$

where $q = 0 : M-1$ and $q' = M-q$. Similarly, it is easy to see that

$$\begin{aligned} \widetilde{B}(2M+q) &= -\eta(q) \ , \\ \widetilde{B}(3M+q) &= \eta(q') \ . \end{aligned} \tag{9.35}$$

Use of (9.31),(9.34), and (9.35) in (9.21) results in

$$\begin{aligned} \Phi(j) = & \left(\frac{2}{a}\right) \sum_{q=0}^{M-1}{}' \ \eta(q) \cdot [\lambda(q) \cdot U_M^2(q)] \cdot \cos\left(\frac{\pi q(j+1/2)}{M}\right) \\ & + \left(\frac{2}{a}\right) \sum_{q=1}^{M-1} \eta(q) \cdot [\lambda(2M-q) \cdot U_M^2(2M-q)] \cdot \cos\left(\frac{\pi q(j+1/2)}{M}\right) \end{aligned}$$

$$+\left(\frac{2}{a}\right)\sum_{q=1}^{M-1}\eta(q)\cdot[\lambda(2M+q)\cdot U_M^2(2M+q)]\cdot\cos\left(\frac{\pi q(j+1/2)}{M}\right)$$
$$+\left(\frac{2}{a}\right)\sum_{q=1}^{M-1}\eta(q)\cdot[\lambda(4M-q)\cdot U_M^2(4M-q)]\cdot\cos\left(\frac{\pi q(j+1/2)}{M}\right) \qquad (9.36)$$

which again leads to the simplified expression (9.32) for $\Phi(j)$, except that the modified eigenvalues are now computed using a folding and shifting scheme

$$\begin{aligned}\widehat{\lambda}(q) \;\equiv\; & \lambda(q)U_M^2(q)+\lambda(2M-q)U_M^2(2M-q)+\lambda(2M+q)U_M^2(2M+q)\\ & +\lambda(4M-q)U_M^2(4M-q)\ , \quad q=1:M-1\end{aligned} \qquad (9.37)$$

and $\widehat{\lambda}(0) = \lambda(0)U_M^2(0)$. Using the modified eigenvalue approach, we see that only a size-M DCT transform is required regardless of the size of K.

It is easy to generalize the ideas developed thus far to the two-dimensional problem. Given an $M_x \times M_y$ array of cell currents $f(j_x, j_y)$, we first compute its 2-D Type-2 DCT $\eta(q_x, q_y)$ as

$$\eta(q_x,q_y)=\sum_{j_x=0}^{M_x-1}\sum_{j_y=0}^{M_y-1} f(j_x,j_y)\cdot\cos\left(\frac{\pi q_x(j_x+1/2)}{M_x}\right)\cdot\cos\left(\frac{\pi q_y(j_y+1/2)}{M_y}\right) \qquad (9.38)$$

where $q_x = 0 : M_x - 1$ and $q_y = 0 : M_y - 1$. The average cell potentials $\Phi(j_x, j_y)$ are computed via a 2-D Type-2 inverse DCT

$$\Phi(j_x,j_y)=\left(\frac{4}{ab}\right)\sum_{q_x=0}^{M_x-1}{}'\sum_{q_y=0}^{M_y-1}{}'\;\eta(q_x,q_y)\cdot\widehat{\lambda}(q_x,q_y)\cdot\cos\left(\frac{\pi q_x(j_x+1/2)}{M_x}\right)\cdot\cos\left(\frac{\pi q_y(j_y+1/2)}{M_y}\right). \qquad (9.39)$$

where the modified eigenvalues $\widehat{\lambda}(q_x, q_y)$ can be computed via folding and shifting just as in the 1-D case. For the case $K_x = 2M_x$ and $K_y = 2M_y$, we have

$$\widehat{\lambda}(q_x,q_y)\equiv\begin{cases}\lambda(0,0)U_M^2(0,0) & q_x=0, q_y=0\\ \lambda(0,q_y)U_M^2(0,q_y)+\lambda(0,2M_y-q_y)U_M^2(0,2M_y-q_y) & q_x=0, q_y>0\\ \lambda(q_x,0)U_M^2(q_x,0)+\lambda(2M_x-q_x,0)U_M^2(2M_x-q_x,0) & q_x>0, q_y=0\end{cases} \qquad (9.40)$$

and

$$\begin{aligned}\widehat{\lambda}(q_x,q_y) \;\equiv\; & \lambda(q_x,q_y)U_M^2(q_x,q_y)+\\ & \lambda(2M_x-q_x,q_y)U_M^2(2M_x-q_x,q_y)+\\ & \lambda(q_x,2M_y-q_y)U_M^2(q_x,2M_y-q_y)+\\ & \lambda(2M_x-q_x,2M_y-q_y)U_M^2(2M_x-q_x,2M_y-q_y)\ , \quad q_x>0,\ q_y>0.\end{aligned} \qquad (9.41)$$

Similar formulas can be derived for the case $K_x = 4M_x$, $K_y = 4M_y$. Hence, we have derived an eigendecomposition based sparsification technique to compute the $M_x \times M_y$ array of average cell potentials given the $M_x \times M_y$ array of net cell currents.

In practice, since only a fraction of the $M_x \times M_y$ cells correspond to actual *panels* from substrate contacts, it is necessary to *zero-pad* the vector of panel currents q to fill the 2-D data array $f(j_x, j_y)$. After the average cell potentials $\Phi(j_x, j_y)$ everywhere have been computed, the vector of panel potentials v are *lifted* as a subset of this 2-D array. This process is depicted in Figure 9-3. Let N be the number of panels, and let $M = M_x = M_y$. If we assume that the fraction of chip area occupied by substrate contacts is α, then $N = \alpha M^2$. Then computing a single Pq product via dense matrix-vector multiplication costs $\mathcal{O}(N^2) = \mathcal{O}(\alpha^2 M^4)$ operations, whereas the eigendecomposition approach costs only $\mathcal{O}(2M^2 \log_2(M))$.

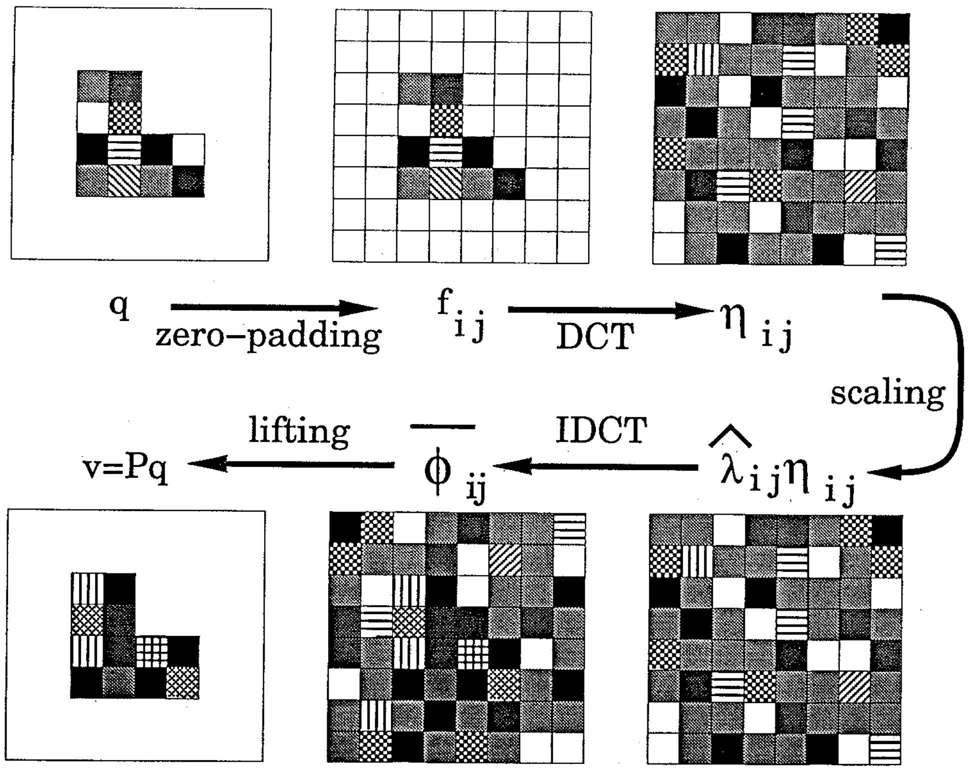


FIGURE 9-3: Sparsification via eigendecomposition.

In addition to being able to compute Pq efficiently given q, it is also necessary in the multigrid algorithm to compute and store interaction coefficients P_{lm} between nearby panels p_l and p_m. We show how this can be done efficiently using a 2-D Type-1 inverse DCT. Of course, the panel-to-panel calculations must be *consistent* with the results given by the eigendecomposition algorithm. Let the source panel p_m be located at position (m_x, m_y) in the 2-D cell array, and similarly let the target panel p_l be located at position (l_x, l_y). To find the potential due to a unit current source distributed uniformly over the cell at (m_x, m_y), we set $f(j_x, j_y) = \delta_{j_x, m_x} \delta_{j_y, m_y}$, where $\delta_{i,j}$ is the Kroneker delta. This turns (9.38) into

$$\eta(q_x, q_y) = \cos\left(\frac{\pi q_x (m_x + 1/2)}{M_x}\right) \cdot \cos\left(\frac{\pi q_y (m_y + 1/2)}{M_y}\right). \tag{9.42}$$

Substituting (9.42) into (9.39), and setting the cell-to-cell interaction coefficient $P(l_x, l_y; m_x, m_y)$ equal to $\Phi(l_x, l_y)$, we get

$$\begin{aligned} P(l_x, l_y; m_x, m_y) &= \left(\frac{4}{ab}\right) \sum_{q_x=0}^{M_x-1}{}' \sum_{q_y=0}^{M_y-1}{}' \widehat{\lambda}(q_x, q_y) \cdot \cos\left(\frac{\pi q_x(l_x+1/2)}{M_x}\right) \cdot \cos\left(\frac{\pi q_y(l_y+1/2)}{M_y}\right) \cdot \\ & \cos\left(\frac{\pi q_x(m_x+1/2)}{M_x}\right) \cdot \cos\left(\frac{\pi q_y(m_y+1/2)}{M_y}\right). \end{aligned} \tag{9.43}$$

With the help of simple trigonometric identities, (9.43) can be rewritten as

$$\begin{aligned} P(l_x, l_y; m_x, m_y) &= \left(\frac{4}{ab}\right) \sum_{q_x=0}^{M_x-1}{}' \sum_{q_y=0}^{M_y-1}{}' \widehat{\lambda}(q_x, q_y) \cdot \left(\frac{1}{4}\right) \\ & \cdot \left\{\cos\left(\frac{\pi q_x(l_x+m_x+1)}{M_x}\right) + \cos\left(\frac{\pi q_x(l_x-m_x)}{M_y}\right)\right\} \\ & \cdot \left\{\cos\left(\frac{\pi q_y(l_y+m_y+1)}{M_y}\right) + \cos\left(\frac{\pi q_y(l_y-m_y)}{M_y}\right)\right\}. \end{aligned} \tag{9.44}$$

Define the array $\omega(q_x, q_y)$ as a scaled version of $\widehat{\lambda}(q_x, q_y)$

$$\omega(q_x, q_y) \equiv \left(\frac{M_x M_y}{4ab}\right) \cdot \widehat{\lambda}(q_x, q_y) \tag{9.45}$$

Using the definition of the 2-D Type-1 inverse DCT in (9.7), we now rewrite (9.44) as

$$\begin{aligned} P(l_x, l_y; m_x, m_y) &= \overline{\omega}(l_x+m_x+1, l_y+m_y+1) + \overline{\omega}(l_x+m_x+1, l_y-m_y) \\ & + \overline{\omega}(l_x-m_x, l_y+m_y+1) + \overline{\omega}(l_x-m_x, l_y-m_y). \end{aligned} \tag{9.46}$$

The fact that $\overline{\omega}(j_x, j_y)$ has indexes in the range $j_x = 0 : M_x$ and $j_y = 0 : M_y$ requires that (9.46) be evaluated with the help of simple identities

$$\begin{aligned} \overline{\omega}(-j_x, j_y) &= \overline{\omega}(j_x, j_y) \\ \overline{\omega}(2M_x - j_x, j_y) &= \overline{\omega}(j_x, j_y) \end{aligned} \tag{9.47}$$

and

$$\begin{aligned} \overline{\omega}(j_x, -j_y) &= \overline{\omega}(j_x, j_y) \\ \overline{\omega}(j_x, 2M_y - j_y) &= \overline{\omega}(j_x, j_y). \end{aligned} \tag{9.48}$$

Once the $(M_x+1) \times (M_y+1)$ array $\overline{\omega}(j_x, j_y)$ has been computed via a 2-D Type-1 inverse DCT from the modified eigenvalue array $\widehat{\lambda}(q_x, q_y)$, it serves as a *look-up table* from which individual, panel-to-panel interaction coefficients can be extracted via (9.46) It is not difficult to check that (9.46) is consistent with the definition of the Galerkin matrix element in (7.5) if the the Green's function $G(r; r')$ is taken to be the first $K_x \times K_y$ terms in the infinite series (7.11).

10

Multigrid Method on Regular Domains

For the solution of elliptic partial differential equations discretized with the finite-difference or the finite-element mesh, multigrid methods are well-developed and known to be the most efficient class of numerical schemes [44, 45, 46]. In the area of integral equations, however, multigrid-style methods have received much less attention, although there exist some literature on second-kind integral equations defined on simple curves or surfaces [45, 2]. For the solution of first-kind integral equations on an irregular, multiply-connected surface, as is the case here for the substrate coupling problem, we are not aware of any previous work based on the multigrid idea. We address this void by developing the many algorithmic components necessary for an efficient multigrid-style solution scheme. The ideas proposed here can be generalized to solve other problems arising from first-kind integral equations defined over complicated surfaces, such as the boundary-element (BEM) based capacitance extraction [3] problem.

To best present the general multigrid method, we first describe in this chapter the simpler case of a uniformly discretized contact that covers the entire substrate. We then describe the modifications needed for many irregularly shaped contacts in Chapter 11.

10.1 Two-Grid Method (TGM)

In this section, we assume that the integral equation is defined over the *entire* substrate $\Omega = [0,a] \times [0,b]$

$$\phi(\mathbf{r}) = \int_{\Omega} \rho_S(\mathbf{r}')G(\mathbf{r};\mathbf{r}')da', \qquad \mathbf{r} \in \Omega, \tag{10.1}$$

and that Ω is discretized into a uniform array of $M \times M$ square panels. We assume further that M is a power of two, *i.e.* $M = 2^l$ for integer l. The number of panel unknowns, and hence the size of the linear system $Pq = v$, is then $N_l = M^2$. We refer to this discrete IE system as a level l, or *fine-grid*, representation of (10.1)

$$P_{\{l\}} \cdot q_{\{l\}} = v_{\{l\}}. \tag{10.2}$$

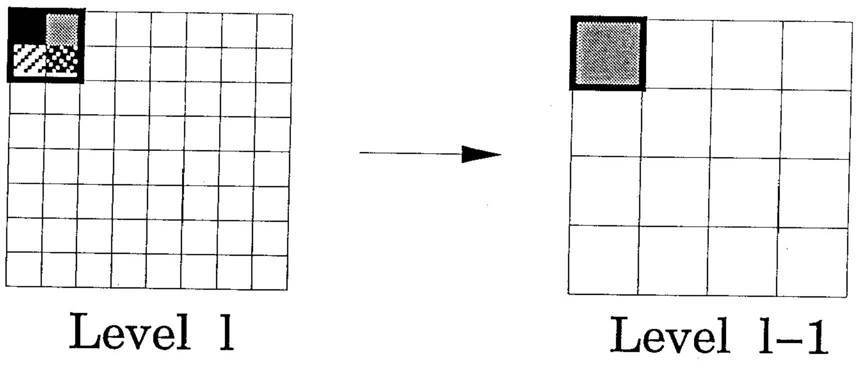


FIGURE 10-1: Two-level Representation and Restriction for Uniform Grid Problem.

Suppose we also discretize (10.1) using a coarser, uniform $(M/2) \times (M/2)$ array of panels, yielding a discrete linear system of size $N_{l-1} = (M^2/4)$. This results in a level $(l-1)$, or *coarse-grid*, representation

$$P_{\{l-1\}} \cdot q_{\{l-1\}} = v_{\{l-1\}}. \tag{10.3}$$

See Figure 10-1 for the two discretizations.

```
Algorithm 4 (Two-Grid Method (TGM) for solving $P_{\{l\}} \cdot q_{\{l\}} = v_{\{l\}}$).

Set $k \equiv 1, q_{\{l\}}^{(1)} \equiv 0$.
Repeat {
  Fine-Grid Smoothing:
    Solve $D_{\{l\}} \cdot \Delta q^*_{\{l\}} = -P_{\{l\}} \cdot q_{\{l\}}^{(k)} + v_{\{l\}}$ for $\Delta q^*_{\{l\}}$.
    Compute intermediate guess $q^*_{\{l\}} = q_{\{l\}}^{(k)} + \Delta q^*_{\{l\}}$.
  Compute residual $u_l = P_{\{l\}} \cdot q^*_{\{l\}} - v_{\{l\}}$.
  Project to coarse grid $u_{\{l-1\}} = r u_{\{l\}}$.
  Coarse-Grid Correction:
    Solve for $\Delta q_{\{l-1\}}$ in $P_{\{l-1\}} \cdot (\Delta q_{\{l-1\}}) = u_{\{l-1\}}$.
    Project to fine grid $\Delta q_{\{l\}} = p(\Delta q_{\{l-1\}})$.
    Update intermediate guess $q_{\{l\}}^{(k+1)} = q^*_{\{l\}} - \Delta q_{\{l\}}$.
  Set $k = k + 1$.
} Until residual norm $\|u_{\{l\}}\| < \varepsilon$.
```

Solving the fine-grid problem (10.2) by direct matrix factorization is impractical for large N_l since $P_{\{l\}}$ is dense. However, it may be possible to factor the smaller matrix $P_{\{l-1\}}$ corresponding to the coarse-grid problem (10.3), since $N_{l-1} = N_l/4$. This motivates our development of a *two-grid method* (TGM), in which the problem is solved *iteratively* at level l with the help of direct solution at level $(l-1)$. The two principal algorithmic components, analogous to TGM for PDE's [44, 45, 46], are the *smoothing* operator and the *intergrid transfer*, or restriction-prolongation, operators. In our TGM iteration for solving (10.2), the *error* in the k-th iterate, $q_{\{l\}}^{(k)}$, is *smoothed* by carefully solving a series of local problems. This first stage is typically called fine-grid smoothing or *relaxation*, and results in an intermediate guess $q^*_{\{l\}}$. Next, we compute the *residual* $u_{\{l\}} = P_{\{l\}} \cdot q^*_{\{l\}} - v_{\{l\}}$ and project it onto the coarse grid via $u_{\{l-1\}} = ru_{\{l\}}$, where r is a *restriction* operator. Then we solve explicitly the coarse-grid problem $P_{\{l-1\}} \cdot (\Delta q_{\{l-1\}}) = u_{\{l-1\}}$ for $\Delta q_{\{l-1\}}$, and project the result onto the fine grid via $\Delta q_{\{l\}} = p(\Delta q_{\{l-1\}})$, where p is a *prolongation* operator. Finally, the intermediate guess on the fine grid is updated to yield the $(k+1)$-st iterate $q_{\{l\}}^{(k+1)} = q^*_{\{l\}} - \Delta q_{\{l\}}$. This second stage is termed coarse-grid correction and is responsible for "long-range" interactions. The fine-grid smoothing/coarse-grid correction cycle is repeated until the norm of the residual $u_{\{l\}}$ is below some tolerance. The entire two-grid method is summarized in Algorithm 4, where the matrix D is described below.

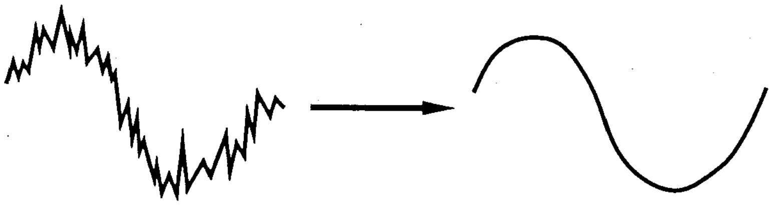


FIGURE 10-2: Smoothing of error at fine grid.

Since fine-grid smoothing is responsible for reducing only the "high-frequency" components of the error (as shown in Figure 10-2), and since the resulting, *smoothed* error is well-represented on the coarse grid where explicit solution is performed, the two-grid scheme effectively *decouples* the original problem into a high-frequency sub-problem and a low-frequency sub-problem. This decoupling is depicted in Figure 10-3, which shows three-quarters of the total number of eigenvalues being classified as "high-frequency" and handled by fine-grid relaxation. The remaining one quarter of eigenvalues, classified as "low-frequency", are handled by the explicit solution at the coarse grid. The fact that the high-frequency eigenvalues occupy an interval half as large as that of the low-frequency eigenvalues is a direct consequence of the eigenvalue relation $\lambda_{mn} \sim 1/\sqrt{m^2+n^2}$.

To derive the smoothing operator, we first make the important observation that the IE matrix P is derived from a Green's function $G(\mathbf{r};\mathbf{r}')$ that is sharply peaked as $\mathbf{r} \to \mathbf{r}'$, but is *smooth* otherwise, *i.e.* $|\mathbf{r}-\mathbf{r}'| > d$ for some distance d. We seek an *operator splitting* at level l

$$P_{\{l\}} = D_{\{l\}} + S_{\{l\}}, \tag{10.4}$$

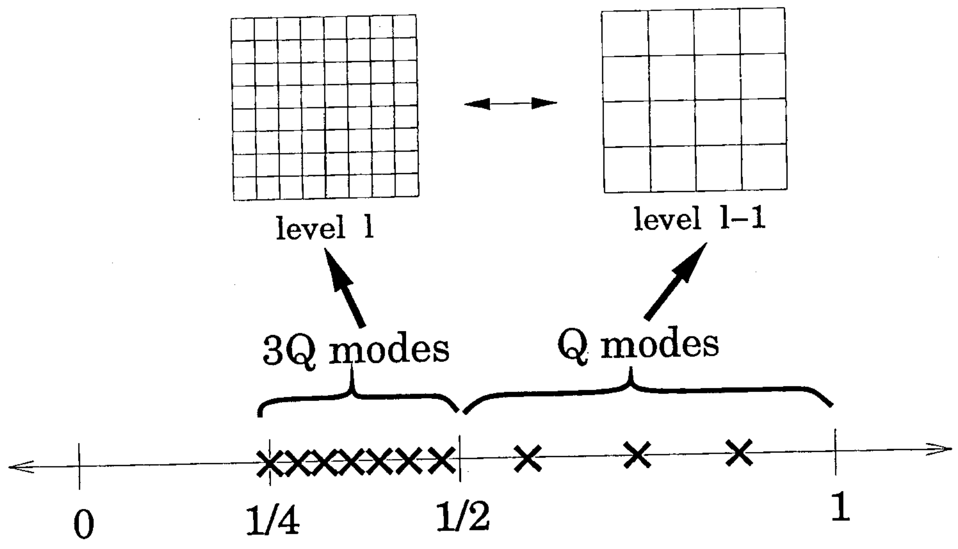


FIGURE 10-3: Two-grid method and eigenspectrum.

such that $D_{\{l\}}$ captures the short-range, sharply-peaked portion of $P_{\{l\}}$, and $S_{\{l\}}$ captures the long-range, smooth portion of $P_{\{l\}}$. See Figure 10-4 for a rough depiction. Given (10.4), we define the smoothing operator as the result of solving

$$D_{\{l\}} \cdot q^{*}_{\{l\}} = -S_{\{l\}} \cdot q^{(k)}_{\{l\}} + v_{\{l\}} \tag{10.5}$$

for the vector $q^{*}_{\{l\}}$. Equation (10.5) defines a *fixed-point* iteration [14], since the condition $q^{(k)}_{\{l\}} = q_{\{l\}}$, where $q_{\{l\}}$ is the exact solution of (10.2), would lead to $q^{*}_{\{l\}} = q_{\{l\}}$. Since it is necessary that the above smoothing step be done cheaply, we require that $D_{\{l\}}$ be easy to invert, or that $D^{-1}_{\{l\}}$ has a *sparse* matrix structure, since

$$q^{*}_{\{l\}} = D^{-1}_{\{l\}} \cdot \left(-S_{\{l\}} \cdot q^{(k)}_{\{l\}} + v_{\{l\}}\right). \tag{10.6}$$

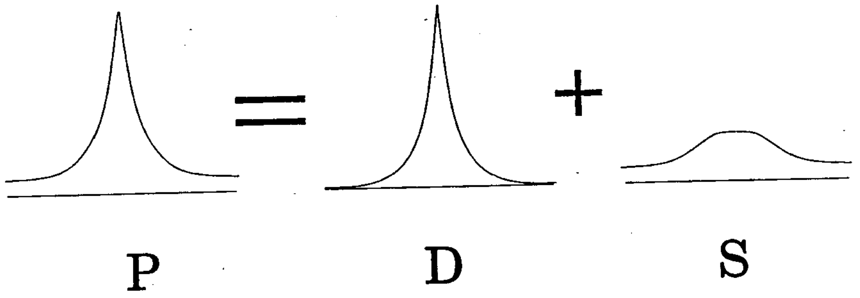


FIGURE 10-4: Operator Splitting $P = D + S$.

We now determine whether it is more efficient to contruct $D_{\{l\}}$ as an approximation of $P_{\{l\}}$ or $D^{-1}_{\{l\}}$ as an approximation of $P^{-1}_{\{l\}}$. First, we note that the matrix $P_{\{l\}}$ is "non-local", since its matrix elements decay roughly as $1/r$ away from the source. This slow rate of decay makes it difficult to truncate elements of $P_{\{l\}}$. On the other hand, the matrix $P_{\{l\}}^{-1}$ is a *capacitance matrix* and hence is strictly diagonally dominant. This property implies that $P_{\{l\}}^{-1}$ is "local", and can be truncated to form a sparse approximation $D_{\{l\}}^{-1}$. Since $P_{\{l\}}$ is the discrete analog of a smoothing operator (pseudo-differential operator of order -1), the capacitance matrix $P_{\{l\}}^{-1}$ corresponds to a differential operator of order $+1$. It is to be expected that sparse matrix approximations can be more easily constructed for differential operators than for smoothing operators. This is the reason why multigrid relaxation algorithms are easy to construct when solving elliptic PDE's.

Hence, we construct directly a sparse matrix $D_{\{l\}}^{-1}$ based on the overlapping, local-inversion *preconditioner* developed in Fastcap [3], a BEM-based capacitance-extraction program. Similar ideas have been proposed in [59]. Our particular implementation is outlined as follows. For each panel $p_k^{\{l\}}$, a local coefficient-of-potential matrix $P_{loc}^{\{l\}}[k]$ involving only itself and its immediate neighbors is constructed. For the uniform grid problem, the size of $P_{loc}^{\{l\}}[k]$ is at most 9×9, as shown in Figure 10-5. This small matrix is easily inverted to yield $(P_{loc}^{\{l\}}[k])^{-1}$ whose elements from the *row* corresponding to panel k are then extracted and stamped into corresponding locations in the k-th *row* of $D_{\{l\}}^{-1}$. Hence, the matrix $D_{\{l\}}^{-1}$ contains at most 9 non-zero entries per row and is *sparse.* We recall that the panel-to-panel interaction coefficient can be computed inexpensively from a $M \times M$ DCT array described in Chapter 7.

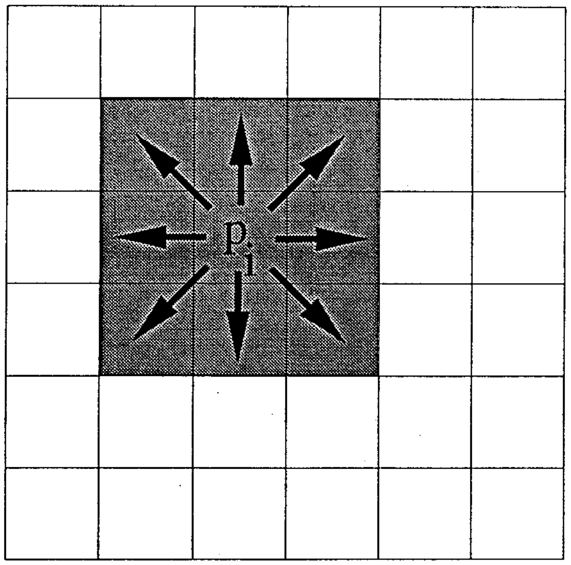


FIGURE 10-5: Local coefficient of potential matrix P_{loc}.

Because we do not construct $D_{\{l\}}$ directly, it may seem at first glance that the matrix $S_{\{l\}} \equiv P_{\{l\}} - D_{\{l\}}$ is difficult to obtain. But if we subtract $(D_{\{l\}} \cdot q_{\{l\}}^{(k)})$ from both sides of (10.5) and then multiplying through by $D_{\{l\}}^{-1}$, the result

$$\Delta q_{\{l\}}^{*} = D_{\{l\}}^{-1} \cdot \left(-P_{\{l\}} \cdot q_{\{l\}}^{(k)} + v_{\{l\}}\right) \tag{10.7}$$

can be used to compute $q_{\{l\}}^{*}$

$$q_{\{l\}}^{*} = q_{\{l\}}^{(k)} + \Delta q_{\{l\}}^{*}. \tag{10.8}$$

We notice that the modified smoothing steps (10.7) and (10.8) require only operators $D_{\{l\}}^{-1}$ and $P_{\{l\}}$ which are readily available.

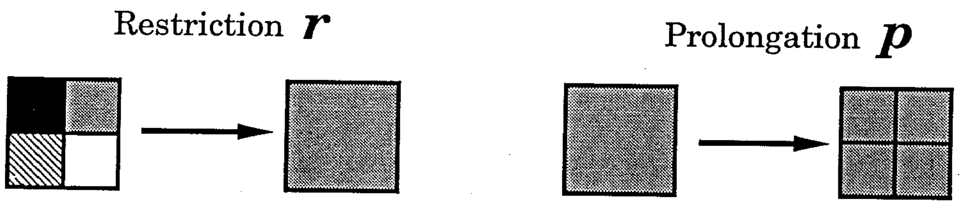


FIGURE 10-6: Intergrid transfers for simple domain.

In addition to the smoothing operator, we require transfer operators r and p between the two grids. They are trivial in the case of uniform grids, where a coarse-grid panel, called a *parent*, is composed of four fine-grid panels, called *kids*. See Figure 10-6. Recall that in the Galerkin formulation, q_i is the net current on panel i, and v_j the average potential on panel j. For the restriction operator r mapping from level l to level $(l-1)$, the net current on a parent is simply the *sum* of the currents on the four kids, and the average potential on a parent is the average of the four kid potentials. The prolongation operator p mapping from level $(l-1)$ to level l is defined as the *adjoint*, or transpose, of the restriction r [46].

10.2 Multigrid Method (MGM)

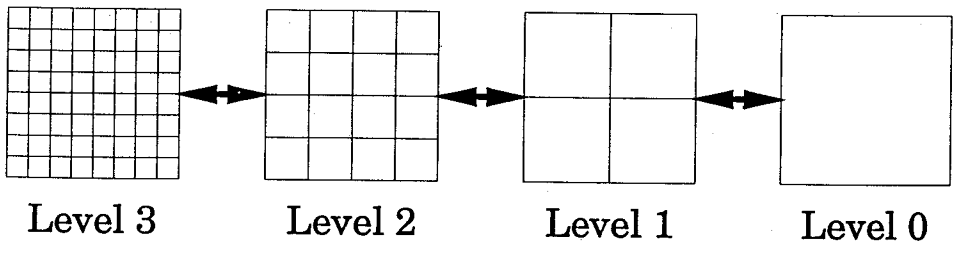


FIGURE 10-7: Multilevel discretization of simple domain.

The *multigrid method* (MGM) is the generalization of the two-grid method to an arbitrary number of levels. Instead of solving the problem (10.3) explicitly at level $(l-1)$, which may still be too expensive, we apply a similar smoothing-correction cycle at level $(l-1)$. In the same manner, the correction cycle at level $(l-1)$ becomes a smoothing-correction cycle at level $(l-2)$, and so on. The integral equation (10.1) is now discretized at all levels $\{l_{min}, \ldots, l_{max}\}$, as shown in Figure 10-7. Only at the coarsest level, $l = l_{min}$, is the system $P_{\{l\}} \cdot q_{\{l\}} = v_{\{l\}}$ solved explicitly. Each multigrid iteration is best described as a *recursion*, and is summarized in Algorithm 5. Notice the recursive functional call **MGM** in the algorithm. The function **MGM**$(l, q_{\{l\}}, v_{\{l\}})$ is called repeatedly at the finest level ($l = l_{max}$) until it has been determined externally that the desired accuracy is achieved. Algorithm 5 describes the basic multigrid V-cycle, although the W-cycle and the full multigrid (FMG) cycle can also be easily implemented.

Algorithm 5 (Multigrid Iteration (MGM) for solving $P_{\{l\}} \cdot q_{\{l\}} = v_{\{l\}}$).

```
MGM($l, q_{\{l\}}, v_{\{l\}}$) {
  If ($l = l_{min}$)
    Solve $P_{\{l\}} \cdot q_{\{l\}} = v_{\{l\}}$ explicitly.
  Else {
    Fine-Grid Smoothing:
      Solve $D_{\{l\}} \cdot \Delta q^*_{\{l\}} = -P_{\{l\}} \cdot q^{(k)}_{\{l\}} + v_{\{l\}}$ for $\Delta q^*_{\{l\}}$.
      Compute intermediate guess $q^*_{\{l\}} = q^{(k)}_{\{l\}} + \Delta q^*_{\{l\}}$.
    Compute residual $u_{\{l\}} = P_{\{l\}} \cdot q^*_{\{l\}} - v_{\{l\}}$.
    Restriction $u_{\{l-1\}} = r u_{\{l\}}$.
    Coarse-Grid Correction:
      Set initial guess $\Delta q_{\{l-1\}} = 0$.
      Do MGM($l-1, \Delta q_{\{l-1\}}, u_{\{l-1\}}$).
      Prolongation $\Delta q_{\{l\}} = p(\Delta q_{\{l-1\}})$.
      Update intermediate guess $q^{(k+1)}_{\{l\}} = q^*_{\{l\}} - \Delta q_{\{l\}}$.
  }
}
```

The manner in which multigrid methods accelerate iterative convergence can be visualized with the eigenspectrum in Figure 10-8. Suppose, for example, that a grand total of $64Q$ eigenmodes exist in the discrete system, and that their corresponding eigenvalues range between $1/16$ and 1. Then a four level multigrid scheme will break up the eigenspectrum into four "clusters" approximately as shown in Figure 10-8. The finest grid relaxation is responsible for the $48Q$ lowest eigenvalues, the second-finest grid for the next $12Q$ eigenvalues, and so on. Because the ratio of largest-to-smallest eigenvalues within each cluster is always 2, the "effective" condition number has been reduced to 2 from 16. This ratio of two is a direct

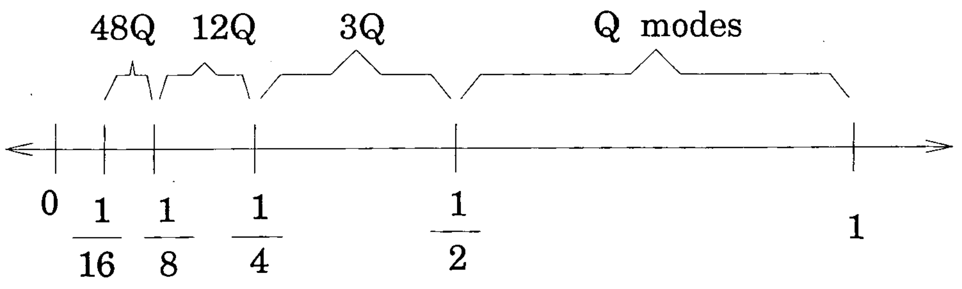


FIGURE 10-8: Eigenvalue clusters corresponding to multigrid levels.

consequence of the asymptotic eigenvalue distribution (8.74) and the $2x$ coarsening in each direction between successive grids. Hence, multigrid schemes converge at a constant rate, independent of mesh refinement.

10.3 Smoothing Results and High-Pass Filtering

To see how the matrix splitting $P_{\{l\}} = D_{\{l\}} + S_{\{l\}}$ achieves the desired *smoothing* effect on the error, we note that the smoothing iteration (10.6) implies

$$e^*_{\{l\}} = -D^{-1}_{\{l\}} S_{\{l\}} \cdot e^{(k)}_{\{l\}}, \tag{10.9}$$

where $e^{(k)}_{\{l\}} \equiv q^{(k)}_{\{l\}} - q_{\{l\}}$ and $e^*_{\{l\}} \equiv q^*_{\{l\}} - q_{\{l\}}$ are the errors before and after smoothing, respectively. Hence we define

$$M_{\{l\}} \equiv -D^{-1}_{\{l\}} S_{\{l\}} \tag{10.10}$$

as the smoothing matrix at level l. If $M_{\{l\}}$ is used in isolation as a relaxation algorithm, then (10.9) requires that $\|M_{\{l\}}\| < 1$ for convergence. However, since $M_{\{l\}}$ is used only as a *smoother* at level l, we require only that its eigenvalues be less than one for eigenmodes with "high" spatial frequencies. It is expensive to compute the eigenvectors and eigenvalues directly for $M_{\{l\}}$, since it is a dense matrix of size $M^2 \times M^2$. However, we can guage the effect of $M_{\{l\}}$ on a "delta" error, which is simply a current density function that takes on a non-zero, uniform value on a given panel p_i, and is zero everywhere else. This corresponds to a vector $e_i = [0, \ldots, 0, 1, 0, \ldots, 0]^t$ with a non-zero entry in the i-th position. For each panel p_i, the Fourier cosine (DCT) components of the error before and after smoothing (*i.e.* e_i and $M_{\{l\}} \cdot e_i$) can be compared term-by-term to observe the effect of smoothing at each spatial frequency. Figure 10-9 plots the *ratio* of the Fourier coefficients of the smoothed error to those of the original delta error on a 32×32 grid, with the delta error located near the center of the substrate. The vertical axis gives the reduction ratio, and the horizontal axis the Fourier mode index. Since the Fourier mode index (m, n) is two-dimensional, we have chosen to plot a transparent "side-view" of the 2-D data from one of the axes. Similar results are obtained for delta errors located elsewhere on the substrate. It is seen that almost all error components are reduced by a factor of 10, except at very low spatial frequencies, where error *magnification* is possible. The size of this magnification depends on the thickness of the substrate, as well as the level of discretization. This usually has no effect on convergence since the low-frequency errors are effectively dealt with at coarser levels. However, for cases involving very thick substrates or a very large number of levels, convergence may slow down, and even divergence is possible. To ensure optimal multigrid convergence in all situations, we insert an additional component into the multigrid algorithm, termed "high-pass filtering", which is aimed at explicitly removing any effect the smoother $M_{\{l\}}$ may have on low-frequency modes. For the case of a uniformly discretized contact which covers the entire substrate, the high-pass filter is simple to construct. At each level l, after the vector $\Delta q^*_{\{l\}}$ has been obtained, we remove any *projection* $\Delta q^*_{\{l\}}$ has on the next coarser level before computing the intermediate guess $q^*_{\{l\}}$. The modifications may be summarized as

$$\begin{aligned} \widetilde{\Delta q^*_{\{l\}}} &= (I - pr) \cdot \Delta q^*_{\{l\}} \\ q^*_{\{l\}} &= q^{(k)}_{\{l\}} + \widetilde{\Delta q^*_{\{l\}}}, \end{aligned} \tag{10.11}$$

where the operator $(I - pr)$ *orthogonalizes* a density function at level l against all density functions at level $(l-1)$.

To show that the two-grid and multigrid algorithms we have developed for simple domains indeed converge at the rate predicted by the behavior of the smoother $M_{\{l\}}$, we plot the error norm versus iteration count for a two-level ($L_{max} = 2$), three-level ($L_{max} = 3$), and four-level ($L_{max} = 4$) algorithm. Figure 10-10 displays results for a 32×32 grid, Figure 10-11 for a 128×128 grid. As expected, the error norm is reduced by an order of magnitude per iteration, *independent* of the number of multigrid levels applied or the grid size of the finest mesh.

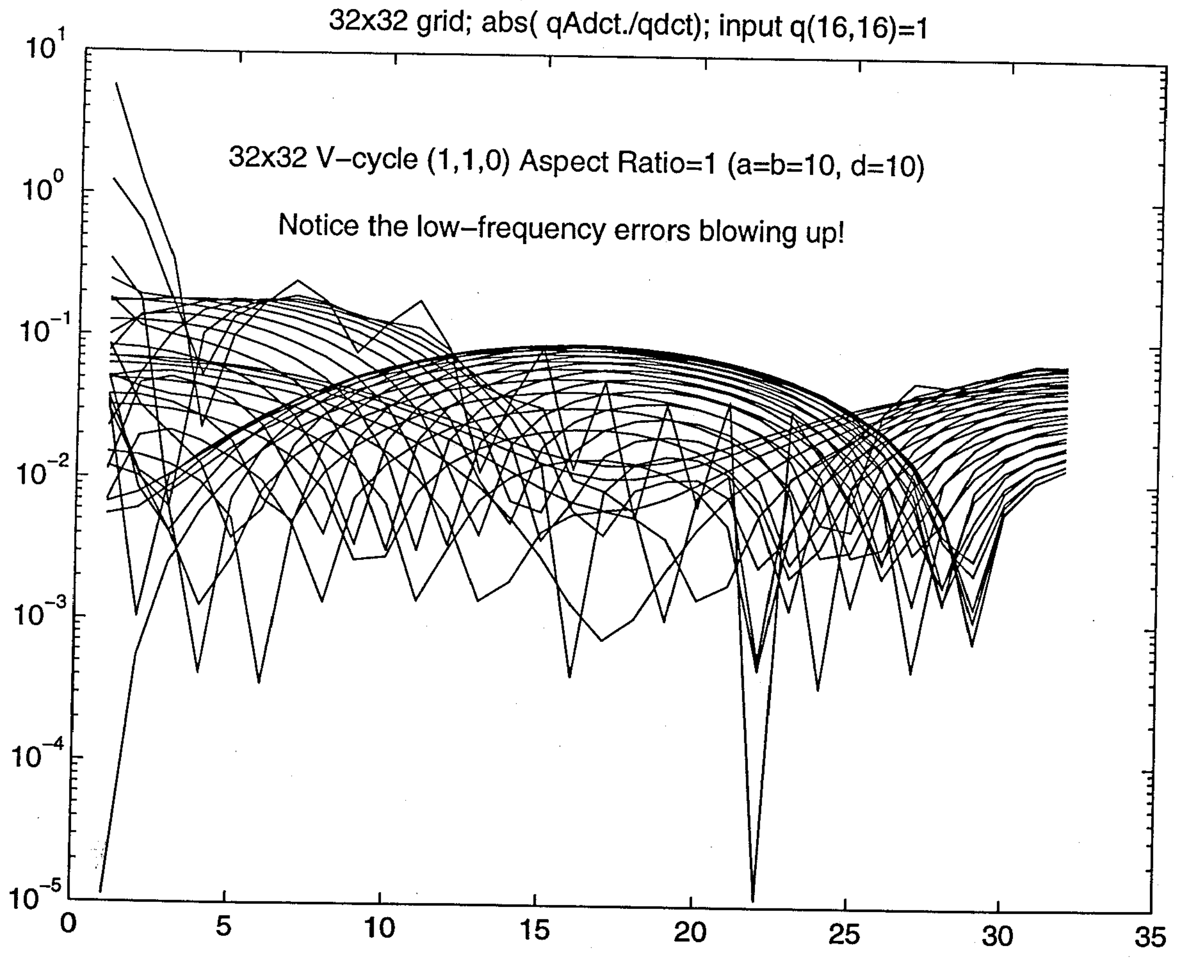


FIGURE 10-9: Effect of smoother $M_{\{l\}}$ on "delta" error.

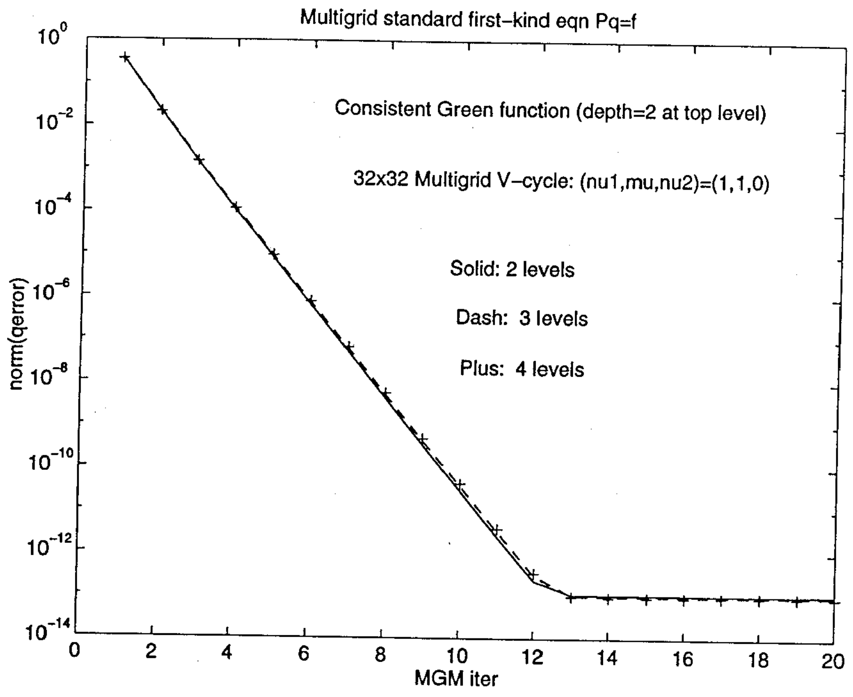


FIGURE 10-10: Multigrid convergence on 32×32 grid.

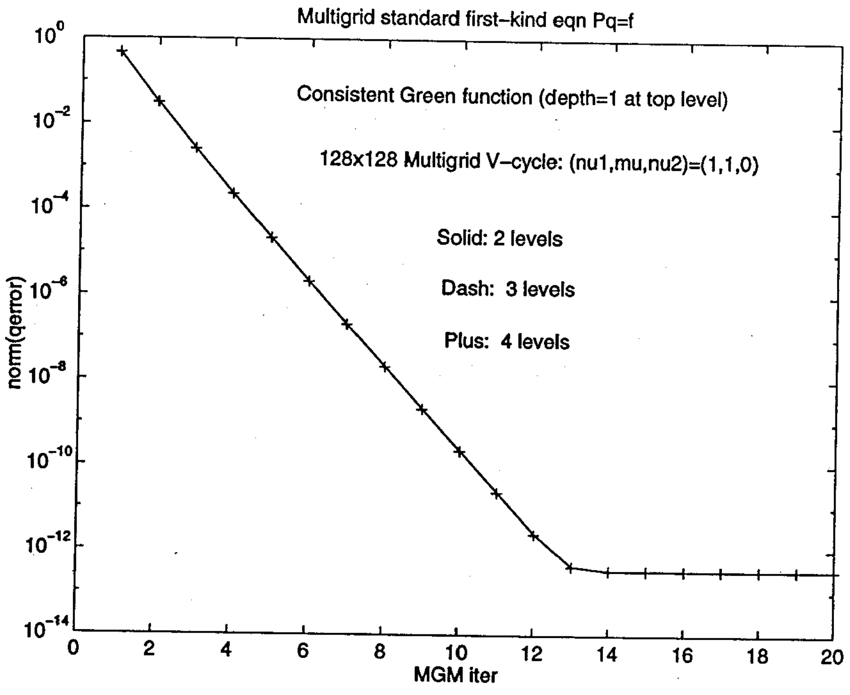


FIGURE 10-11: Multigrid convergence on 128×128 grid.

11

Multigrid Method on Irregular Domains

The basic idea behind any multigrid algorithm is as follows. Suppose that the original discretized problem can be solved on a *coarser grid*, then this coarse solution can be interpolated onto the original fine grid to form an approximate solution. Now all we need to do is to make *local corrections* on this approximate solution, since the long range interactions have been taken care of already on the coarse grid. Multigrid methods have been popular in the solution of PDE's using finite differences or finite elements in large part because it is straightforward to construct coarse-grid representations of the original discretization. However, in the case of *integral equations* defined over *irregular domains*, no previous work exists to the best of our knowledge on the construction of coarse-grid representations. This has been a major impediment to the adoption of multilevel methods by the integral equation community. Perhaps the most important contribution in this thesis is the concept of hierarchical basis functions which are derived from *characteristic functions* associated with panels in the original discretization. This leads naturally to a multilevel representation via the Galerkin approach, and is detailed in Section 11.1. Efficient algorithms to accelerate the matrix-vector multiplication at each level is described in Section 11.2.

11.1 Hierarchical Basis and Multilevel Representation

To handle layouts involving many irregularly shaped substrate contacts, we allow the integral equation (7.3) to be defined over an arbitrarily shaped surface S. Again, we make the assumption that S can be discretized into a collection of panels $\{p_i\}$, each of which coincides with a cell on a regular $M \times M$ array covering $\Omega \equiv [0, a] \times [0, b]$. This assumption is only mildly restrictive since most IC layouts are based on rectangular, or *Manhattan*, geometries. We also assume that M is a power of two. Of course not all of the $M \times M$ cells are occupied by panels. See Figure 7-5 for a simple example. This is the original discretization, which corresponds to the *finest-grid* representation. Recall that the linear system 7.4 results from a Galerkin discretization based on *constant strength* panels. Hence, we define here the *characteristic function*, $\mathcal{X}_i(r)$,associated with each panel p_i

$$\mathcal{X}_i(r) = \begin{cases} 1/a_i & \text{if } r \in p_i \\ 0 & \text{otherwise} \end{cases} \tag{11.1}$$

where a_i is the area of p_i. The Galerkin coefficients P_{ij} given in (7.5) is equivalent to the definition

$$P_{ij} = \int_S \int_S G(r;r') \cdot \mathcal{X}_i(r) \cdot \mathcal{X}_j(r') da\, da' \tag{11.2}$$

where the integrations are now over the entire surface S.

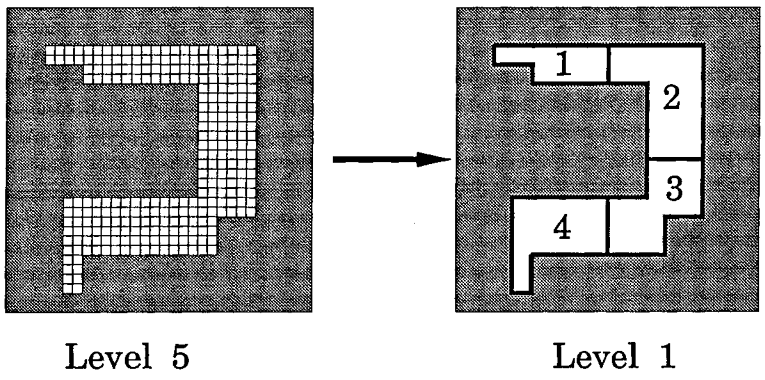


FIGURE 11-1: Characteristic functions at levels 1 and 5.

The explicit use of Galerkin characteristic functions allows us to construct coarser-level representations of the integral equation defined over complicated geometries. Let $M = 2^{l_{max}}$. Then the matrix elements given in (11.2) correspond to a discrete representation at level $l = l_{max}$, or the *finest level*

$$P^{\{l_{max}\}} \cdot q^{\{l_{max}\}} = v^{\{l_{max}\}}. \tag{11.3}$$

To construct coarser representations of (11.3), we need to first *define* panels at the coarser levels $l = 0 : (l_{max} - 1)$ in the following manner. At each level l, the rectangular domain Ω is covered with a regular $2^l \times 2^l$ array of level-l cells. For each non-empty level-l cell, a level-l panel is defined as the *union* of all finest-level panels within that cell. The k-th panel at level l is denoted by $p_k^{\{l\}}$, and is associated with a characteristic function $\mathcal{X}_k^{\{l\}}$ defined as

$$\mathcal{X}_k^{\{l\}}(r) = \begin{cases} 1/a_k^{\{l\}} & \text{if } r \in p_k^{\{l\}} \\ 0 & \text{otherwise} \end{cases} \tag{11.4}$$

where $a_k^{\{l\}}$ is the area of $p_k^{\{l\}}$. Given the set of hierarchical basis functions $\{\mathcal{X}_k^{\{l\}}\}$, we can now easily define discrete representations of the integral equation (7.3) at each level $l = 0 : l_{max}$

$$P^{\{l\}} \cdot q^{\{l\}} = v^{\{l\}} \tag{11.5}$$

where the Galerkin matrix elements $P_{ij}^{\{l\}}$ are computed as

$$P_{ij}^{\{l\}} = \int_S \int_S G(r;r') \cdot \mathcal{X}_i^{\{l\}}(r) \cdot \mathcal{X}_j^{\{l\}}(r') da\, da' \tag{11.6}$$

Figure 11-1 shows a finest-grid discretization at level 5 being mapped into a coarse representation with only four panels at level 1.

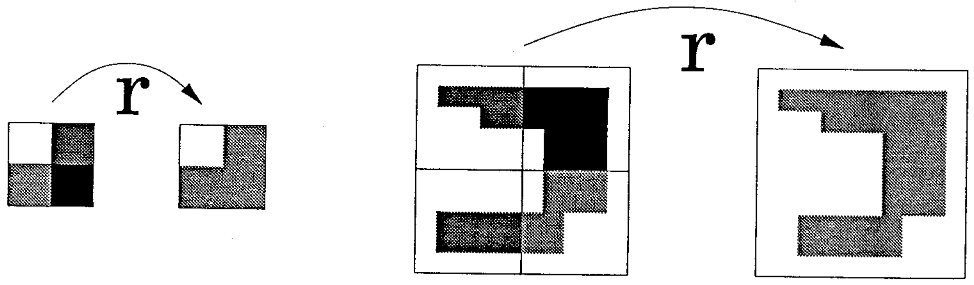


FIGURE 11-2: Restriction operator for irregular domain.

The use of multilevel characteristic functions leads to very straightforward definitions for the intergrid transfer operators. The *prolongation* operator p mapping from level l to $l+1$ is trivial to define since every characteristic function at level l is a linear combination of characteristic functions at level $l+1$. This can be expressed as

$$\mathcal{X}_i^{\{l\}} \in \text{span}\left\{\mathcal{X}_j^{\{l+1\}}\right\}. \tag{11.7}$$

If $H^{\{l\}}$ is defined as the space of functions

$$H^{\{l\}} \equiv \text{span}\left\{\mathcal{X}_j^{\{l\}}\right\}, \tag{11.8}$$

then we have the general relation

$$H^{\{0\}} \subset H^{\{1\}} \subset \cdots \subset H^{\{l_{max-1}\}} \subset H^{\{l_{max}\}} \quad . \tag{11.9}$$

Once the prolongation operator p is available, the restriction operator r is defined as its *adjoint* [46, 45]

$$r \equiv p^{\dagger} \tag{11.10}$$

An illustration of the restriction operator in action is given in Figure 11-2, which shows the combination of three finest-level panels into a coarse panel at level $(l_{max} - 1)$, and also the combination of four coarse panels into a single panel at the next coarser level. The high-pass filtering described in Section 10.3 can be easily implemented for irregular domains after the intergrid transfer operators have been constructed. Again, the filter is implemented as

$$\begin{aligned} \widetilde{\Delta q^*_{\{l\}}} &= (I - pr) \cdot \Delta q^*_{\{l\}} \\ q^*_{\{l\}} &= q^{(k)}_{\{l\}} + \widetilde{\Delta q^*_{\{l\}}}, \end{aligned} \tag{11.11}$$

For integral equations defined on irregular domains, we again define the operator splitting at each level l as

$$P_{\{l\}} = D_{\{l\}} + S_{\{l\}} \ , \tag{11.12}$$

and directly construct a sparse matrix $D_{\{l\}}^{-1}$ at each level using a slightly modified version of the algorithm described in Chapter 10 for regular domains. At a given level l, two panels $p_i^{\{l\}}$ and $p_j^{\{l\}}$ are considered to be *nearest neighbors* if the cells to which they belong share at least one vertex. By this definition, a panel is also its own nearest neighbor. For each panel $p_k^{\{l\}}$, a local coefficient-of-potential matrix $P_{loc}^{\{l\}}[k]$ is constructed from the interactions among its nearest neighbors. Then appropriate entries are taken from its inverse $(P_{loc}^{\{l\}}[k])^{-1}$ and stamped into the k-th row of $D_{\{l\}}^{-1}$. This is illustrated in Figure 11-3, in which the panel $p_k^{\{l\}}$ is shown to have six nearest neighbors including itself.

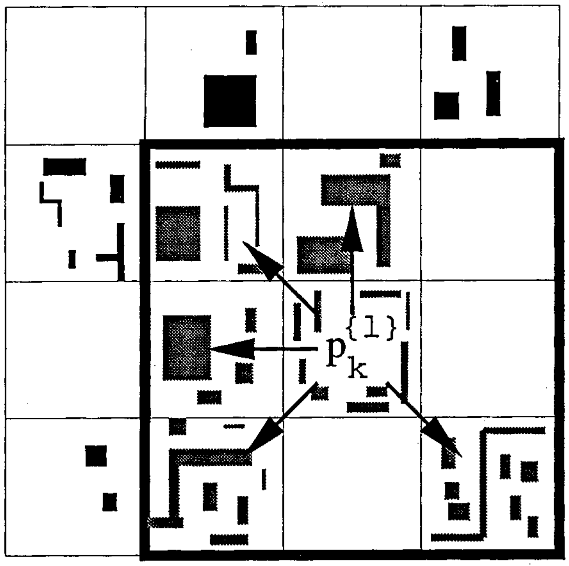


FIGURE 11-3: Local interactions on irregular domains.

Efficient algorthms are required to evaluate the Galerkin integrals in (11.6) and to perform the matrix-vector product (11.5) at each level. These are developed in Section 11.2.

11.2 Moment-Matching and Precorrected-DCT Acceleration

We have shown in Chapter 9 how to compute $P^{\{l\}} \cdot q^{\{l\}}$ efficiently at the finest level $l = l_{max}$, but it is also necessary to be able to construct the operators $P^{\{l\}}$ and to compute $P^{\{l\}} \cdot q^{\{l\}}$ efficiently at the coarser levels $l = 0 : (l_{max} - 1)$. Efficient implementation of multigrid schemes requires that the coarse-grid operators are cheaper to construct and to apply than at the finest grid. Idealy, the cost of applying the coarser-grid operator $P^{\{l-1\}}$ should be one-fourth that of the finer-grid operator $P^{\{l\}}$, since there are only one-fourth as many unknowns at the $(l-1)$

level. In practice, this is difficult to achieve, but multigrid schemes are efficient whenever the cost of an entire MG iteration is bounded by a small multiple of the cost required to compute the residual at the finest grid. In this section, we develop efficient techniques to *approximately* construct and apply the coarse-grid operators developed in Section 11.1.

To calculate a single Galerkin coefficient $P_{ij}^{\{l\}}$ between two coarse-grid panels defined by (11.6), it is sufficient to perform a double summation of the panel-to-panel coefficients $P_{ij}^{\{l_{max}\}}$ at the finest level. However, this leads to an $\mathcal{O}(N^2)$ algorithm. Instead, we make the observation that when two coarse panels $p_i^{\{l\}}$ and $p_j^{\{l\}}$ are "well-separated", their interaction coefficient $P_{ij}^{\{l\}}$ can be computed approximately by leaving out much of the detail in the characteristic functions $\mathcal{X}_i^{\{l\}}$ and $\mathcal{X}_j^{\{l\}}$. Similar ideas have been used extensively in multipole-accelerated algorithms[60, 3, 17]. For the multipole approximation used in [17], it was necessary to assume a substrate Green's function which has translational invariance and which can be fitted to a sum of polynomials in $(1/r)$, where $r = |r - r'|$. In contrast, we develop here a moment-matching method similar to [61] which can be used in combination with the cosine transform to accelerate the coarse-grid computations *and* account properly for all the substrate edge effects. This approach approximates the potentials produced by a coarse panel $p_i^{\{l\}}$ by constructing a simpler *representation* of the associated characteristic function $\mathcal{X}_i^{\{l\}}$.

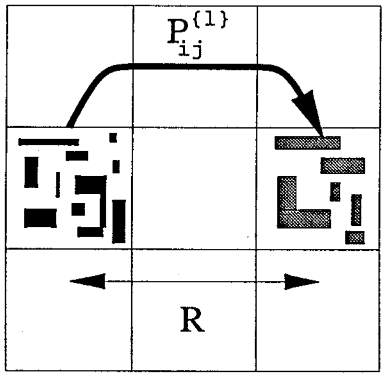


FIGURE 11-4: Two panels which are "well-separated".

Before we derive the moment-matching method, let us first make some definitions. First, let $c^{\{l\}}[m,n]$ represent the level-l *cell* at position (m,n) in the two-dimensional cell array. Then we define the normalized characteristic function $\mathcal{W}^{\{l\}}[m,n]$ associated with the cell $c^{\{l\}}[m,n]$ as

$$\mathcal{W}^{\{l\}}[m,n](r) \equiv \begin{cases} 1/a^{\{l\}} & r \in c^{\{l\}}[m,n] \\ 0 & \text{otherwise} \end{cases} \tag{11.13}$$

where $a^{\{l\}}$ is the area of an level-l cell. Also, let $f^{\{l\}}(k)$ and $g^{\{l\}}(k)$ be integer functions which return the (m,n) position indexes for the level-l cell occupied by the panel $p_k^{\{l\}}$. For example, $p_k^{\{l\}}$ is associated with the cell $c^{\{l\}}[f^{\{l\}}(k), g^{\{l\}}(k)]$. An additional definition concerns the distance between panels. Two panels at a given level l are classified as *well-separated* if they are not nearest neighbors. The equivalent definition is that they are separated by at least one intervening cell at level l. See Figure 11-4 for example.

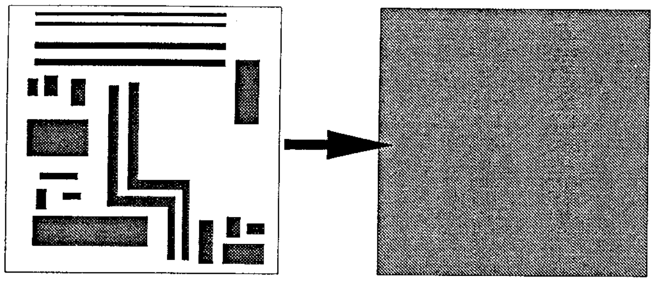


FIGURE 11-5: Zero-order approximation of coarse panel.

Assume we are to calculate $P_{ij}^{\{l\}}$, the average potential over panel $p_i^{\{l\}}$ due to a unit current distributed uniformly over panel $p_j^{\{l\}}$. If the two panels are *well-separated*, it might be a reasonable approximation to distribute the current uniformly over the *entire* cell $c^{\{l\}}[f^{\{l\}}(j), g^{\{l\}}(j)]$, and to average the resulting potential over the *entire* cell $c^{\{l\}}[f^{\{l\}}(i), g^{\{l\}}(i)]$. This amounts to approximating each panel characteristic function $\mathcal{X}_k^{\{l\}}$ with the simpler cell characteristic function $\mathcal{W}^{\{l\}}[f^{\{l\}}(k), g^{\{l\}}(k)]$, as depicted in Figure 11-5.

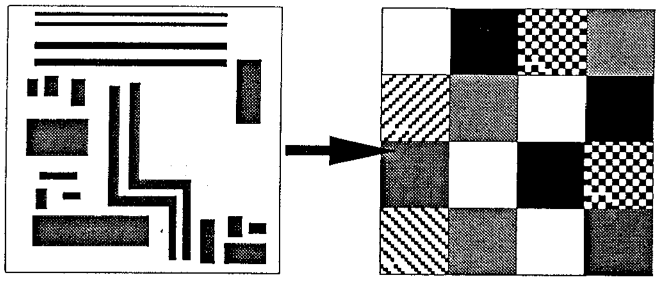


FIGURE 11-6: Third-order approximation of coarse panel.

This idea can be refined by matching higher-order moments of the characteristic function $\mathcal{X}_i^{\{l\}}$ with a regular $2^\nu \times 2^\nu$ array of characteristic functions associated with the cells at level $(l+\nu)$. For the choice $\nu = 2$, each coarse panel is approximated with $4 \times 4 = 16$ cells, as shown in Figure 11-6. Define $\overline{\mathcal{X}}_i^{\{l\}}$ as the result of combining the 4×4 level-$(l+2)$ cell characteristic functions

$$\overline{\mathcal{X}}_i^{\{l\}} \equiv \sum_{m=m_0}^{m_0+3} \sum_{n=n_0}^{n_0+3} h_i^{\{l\}}[m,n] \cdot \mathcal{W}^{\{l+2\}}[m,n] \tag{11.14}$$

where $m_0 = 4 \cdot f^{\{l\}}(i)$ and $n_0 = 4 \cdot g^{\{l\}}(i)$. There are 16 coefficients $h_i^{\{l\}}[m,n]$ to be determined for each approximate representation $\overline{\mathcal{X}}_i^{\{l\}}$. The moment matching conditions require that the Cartesian moments $Q_{\alpha\beta}^{(\gamma)}$ and $\overline{Q}_{\alpha\beta}^{(\gamma)}$ defined by

$$Q_{\alpha\beta}^{(\gamma)} = \int \mathcal{X}_i^{\{l\}} \cdot x^\alpha \cdot y^\beta \, dx \, dy,$$

$$\overline{Q}_{\alpha\beta}^{(\gamma)} = \int \overline{\mathcal{X}}_i^{\{l\}} \cdot x^\alpha \cdot y^\beta \, dx \, dy, \tag{11.15}$$

match exactly up to a certain order γ_{max} for each $\gamma = 0 : \gamma_{max}$. Since $\gamma = \alpha+\beta$ and $0 \leq \alpha, \beta \leq \gamma$ for each moment $Q_{\alpha\beta}^{(\gamma)}$, the number of moments corresponding to each order γ is $\gamma + 1$ as α ranges from zero to γ. It is easy to show that the total number of moments, or the number of constraints imposed by (11.15), is

$$m_{tot} = \frac{(\gamma_{max}+1)(\gamma_{max}+2)}{2}. \tag{11.16}$$

For a third-order approximation $\gamma_{max} = 3$, we are required to match 10 Cartesian moments according to (11.16). Given the 16 unknown coefficients $h_i^{\{l\}}[m,n]$ and 10 constraints for each $\overline{\mathcal{X}}_i^{\{l\}}$, this results in an *underdetermined* linear system, which can be solved with the singular-value decomposition (SVD) [13]. It can be shown that if the Cartesian moments match up to order γ_{max}, then the *difference* between the the potentials generated by $\overline{\mathcal{X}}_i^{\{l\}}$ and $\mathcal{X}_i^{\{l\}}$ is of order $(1/r^{\gamma_{max}+1})$, but we shall omit the proof here. Empirically, we observe that the error in the approximate $P_{ij}^{\{l\}}$ computed using third-order moment-matching is within one part in a thousand for panels which are "well-separated".

We now describe an efficient method to compute the product $v^{\{l\}} = P^{\{l\}} \cdot q^{\{l\}}$ given $q^{\{l\}}$ at a coarse level l, where $(l < l_{max} - 2)$, by combining the moment-matching approximations just described with a *precorrected*-DCT algorithm. This algorithm is divided into two stages. In the first stage, a first approximation $\overline{v}^{\{l\}}$ is computed using the moment-matched, 4×4-cell representation for each panel $p_i^{\{l\}}$. This can be performed efficiently using a Type-2 DCT and inverse DCT of size $2^{l+2} \times 2^{l+2}$. Since the moment-matching approximations are of poor accuracy for nearby panel interactions, it is necessary to compute the nearest-neighbor interactions directly for each panel, and to make appropriate corrections on the previous result $\overline{v}^{\{l\}}$. This forms the second stage of the algorithm.

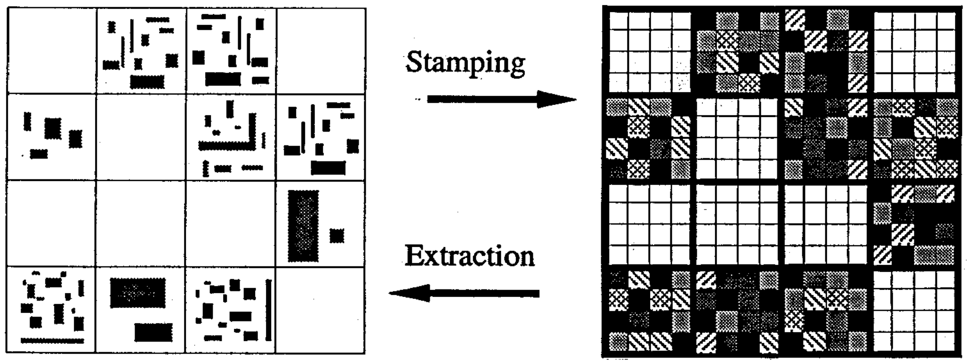


FIGURE 11-7: Adjoint operators between coarse panels and their cell representations.

The first stage of the algorithm again makes use of the sparsification via eigendecomposition technique developed in Chapter 9. Hence this is also called the *transform* stage. Since each coarse-grid panel $p_i^{\{l\}}$ is associated with 16 *geometric coefficients* $h_i^{\{l\}}[m,n]$, and since these coefficients correspond to cell currents at level $(l+2)$, we scale each set of coefficients $h_i^{\{l\}}[m,n]$ by the net panel current $q_i^{\{l\}}$ and stamp the resulting cell currents onto a regular $2^{l+2} \times 2^{l+2}$ grid. Let $A_{\{l\}}^{\{l+2\}}$ be called the *stamping* operator defined by

$$q_c^{\{l+2\}} = A_{\{l\}}^{\{l+2\}} q^{\{l\}} \tag{11.17}$$

where $q_c^{\{l+2\}}$ is the resulting cell currents at level $(l+2)$. A two-dimensional Type-2 DCT of size $2^{l+2} \times 2^{l+2}$ is then performed on the cell currents $q_c^{\{l+2\}}$. The result is multiplied with the modified eigenvalues, and then a Type-2 inverse DCT is performed to give a $2^{l+2} \times 2^{l+2}$ array of average cell potentials $v_c^{\{l+2\}}$. Let $\overline{P}_{\{l+2\}}^{\{l+2\}}$ represent this transform operation between cells

$$v_c^{\{l+2\}} = \overline{P}_{\{l+2\}}^{\{l+2\}} q_c^{\{l+2\}}. \tag{11.18}$$

Finally, the average potential $\overline{v}_i^{\{l\}}$ over each panel $p_i^{\{l\}}$ is extracted by taking a *weighted sum* over the 16 associated cell currents, with $h_i^{\{l\}}[m,n]$ now being used as the *weights*. Let $A_{\{l+2\}}^{\{l\}}$ denote the *extraction* operator defined by

$$\overline{v}^{\{l\}} = A_{\{l+2\}}^{\{l\}} v_c^{\{l+2\}}. \tag{11.19}$$

The entire computation in the first stage is then summarized as

$$\overline{v}^{\{l\}} = A_{\{l+2\}}^{\{l\}} \overline{P}_{\{l+2\}}^{\{l+2\}} A_{\{l\}}^{\{l+2\}} q^{\{l\}}. \tag{11.20}$$

Since the *same* coefficents $h_i^{\{l\}}[m,n]$ are used to expand the panel current densities as well as to compute average panel potentials, the extraction operator is the *adjoint* of the stamping operator, *i.e.*

$$A_{\{l+2\}}^{\{l\}} = \left(A_{\{l\}}^{\{l+2\}}\right)^{\dagger}. \tag{11.21}$$

This is illustrated in Figure 11-7 for an example where $l = 2$. This symmetry is a direct consequence of the Galerkin formulation (11.6). At level l, the cost of computing $\overline{v}^{\{l\}}$ given $q^{\{l\}}$ is $\mathcal{O}(2\overline{M}^2 \log_2(\overline{M}))$, where $\overline{M} \equiv 2^{l+2}$.

In the second stage, corrections are made for the large errors in nearest-neighbor panel interactions produced by moment-matching approximations during the transform stage. This is done in a similar manner as in the pre-corrected FFT scheme [62]. During the set-up phase in the multigrid algorithm, a sparse matrix $P_{NN}^{\{l\}}$ containing accurate nearest-neighbor interaction coefficients is created at each coarse level and stored for subsequent use. This can be accomplished during the smoother construction by extracting the row corresponding to panel $p_k^{\{l\}}$ from its associated local coefficient-of-potential matrix $P_{loc}^{\{l\}}[k]$ and stamping it into the k-th row of $P_{NN}^{\{l\}}$. Also during set-up, another matrix $\overline{P}_{NN}^{\{l\}}$ with the same sparsity pattern as $P_{NN}^{\{l\}}$ is constructed in a similar manner, except that the 4×4-cell representation for each panel derived from moment-matching is now used to calculate nearest-neighbor potential coefficients. With both $P_{NN}^{\{l\}}$ and $\overline{P}_{NN}^{\{l\}}$ available, the correction $\Delta v^{\{l\}}$ to be made in the second stage is

$$\begin{aligned} \Delta v^{\{l\}} &= P_{NN}^{\{l\}} q^{\{l\}} - \overline{P}_{NN}^{\{l\}} q^{\{l\}} \\ v^{\{l\}} &= \overline{v}^{\{l\}} + \Delta v^{\{l\}} \ . \end{aligned} \tag{11.22}$$

Hence the entire algorithm for computing $v^{\{l\}} = P^{\{l\}} \cdot q^{\{l\}}$ given $q^{\{l\}}$ can be summarized as

$$v^{\{l\}} = A_{\{l+2\}}^{\{l\}} \overline{P}_{\{l+2\}}^{\{l+2\}} A_{\{l\}}^{\{l+2\}} q^{\{l\}} + P_{NN}^{\{l\}} q^{\{l\}} - \overline{P}_{NN}^{\{l\}} q^{\{l\}} \ . \tag{11.23}$$

Because each panel at level $(l_{max} - 2)$ is still represented *exactly* with a 4×4 cell array, moment-matching approximations are not necessary for the finest three levels $(l_{max}, l_{max} - 1, l_{max} - 2)$, and the $P^{\{l\}} \cdot q^{\{l\}}$ product is to be computed on the finest grid. Since the cost of computing a $P^{\{l\}} \cdot q^{\{l\}}$ product becomes cheaper than $P^{\{l_{max}\}} \cdot q^{\{l_{max}\}}$ only when $l < (l_{max} - 2)$, the result is that for large l_{max}, the cost of a complete multigrid V-cycle is between three to four times that of a finest-grid calculation. This factor may be reduced as more efficient approximation and sparsification algorithms become available.

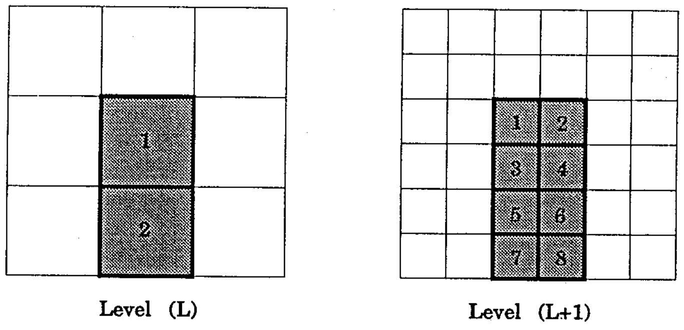


FIGURE 11-8: Hierarchical approach to compute $P_{ij}^{\{l\}}$ for coarse-grid panels.

We now turn to the problem of how to compute the Galerkin matrix element $P_{ij}^{\{l\}}$, between two coarse panels $p_i^{\{l\}}$ and $p_j^{\{l\}}$ at level l. This computation is necessary in building the local coefficient-of-potential matrices $P_{loc}^{\{l\}}[k]$ required during the construction of the correction operator $P_{NN}^{\{l\}}$ and the smoothing operator $D_{\{l\}}^{-1}$. As explained previously, $P_{ij}^{\{l\}}$ may be computed as a double sum of potential coefficients $P_{mn}^{\{l_{max}\}}$ at the finest level, but this leads to an $\mathcal{O}(N^2)$ algorithm, where N is the number of panels. Instead, we introduce here a *hierarchical* algorithm which makes use of coefficients stored at the next finer level as well as moment-matching approximations. The algorithm is as follows. Suppose that panels $p_i^{\{l\}}$ and $p_j^{\{l\}}$ are both nearest neighbors to $p_k^{\{l\}}$. Then we need to calculate $P_{ij}^{\{l\}}$ as a required entry in the matrix $P_{loc}^{\{l\}}[k]$. Further suppose that the matrix $P_{NN}^{\{l+1\}}$ has been created and stored, *i.e.* each element $P_{mn}^{\{l+1\}}$ between two nearest neighbors $p_m^{\{l+1\}}$ and $p_n^{\{l+1\}}$ are available at level $(l+1)$. If $p_i^{\{l\}}$ and $p_j^{\{l\}}$ are "well-separated", then the 4×4 moment-matching cell arrays are used to compute $P_{ij}^{\{l\}}$ at level l. If, on the other hand, $p_i^{\{l\}}$ and $p_j^{\{l\}}$ are themselves nearest neighbors, $P_{ij}^{\{l\}}$ is computed as a double sum over panel coefficients at level $(l+1)$. This double summation involves at most 16 terms since each level-l panel contains at most 4 level-$(l+1)$ panels. Each term $P_{mn}^{\{l+1\}}$ in the sum is either already available from memory, in the case where $p_m^{\{l+1\}}$ and $p_n^{\{l+1\}}$ are nearest neighbors, or it is calculated using moment-matching approximations, in the case where the panels are "well-separated" at level $(l+1)$. This requires a "bottom-up" approach, in which the finest grid is handled first, and the coarsest grid handled last. This algorithm is illustrated by an example shown in Figure 11-8, where $P_{12}^{\{l\}}$ is computed as a 4×4 double sum over $P_{mn}^{\{l+1\}}$

$$P_{12}^{\{l\}} = \sum_{m=1}^{4} \sum_{n=5}^{8} P_{mn}^{\{l+1\}} \tag{11.24}$$

where $P_{mn}^{\{l+1\}}$ is either taken directly from memory (*e.g.* $P_{35}^{\{l+1\}}$) or computed using moment-matching approximation (*e.g.* $P_{37}^{\{l+1\}}$).

12

Computational Results

In this section, we present numerical experiments comparing two iterative methods for solving (7.4): our new multigrid algorithm and the standard Generalized Minimal RESidual algorithm (GMRES [9]) *without preconditioning.* Since (7.4) results from a first-kind integral operator (7.3), the smallest eigenvalues of the matrix $P_{\{l\}}$ approach zero with increasing mesh refinement [5] and $P_{\{l\}}$ becomes more *ill-conditioned.* It is well-known that Krylov-subspace based iterative methods such as GMRES or CG (Conjugate Gradient) suffer from slow convergence for ill-conditioned linear systems [14]. Although it is possible to appply *preconditioning* to the linear system (7.4) to accelerate GMRES convergence in a similar way as done in [3], an increasing number of iterations is still required for finer discretizations. We demonstrate here that the multigrid algorithm resolves this difficulty by retaining a *constant* convergence rate per iteration, independent of mesh refinement, and hence problem size. Thus, for a fixed relative error tolerance $\|r^{(k)}\|/\|r^{(0)}\| < \varepsilon$, the number of multigrid (MG) iterations required does not grow with mesh refinement.

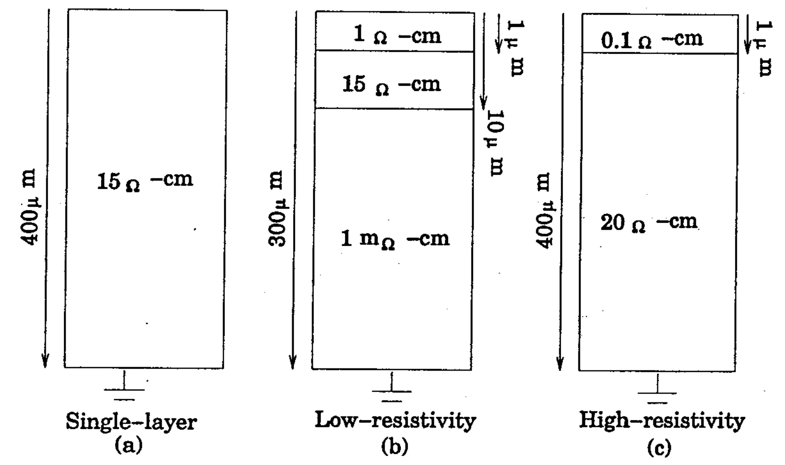


FIGURE 12-1: Example substrate profiles.

In Figure 12-1, we display three possible vertical substrate profiles used in this chapter: the single-layer substrate, the low-resistivity substrate, and the high-resistivity substrate. The lateral dimensions of the substrate is always assumed to be 1mm × 1mm (or 1000 μm × 1000 μm).

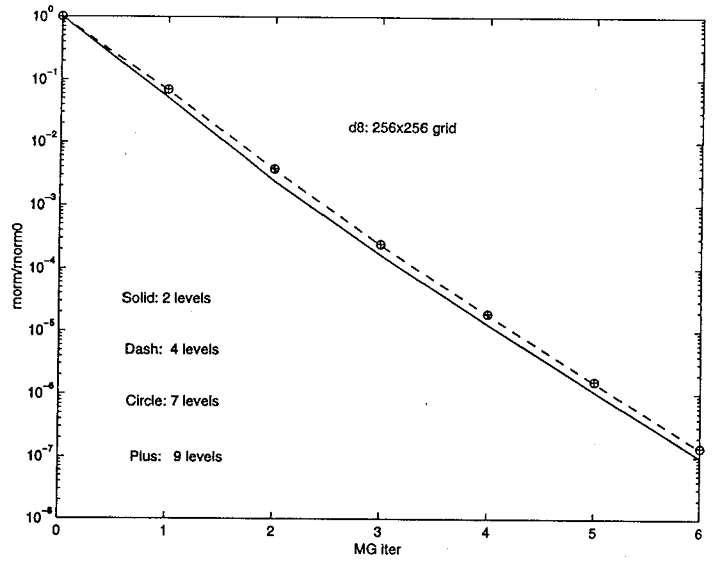


FIGURE 12-2: TGM vs. MGM convergence rates.

The efficiency of multigrid algorithms in general arises from the fact that the *smoothing operator* $M_{\{l\}} \equiv -D_{\{l\}}^{-1}S_{\{l\}}$ at each level reduces the corresponding error components by the *same* numeric ratio [46]. This was observed in Chapter 10 for the integral equation defined over the entire substrate $\Omega = [0, a] \times [0, b]$. We show that when the multigrid algorithm developed in Chapter 11 is used to solve problems involving many irregularly shaped contacts, the same results hold. For this purpose, a test layout was created. The single-layer substrate is first covered with a regular, 256 × 256 grid. Then half of the cells on this grid are randomly selected and labeled as panels. This gives $l_{max} = 8$ at the finest level. We apply a two-level, four-level, seven-level, and nine-level multigrid iteration to solve this problem, and plot the resulting normalized residual, $\|P_{\{l\}}q_{\{l\}}^{(k)} - v_{\{l\}}\|/\|v_{\{l\}}\|$, versus the MG iteration count in Figure 12-2. The coarsest level is $l_{min} = 7$ for the two-level algorithm (TGM) and $l_{min} = 0$ for the nine-level algorithm (MGM). Because the same convergence rate of about an order of magnitude per iteration is observed for multigrid methods of varying *depths*, we conclude that the smoother indeed reduces the error at each length scale by the same factor for the case of irregular geometries. Recall that MGM requires only the *application* of operators $P_{\{l\}}$ at various levels (and the solution of a scalar equation at the coarsest level $l = 0$), whereas TGM requires *solution* of the system at level $(l - 1)$. Hence the MGM iteration is always cheaper to apply than the TGM iteration. Since the size of the linear system decreases *geometrically* with the level index, the cost of an MGM iteration is a constant multiple of an operator application at the finest level, or equivalently, a single GMRES iteration. Using the algorithms developed in Section 11.2, we observe this factor to be three to four in our implementation.

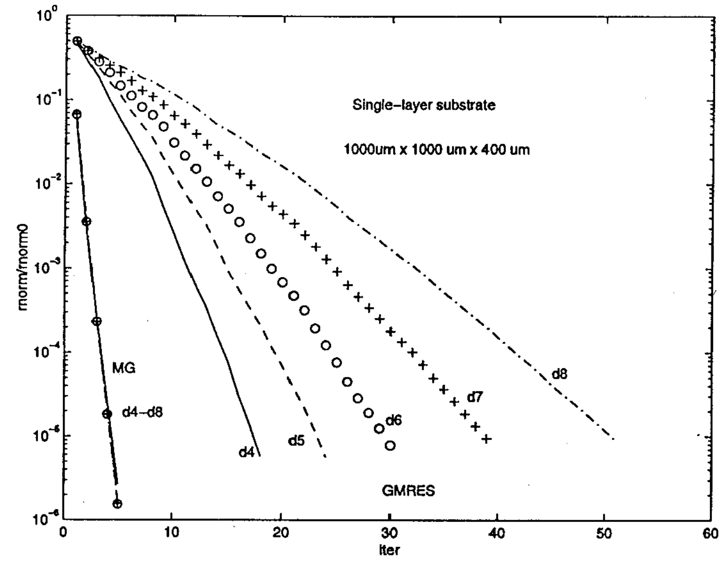
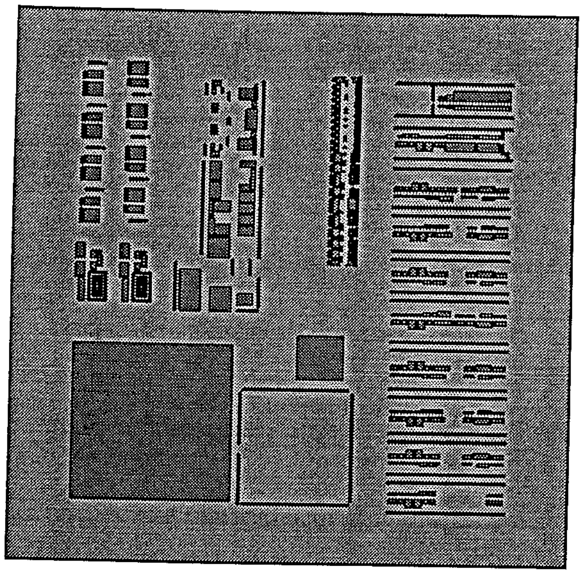


FIGURE 12-3: Effect of mesh refinement on convergence.

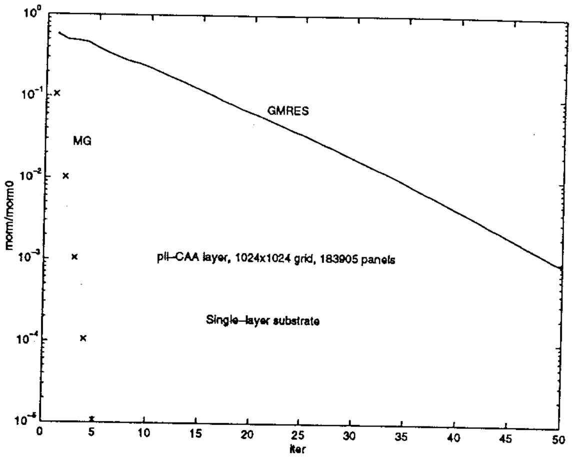
The crucial feature of multigrid schemes is that the convergence rate is independent of discretization, and hence problem size. We perform our next experiment with the single-layer substrate on five random test layouts of increasing mesh refinement, created in a similar manner as the previous example and labeled $d = 4$ through $d = 8$. The $d = 4$ layout is discretized on a 16×16 grid, and the $d = 8$ layout is discretized on a 256×256 grid. The multigrid method with maximum depth ($l_{min} = 0, l_{max} = d$) is applied to solve each problem. The observed MG convergence rate is indeed independent of mesh size, as shown by the residual versus iteration plot in Figure 12-3. Also displayed are the GMRES convergence rates, which deteriorate with increasing mesh refinement as expected. Since the cost of a single MG iteration is a constant multiple of that of a GMRES iteration (three to four in our case), it is clear that MG is superior to GMRES, especially for large problems requiring fine discretization.

To show that the multigrid approach can be applied to realistic problems, we perform substrate parameter extraction on a a Phase Lock Loop (PLL) frequency synthesizer circuit [63] on a 1mm $\times$ 1mm chip. There are 478 substrate contacts defined by the active layer mask CAA, shown in Figure 12-4(a). Discretized with the help of a 1024×1024 grid, the total number of panels, or minimum-size cells, is $N = 183905$. This corresponds to roughly 20% of the chip area. The resolution thus achieved is about 1 micron.

The GMRES algorithm with sparsification via eigendecomposition [50] is used as a timing benchmark. Multigrid and GMRES convergence rates for a single solution of (7.4) are plotted in Figure 12-4(b) assuming the single-layer substrate profile. Similar results are shown in Figure 12-5(a) for the low-resistivity profile and in Figure 12-5(b) for the high-resistivity profile. In

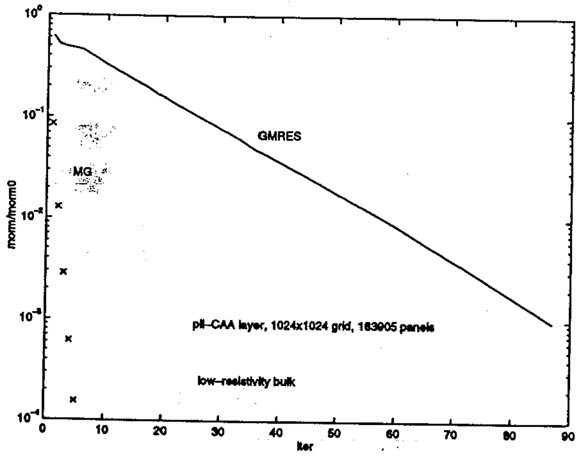


(a)

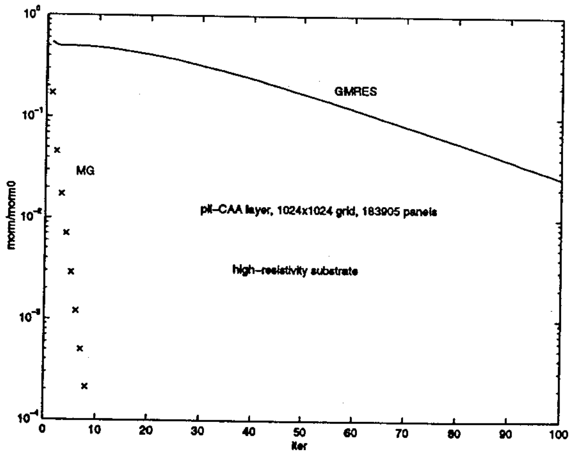


(b)

FIGURE 12-4: (a) PLL active area layout. (b) PLL on single-layer substrate.



(a)



(b)

FIGURE 12-5: (a) PLL on low-resistivity substrate. (b) PLL on high-resistivity substrate.

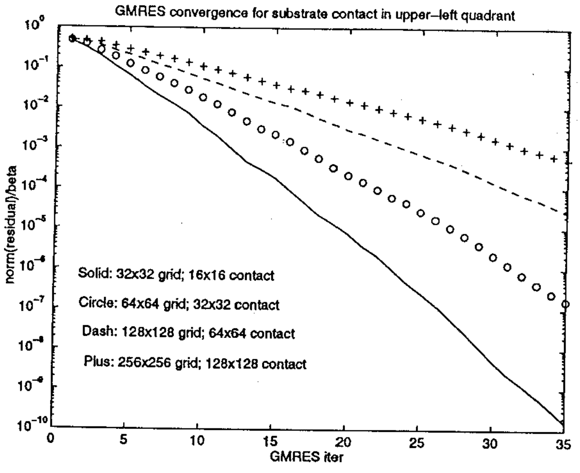
all three cases it is seen that MG converges much more rapidly than GMRES. However, each multigrid iteration costs more than each GMRES iteration, and the multigrid setup cost is many times that of the GMRES setup. We demonstrate the efficiency of MG versus GMRES by comparing the CPU times required to extract the entire 478×478 substrate conductance matrix. We also require convergence to a tolerance of 1e-3 in the relative residual norm for each of the 478 solves. The timing results are summarized in Table 12-1. It is seen from the total extraction time that MG is faster than GMRES by almost an order of magnitude for the low-resistivity and high-resistivity substrates. More significant gains will be seen for even larger problems requiring finer meshes.

	Setup time	Time per iter	Iters per solve	Time per solve	Total time
MG (single)	317s	28.5s	3	85.5s	11.4h
GMRES (single)	11.2s	7.32s	50	366s	48.6h
MG (lo-res)	343s	28.4s	4	114s	15.2h
GMRES (lo-res)	29.9s	8.1s	95	771s	102h
MG (hi-res)	333s	28.0s	6	168s	22.3h
GMRES (hi-res)	23.0s	8.4s	180	1512s	201h

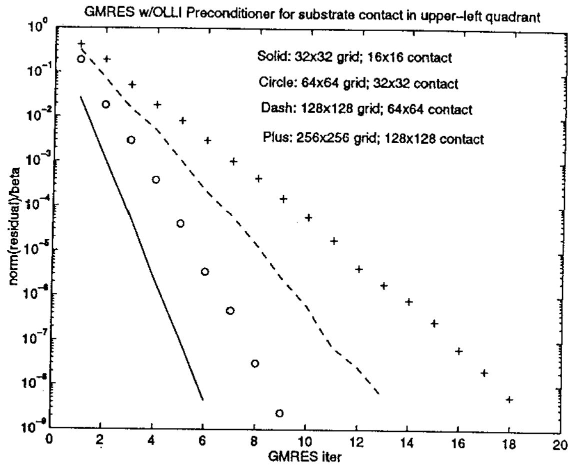
Table 12-1: Computational cost for PLL substrate extraction.

A possible limitation of the multigrid algorithm is that it resolves only ill-conditioning caused by mesh refinement. It is less effective in dealing with ill-conditioning caused by the apparent loss of groundplane, as seen in a slowdown of multigrid convergence for the high-resistivity case in Table 12-1. The reason for this difficulty is that the Green's function is raised by a constant DC level as the bulk resistivity increases. This causes the eigenvalues for the low-frequency modes to move toward zero, causing the linear system to be much more ill-conditioned than in the single-layer substrate case. This ill-conditioning is extremely difficult to resolve because the eigenvalue spectrum of the discretized system essentially resembles that of a pseudo-differential operator of order $-\alpha$ where α can be much larger than one. The true nature of the integral equation will not be observed until the discretization become extremely fine. To maintain the optimal multigrid convergence rate for the high-resistivity case, it is necessary to *explicitly solve* the problem at a level $l_{min} > 0$ which is fine enough to capture the ill-behaved part of the eigenspectrum. However, even this strategy will fail as the bulk resistivity becomes high enough to make the condition number larger than machine precision. For such problems to be well-posed, there must be *substrate plugs* which help establish substrate potential from the *top surface.* These additional *Dirichlet* boundary conditions give rise to well-behaved Green's functions.

Finally, we investigate the effects of various *preconditioners* for the GMRES iterative solver. The test layout for this case is a large contact which occupies a quadrant of the chip area. The coarsest discretization used for the contact is 16×16 on a 32×32 substrate grid, and the finest discretization used for the contact is 128×128 on a 256×256 substrate grid. For each case, the



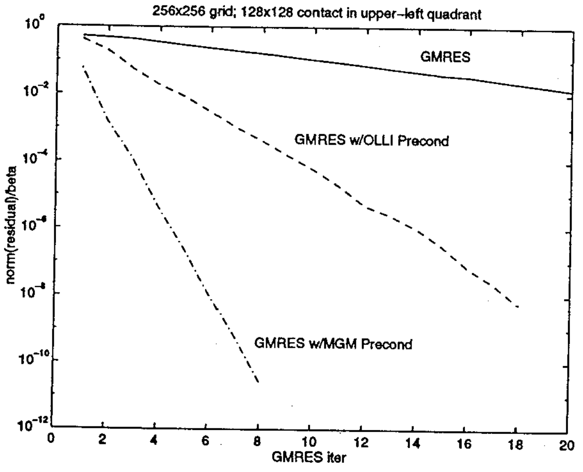
(a)



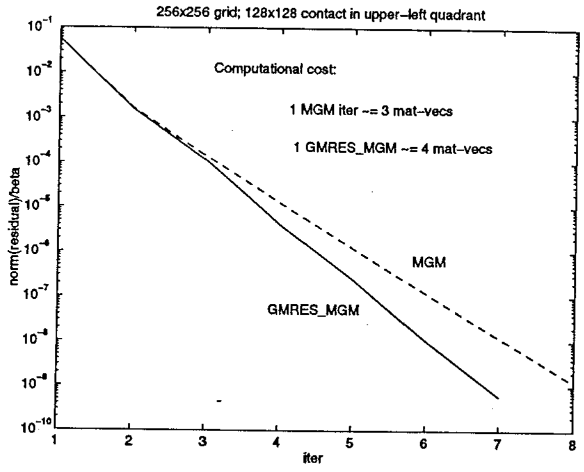
(b)

FIGURE 12-6: (a) GMRES w/o preconditioner. (b) GMRES with nearest neighbor preconditioner.

right-hand side is set to be a random vector. Figure 12-6(a) displays the convergence rate for GMRES without preconditioning. As expected, convergence slows down with increasing mesh refinement. Figure 12-6(b) displays the convergence rate for GMRES with an over-lapping, local-inversion (OLLI) preconditioner using nearest neighbors. This preconditioner is equivalent to the finest-level smoother used in the multigrid algorithm. Convergence is enhanced by this preconditioner, but still slows down with finer meshing. Next, we consider using the multigrid V-cycle (MGM) as a *multilevel* preconditioner for GMRES. We observe that as expected, the resulting convergence rate is independent of discretization, just as in the multigrid case. Figure 12-7(a) compares convergence rates for the GMRES algorithm with MGM preconditioning, OLLI preconditioning, and without preconditioning. The multigrid-preconditioned GMRES algorithm has essentially the *same* convergence rate as the stand-alone multigrid iteration, as demonstrated by the plot in Figure 12-7(b). It is seen that after three iterations, both methods have knocked down the relative residual to one part in ten thousand. After seven preconditioned GMRES iterations, it is only one iteration "ahead" of plain multigrid. This confirms our earlier assertion that the multigrid iteration is already near optimal, and hence cannot be improved upon very much with a Krylov-subspace search.



(a)



(b)

FIGURE 12-7: (a) GMRES with various preconditioners. (b) Multigrid alone is "good enough".

13 Conclusion

In this thesis, we have investigated the numerical solution of integral equations arising from two distinct problems in the computer-aided design of VLSI circuits. Iterative solvers converge slowly for both problems because of ill-conditioning. However, the source of ill-conditioning is very different in the two cases, as are their remedies.

In the first part of the thesis, we investigated the boundary-element formulation of the transient interconnect problem. We showed that while this approach eliminates the need for exterior volume meshing, it produces large errors when multipole-accelerated due to the somewhat poor conditioning of the problem. The ill-conditioning is inherent to problems with a large range of time constants, or natural frequencies. This magnified error is eliminated in the alternative mixed surface-volume formulation, in which the ill-conditioned interior Laplace problem is separated from the well-conditioned capacitance problem and solved explicitly at a small additional cost.

To construct reduced-order models by matching Taylor series terms of the transfer function at $s = 0$, iterative solutions of a linear system must be performed repeatedly. A large number of iterations are required for ill-conditioned problems, such as those involving long wires. By reformulating the surface-volume approach slightly, we found natural preconditioners which produce rapid convergence in the iterative solve. We presented results which demonstrate that the cost of computing a fixed-order reduced model is order N, independent of condition number, and is only several times that of a multipole-accelerated capacitance extraction. Then we used our multipole-accelerated code to investigate the accuracy of the one-dimensional diffusion equation for long RC lines. Our simulations show that the diffusion equation is accurate up to relatively high frequencies, unless the line is some distance from the ground plane.

In the second part of this thesis, we focused on solving the first-kind integral formulation of the substrate extraction problem. It is well-known that both classical and Krylov-subspace iterative algorithms converge slowly for ill-conditioned linear systems. However, we recognized that when the linear system is derived from a first-kind integral equation, the eigenvalues are intimately related to the characteristic length scales, or spatial frequencies, of the eigenmodes. Specifically, the eigenvalue approaches zero monotonically as the spatial frequency of the eigenmode increases. Because of this special connection, it is possible to remove the effects of ill-conditioning by breaking up the original problem into a sequence of sub-problems, each with a distinct characteristic length scale. This motivated our development of a multigrid iterative method.

The two core components of the multigrid scheme were developed for first-kind integral equations defined on irregular geometries. We constructed the first component, the smoothing operator, by solving a series of local, over-lapping sub-problems in an attempt to approximate the inverse operator directly. The second component is the discrete representation of the original problem at various length scales. This was accomplished with the construction of a hierarchy of characteristic functions which form subspaces of the space spanned by the original panel functions. The Galerkin integrals then lead naturally to a multilevel discrete representation.

For an efficient implementation of the multigrid scheme, it is necessary to sparsify the dense matrix-vector multiplications required at each level. For the finest level, we developed an eigendecomposition approach which takes advantage of regularity and homogeneity in the distribution of panels, or current sources. For coarser levels, we developed moment-matching approximations which transform the current distribution into a regular array, on which the eigendecomposition technique may be applied. Our sparsification approach accounts properly for substrate edge effects, unlike previously applied multipole approximations.

Results on realistic examples demonstrate that the multigrid approach combined with sparsification via moment-matching and eigenexpansion is up to an order of magnitude faster than the sparsification plus a Krylov-subspace method, and orders of magnitude faster than not using sparsification at all. We believe that the ideas proposed here can be generalized to solving other problems arising from first-kind integral equations defined over complicated surfaces, such as BEM capacitance extraction [3].

Bibliography

[1] V. Rokhlin. Rapid Solution of Integral Equations of Classical Potential Theory. *Journal of Computational Physics*, 60(2):187–207, September 1985.

[2] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. *Journal of Computational Physics*, 90:348–370, 1990.

[3] K. Nabors and J. White. Fastcap: A Multipole Accelerated 3-D Capacitance Extraction Program. *IEEE Transactions on Computer-Aided Design*, pages 1447–1459, November 1991.

[4] 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.

[5] R. Kress. *Linear Integral Equations.* Springer-Verlag, 1989.

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

[7] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.

[8] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* M.I.T. Press, Cambridge, Massachusetts, 1988.

[9] Y. Saad and M. H. Schultz. Gmres: A generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM J. Stat. Comp.*, 7:856–869, 1986.

[10] T. Korsmeyer and J. White. Multipole-accelerated preconditioned iterative methods for three-dimensional potential integral equations. In *Proceedings of Boundary Element Methods 15 (BEM15)*, August 1993.

[11] 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.

[12] G. Dahlquist and A. Bjorck. *Numerical Methods.* Prentice Hall, 1974.

[13] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C.* Cambridge University Press, second edition, 1992.

[14] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis.* Springer-Verlag, second edition, 1993.

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

[16] A. Greenbaum, L. Greengard, and G. B. McFadden. Laplace's equation and the dirichlet-neumann map in multiply connected domains. *Journal of Computational Physics*, 105:267–278, 1993.

[17] Nishath K. Verghese, David J. Allstot, and Mark A. Wolfe. Verification techniques for substrate coupling and their application to mixed-signal ic design. *IEEE Journal Solid-State Circuits*, 31(3):354–365, March 1996.

[18] Nishath Verghese. *Extraction and Simulation Techniques for Substrate-Coupled Noise in Mixed-Signal Integrated Circuits.* PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, August 1995.

[19] S. Kumashiro, R. Rohrer, and A. Strojwas. A new efficient method for the transient simulation of three-dimensional interconnect structures. In *Proc. Int. Electron Devices Meeting*, pages 193–196, December 1990.

[20] D. Ling, S. Kim, and J. White. A boundary-element approach to transient simulation of three-dimensional integrated circuit interconnect. In *Proceedings of the 29th Design Automation Conference*, pages 93–98, June 1992.

[21] Lawrence T. Pillage and Ronald A. Rohrer. Asymptotic Waveform Evaluation for Timing Analysis. *IEEE Trans. CAD*, 9(4):352–366, April 1990.

[22] Peter Feldmann and Roland W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. In *Proceeding of the Euro-DAC*, pages 170–175, September 1994.

[23] L. Miguel Silveira, Mattan Kamon, and Jacob K. White. Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-d interconnect structures. In 32^{nd} *ACM/IEEE Design Automation Conference*, pages 376–380, San Francisco, California, June 1995.

[24] H. Haus and J. Melcher. *Electromagnetic Fields and Energy.* Prentice Hall, Englewood Cliffs, N.J., 1989.

[25] J. D. Jackson. *Classical Electrodynamics.* John Wiley & Sons, New York, second edition, 1975.

[26] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* M.I.T. Press, Cambridge, Massachusetts, 1988.

[27] T. Korsemeyer, K. Nabors, and J. White. *FastLap: Version 1.0.* Computational Hydrodynamics Facility and Research Laborotory of Electronics, M.I.T., Cambridge, MA 02139, U.S.A., April 1993.

[28] A. H. Schatz, V. Thomée, and W. L. Wendland. *Mathematical Theory of Finite and Boundary Element Methods.* Birkhäuser Verlag, Basel/Boston/Berlin, 1990.

[29] L. Miguel Silveira, Mattan Kamon, and Jacob K. White. Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-d interconnect structures. In *Proceedings of the European Design and Test Conference*, pages 534–538, Paris, France, March 1995.

[30] Kevin J. Kerns, Ivan L. Wemple, and Andrew T. Yang. Stable and efficient reduction of substrate model networks using congruence transforms. In *International Conference on Computer Aided-Design*, pages 207–214, San Jose, California, November 1995.

[31] Peter Feldmann and Roland W. Freund. Reduced-Order Modeling of Large Linear Subcircuits via a Block Lanczos Algorithm. In 32^{nd} *ACM/IEEE Design Automation Conference*, pages 474–479, San Francisco, California, June 1995.

[32] C.L. Ratzlaff, N. Gopal, and L.T. Pillage. RICE: Rapid Interconnect Circuit Evaluator. In 28^{th} *ACM/IEEE Design Automation Conference*, pages 555–560, June 1991.

[33] J.R. Phillips and J.K. White. Efficient Capacitance Extraction of 3D Structures using Generalized Precorrected FFT Methods. In *Proceedings of the IEEE 3rd Topical Meeting on Electrical Performance of Electronic Packaging*, pages 253–256, November 1994.

[34] D.K. Su, M.J. Loinaz, S. Masui, and B.A. Wooley. Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits. *IEEEJSSC*, 28(4):420–430, April 1993.

[35] T. A. Johnson, R.W. Knepper, V. Marcellu, and W. Wang. Chip substrate resistance modeling technique for integrated circuit design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, CAD-3(2):126–134, 1984.

[36] Bram Nauta and Gian Hoogzaad. How to deal with substrate noise in analog cmos circuits. In *European Conference on Circuit Theory and Design*, pages Late 12:1–6, Budapest, Hungary, September 1997.

[37] Ranjit Gharpurey. *Modeling and Analysis of Substrate Coupling in Integrated Circuits.* PhD thesis, Department of Electrical Engineering and Computer Science, University of California at Berkeley, Berkeley, CA, June 1995.

[38] Sujoy Mitra, R. A. Rutenbar, L. R. Carley, and D. J. Allstot. A methodology for rapid estimation of substrate-coupled switching noise. In *IEEE 1995 Custom Integrated Circuits Conference*, pages 129–132, 1995.

[39] B.R. Stanisic, N.K. Verghese, R.A. Rutenbar, L.R. Carley, and D.J. Allstot. Addressing substrate coupling in mixed-mode ic's: Simulation and power distribution synthesis. *IEEEJSSC*, 29(3):226–238, March 1994.

[40] Ivan L. Wemple and Andrew T. Yang. Mixed-signal switching noise analysis using voronoi-tesselation substrate macromodels. In 32^{nd} *ACM/IEEE Design Automation Conference*, pages 439–444, San Francisco, CA, June 1995.

[41] T. Smedes, N. P. van der Meijs, and A. J. van Genderen. Extraction of circuit models for substrate cross-talk. In *International Conference on Computer Aided-Design*, pages 199–206, San Jose, CA, November 1995.

[42] R. Gharpurey and R.G. Meyer. Modeling and analysis of substrate coupling in integrated circuits. In *IEEE 1995 Custom Integrated Circuits Conference*, pages 125–128, 1995.

[43] Ranjit Gharpurey and Robert G. Meyer. Modeling and analysis of substrate coupling in integrated circuits. *IEEE Journal Solid-State Circuits*, 31(3):344–353, March 1996.

[44] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, April 1977.

[45] W. Hackbusch. *Multi-Grid Methods and Applications.* Springer-Verlag, Berlin Heidelberg New York Tokyo, 1985.

[46] W. L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, 1987.

[47] A. Brandt. Guide to multigrid development. In W. Hackbusch and U. Trottenberg, editors, *Multigrid Methods. Proceedings of the Conference Held at Koln-Porz, November 23-27, 1981*, pages 220–312. Springer-Verlag, Berlin Heidelberg New York, 1982.

[48] Hermann A. Haus and James R. Melcher. *Electromagnetic Fields and Energy.* Prentice Hall, Englewood Cliffs, New Jersey, First edition, 1989.

[49] A. J. van Genderen, N. P. van der Meijs, and T. Smedes. Fast computation of substrate resistances in large circuit. In *International Conference on Computer Aided-Design*, San Jose, CA, November 1996.

[50] J. P. Costa, M. Chou, and L. M. Silveira. Efficient techniques for accurate modeling and simulation of substrate coupling in mixed-signal ic's. In *DATE'98 - Design, Automation and Test in Europe, Exhibition and Conference*, pages 892–898, Paris, France, February 1998.

[51] G. C. Hsiao and R. E. Kleinman. Error control in numerical solutions of boundary integral equatrions. *unpublished.*

[52] Y. Saad. *Numerical Methods for Large Eigenvalue Problems.* Manchester University Press, UK, 1992.

[53] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Don't Know, 1994?

[54] J. Makhoul. A fast cosine transform in one and two dimensions. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-28(1):27–34, February 1980.

[55] K. R. Rao. *Discrete Cosine Transform: Algorithms, Advantages, and Applications.* Academic Press, Inc., San Diego, CA., 1990.

[56] F. A. Kamangar and K. R. Rao. Fast algorithms for the 2-d discrete cosine transform. *IEEE Transactions on Computers*, C-31(9):899–906, September 1982.

[57] H. Nussbaumer. Fast multidimensional discrete cosine transforms. *IBM Technical Disclose Bulletin*, 23(5):1976–1981, October 1980.

[58] H. Nussbaumer. Improved approach for the computation of multidimensional discrete cosine transforms. *IBM Technical Disclose Bulletin*, 23(10):4517–4521, March 1981.

[59] S. A. Vavasis. Preconditioning for boundary integral equations. *SIAM J. Matrix Anal. Appl.*, 13:905–925, 1992.

[60] V. Rohklin. Rapid solution of integral equation of classical potential theory. *J. Comput. Phys.*, 60:187–207, 1985.

[61] C.L. Berman. Grid-multipole calculations. *SIAM J. Sci. Comput.*, 16(5):1082–1091, September 1995.

[62] J. R. Phillips. *Rapid Solution of Potential Integral Equations in Complicated 3-Dimensional Geometries.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, June 1997.

[63] E. Charbon, R. Gharpurey, R.G. Mayer, and A. Sangiovanni Vincentelli. Semi-analytical techniques for substrate characterization in the design of mixed-signal ics. In *International Conference on Computer Aided-Design*, pages 455–462, San Jose, CA, November 1996.